Lumbini Technological University

Institute of Engineering and Information Technology

Banke, Lumbini Province, Nepal

B. Tech. in Computer Science and Artificial Intelligence (B.Tech. in CSAI)

Program Code: 102

Syllabus

October 2023

B. Tech. in Computer Science & Artificial Intelligence

1. Introduction

Lumbini Technological University (LTU) was established on July 2, 2022 (Ashar 18, 2079 BS) by the Act of Province Legislature with the goal of advancing higher education in the field of information technology, engineering, agricultural and forestry, and tourism, among others, with a particular focus on the application and development of technology and innovation.

LTU's mission is to promote and offer comprehensive programs at undergraduate and graduate levels that produce graduates who have both disciplinary expertise and the ability to handle real-world problems by combining theoretical knowledge with practical application along with exposure visits to world-renowned technological schools. LTU recognizes that academic programs are not the only way to foster a rich learning experience. To this end, it also places a strong emphasis on research and continuous education programs by supporting and encouraging students and faculty to engage in rigorous research activities, thus fostering a culture of innovation and creativity that benefits both students and faculty alike.

B. Tech. in Computer Science and Artificial Intelligence (B. Tech. in CSAI) Program

Advancements in computer science and artificial intelligence (AI) are profoundly shaping today's world, revolutionizing industries, and redefining how we interact with technology. Computer science innovations power the backbone of our digital infrastructure, while AI algorithms analyze vast amounts of data to make informed decisions, automate tasks, and improve efficiency across sectors. The fusion of computer science and AI is not only enhancing productivity and convenience but also raising questions about ethics, privacy, and the future of work in our increasingly connected and intelligent world.

The B. Tech. in CSAI program at Lumbini Technological University is designed to produce skilled IT professionals who can contribute to the advancement of technology and address the evolving needs of the digital era. Through a blend of theoretical knowledge, practical experience, and industry exposure, students are prepared to embark on successful and fulfilling careers in the dynamic and ever-growing field of Information Technology.

2. Duration and Total Credit

The B. Tech. in CSAI program spans over four years, divided into eight semesters. The total credit of courses is 127, where 1 credit is equivalent to 16 hours of lecture.

3. Eligibility for Admission

Applicants must have passed Class 12 from National Examinations Board (NEB) or equivalent with minimum C Grade / Second Division (Grade D in case of A-level) with Physics, Chemistry and Mathematics. They must pass Lumbini Technological University Entrance Test for admission.

4. Curriculum

The curriculum of B. Tech. in IT program is meticulously designed to cover a wide array of topics, blending foundational computer science principles with advanced courses in artificial intelligence., with the following objectives:

- Provide students with a solid foundation in the fundamental concepts and principles of computer science and AI.
- Equip students with practical skills, tools, and techniques necessary for solving real-world problems in these fields.

- Foster critical thinking, problem-solving, and analytical skills among students.
- Provide students with opportunities to work in teams and collaborate on projects.
- Encourage innovation, creativity, and entrepreneurship among students.
- Foster a sense of ethics, responsibility, and social awareness among students.
- Prepare students for successful careers or further studies in related fields.

In order to achieve these objectives, the curriculum comprises the following components:

- **Core Courses:** These courses provide a solid foundation in computer science and AI. Students are also required to complete courses in mathematics, natural sciences, social sciences, and humanities to develop critical thinking and communication skills.
- **Electives:** These courses provide flexibility to the students to choose elective courses that align with their interests and career goals.
- **Project-based Learning:** Students are required to work on a series of projects throughout the program. These projects provide students with opportunities to apply their knowledge and skills to real-world problems, work in teams, and develop communication and collaboration skills.
- **Industry Engagement:** The program provides opportunities for students to engage with industries (including government agencies) through internships, co-op programs, and guest lectures by industry professionals.
- **Research and Innovation:** Students are encouraged for research activities, innovation challenges, and participation in computer science and AI-related competitions to foster creativity and entrepreneurship among students.

The course structure is presented in Table 1.

5. Evaluation

Students are evaluated based on their performance in both internal assessments and semester exams. Final evaluation employs absolute grading according to the grade scale outlined in the Student Registration and Examination Guidelines, BS 2080.

The model question for semester examination is presented in Annex A.

6. Syllabus

The detailed syllabus of courses is presented in Annex B.

Table 1: Course structure of B. Tech. in CSAI program

First Year			
Semester I	**Credit**	**Semester II**	**Credit**
BAI101: Mathematics I	3	BAI151: Mathematics II	3
BAI102: English Communication	3	BAI152: Discrete Mathematics	3
BAI103: Physics of Computing	3	BAI153: Operating System	3
BAI104: Foundations of Information Technology	3	BAI154: OOP in Java	3
BAI105: C Programming	3	BAI155: Digital Logic	3
Total	**15**	**Total**	**15**
Second Year			
Semester III	**Credit**	**Semester IV**	**Credit**
BAI201: Statistics	3	BAI251: Research Methodology	3
BAI202: Database Management System	3	BAI252: Python Programming	3
BAI203: Web Technology	3	BAI253: Theory of Computation	3
BAI204: Data Structure & Algorithms	3	BAI254: Numerical Method	3
BAI205: Microprocessor and Computer Architecture	3	BAI255: Software Engineering	3
BAI206: System Analysis and Design	3	BAI290: Project I	2
		Non-credit: UI/UX Design	
Total	**18**	**Total**	**17**
Third Year			
Semester V	**Credit**	**Semester VI**	**Credit**
BAI301: Artificial Intelligence	3	BAI351: Artificial Neural Networks	3
BAI302: Design & Analysis and Algorithms	3	BAI352: Cloud Computing	3
BAI303: Compiler Design	3	BAI353: Cryptography & Network Security	3
BAI304: Data Science	3	BAI354: Image Processing	3
BAI305: Computer Graphics	3	Elective I	3
BAI306: Data Communication & Computer Networks	3		
Total	**18**	**Total**	**15**
Fourth Year			
Semester VII		**Semester VIII**	**Credit**
BAI 401: IT Project Management	3	BAI451: IT Entrepreneurship	3
BAI 402: Machine Learning	3	BAI452: Natural Language Processing	3
BAI 403: Blockchain Technology	3	BAI499: Internship	6
Elective II	3		
BAI 445: Capstone Project	5		
Total	**17**	**Total**	**12**
		Total Credit	**127**

Note: The course code of courses offered in B. Tech. in CSAI starts with 'BAI'.

Annex A: Model Question Format

	Lumbini Technological University **Institute of Engineering and Information Technology** **Examination Division** Banke, Lumbini Province Nepal **Exam Year/Month**				
Program:				**Full Marks:**	40
Course Title:				**Pass Marks:**	20
Course Code:		**Year/Semester:**		**Time:**	2 hrs

Candidates are required to give their answers in their own words as far as possible. The figures in the margin indicate full marks.

Group - A

Short Answer Questions

Attempt Any FIVE questions [5x3=15]

1. Question 1
2. Question 2
3. Question 3
4. Question 4
5. Question 5
6. Question 6

Group - B

Long Answer Questions

Attempt Any FIVE questions [5x5=25]

7. Question 7
8. Question 8
9. Question 9
10. Question 10
11. Question 11
12. Question 12

Annex B: Syllabus

Year I Semester I

Mathematics I

Course Code	BAI101	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	0
		Total	64 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding basic concept and its uses to solving the problems of limits, continuity, differentiation and integration with their applications, vector algebra, analytic geometry and vector-valued functions, multivariable calculus.

2. Course Objectives

The course aims to achieve the following objectives:

- Understand and formulate real world problems into mathematical statements.
- Develop solutions to mathematical problems at the level appropriate to the course.
- Describe or demonstrate mathematical solutions either numerically or graphically.

3. Course Detail

3.1. Theory (48 hrs)

Unit 1: Limits and Continuity **(6 hrs)**

Limit of a function; one-sided limits; limits at infinity; asymptotes; continuity of a function.

Unit 2: Differentiation and its Applications **(10 hrs)**

Derivative; geometrical interpretation; differentiable function at a point; differentiability implies continuity; derivatives of different functions; implicit differentiation; maximum and minimum values; theorem (extreme value and Fermat's) (without proof); Rolle's Theorem; Mean-Value Theorem; geometrical meaning of Mean-Value Theorem; monotonic functions; indeterminate forms and L'Hospital's rule; Newton's method.

Unit 3: Integration and its Applications **(12 hrs)**

Integration; integration by parts; trigonometric integrals; trigonometric substitutions; definite integral; properties of definite integral; improper integrals; area between the curves; arc length; area of surface of revolution.

Unit 4: Vector Algebra **(7 hrs)**

Scalar and vector product of two vectors; geometrical meaning of vector product of two vectors; scalar and vector product of three vectors; geometrical meaning of scalar product of three vectors; reciprocal vectors of three vectors; scalar and vector product of four vectors; application of vector algebra: projection of a vector onto another vector, equation of line and plane by vector method.

Unit 5: Analytic Geometry and Vector-Valued Functions **(6 hrs)**

Conic sections; classifying conic sections by eccentricity; cylindrical and spherical coordinates; vector functions; tangent vectors; curvature and normal vectors; torsion and binormal vectors.

Unit 6: Multivariable Calculus **(7 hrs)**

Functions of several variables; limits and continuity; partial derivatives; directional derivatives and gradient vectors; tangent planes; maximum and minimum values; Lagrange multipliers.

3.2 Tutorials

- Find limit of a function
- Find different asymptotes to a curve
- Examine the continuity of a function at a point and find limiting value
- Find derivative of a function at a point
- Show that differentiability implies continuity but the converse may not be true, problems on derivative of implicit function
- Find extreme value
- Problems in Mean value theorems: Rolle's theorem, Lagrange's mean value theorem and related problems
- Solve problems using L'Hospital's rule
- Evaluate various indefinite integrals
- Find value of Definite integrals, Improper Integrals
- Evaluation of area, arc length and area of revolution
- Find the value using Dot and vector product between 2, 3 and 4 vectors
- Find area by scalar product of 2 vectors and volume by scalar product of 3 vectors
- Find the projection vector and its value and Reciprocal vector of given 3 vectors
- Find equation of line and plane by vector method
- Classify the conic and find cylindrical and spherical coordinates
- Find tangent vectors, curvature of a given curve and its normal vector
- Evaluate torsion and binormal vector to given vector
- Examine the continuity of f(x, y)
- Find partial derivative of a function of 2 or 3 variables
- Find directional derivative of a vector and find tangent planes
- Evaluate extrema using Lagrange multipliers.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	40	20
	Assignment	15		
	Seminar/project/presentation	20		
	Practical/lab examination	-	-	-
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

- David C. Lay, Linear Algebra and Its Applications (4th Edition), Addison Wesley, Pearson.
- Thomas, G. & Finney, R. Calculus and Analytical Geometry. New Delhi, Narosa Publishing House.
- Kreyszig, E. Advanced Engineering Mathematics, New Delhi, John Wiley and Sons Inc.

English Communication

Course Code	BAI 102	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	2 hrs/wk
		Practical	0
		Total	80 hrs

1. Course Description

This course is designed to equip students with a wide range of techniques for effectively utilizing English in both academic and communicative contexts.

2. Course Objectives

The course aims to achieve the following objectives:

- Acquaint students with necessary grammar and use them to construct sensible sentences.
- Enable students to answer the questions given after the comprehension exercises.
- Familiarize students with the various genres of literature and analyze them.
- Enable students with the process of writing.
- Enable students to produce different texts and genres of writing.
- Develop students' ability to speak fluently and interactively.
- Enable students to produce professional writing, including proposals and project reports.

3. Course Details

3.1 Theory (48 hrs)

Unit 1: Reading Comprehension (12 hrs)

"The Bet" by Anton Chekov (Story); "The Three Dancing Goats" by an anonymous writer (Story); "The Gift of the Magi" by O'Henry (Story); "The Road Not Taken" by Robert Frost (Poem); "Letter from Foreign Grave" by D. B. Gurung (Poem); "College Teachers" by R.L.S. Malky and Mary K Ruether, Eds. (Essay); "Humility" by Noah Harari (Essay); "I Want a Wife" by Judy Brady (Essay); "9 Thirty-Eight Who Saw the Murder Didn't Call the Police" by Martin Gangsberg (Essay); "Connotation and types of Innovation" by Jin Chen & Ximing Yin (Essay).

Unit 2: Sentence Construction and Mechanics in Writing (6 hrs)

Use of tenses; active and passive voice; subject-verb agreement; conditional sentences; prepositions; mechanics in writing: punctuations, spelling, capitalization, italics, abbreviations, etc.

Unit 3: Writing Skill (20 hrs)

General writing: job application and CV, greeting card and invitation letter, email and SMS, brochures, posters, blogs, business letter, notice, minute, memoranda, essay; technical writing: proposal writing/project writing; types of formal reports: progress report, feasibility report, empirical report, research report, technical report; report writing: memo report, letter report, project/field report.

Unit 4: Oral Communication (10 hrs)

Greeting and introducing; giving instruction/direction; describing; thanking and accepting an apology; expressing individual views; group/round table discussions; conducting meetings; interview (techniques, body language, dos and don'ts); talk delivery; presenting a brief report.

3.2 Tutorials

Technical writing assignments; presentation assignments; peer review exercises; language practice activities; case studies analysis; technical document editing; cross-cultural communication exercises; interactive online exercises; collaborative writing projects; industry-specific communication tasks.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	40	20
	Assignment	15		
	Seminar/project/presentation	20		
	Practical/lab examination	-	-	-
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

5.1. Text Book

- Matreyek, W. (1983). Communicating in English. London: Pergamon Press.
- Davis, F. & Rimmer, W. (2013). Active Grammar 1. Cambridge: Cambridge University Press.
- Thompson, A.J. & Martinet, A. V. (1986). A Practical English grammar. London: Oxford University Press.
- Pilai, S. (2016). Spoken English for My World. Oxford: Oxford University Press.

5.2. Reference Book

- Futherford, A. J. (2001). Basic Communication Skills for Technology. Pearson: Pearson Education.
- Mandel, S. R. & Kirszner, L. G. (eds.)(2009). Patterns for College Writing. NewYork: St. Martin's.

Physics of Computing

Course Code	BAI103	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This Course has been designed keeping in view with recent development in computational physics. It combines computer science, physics and applied mathematics to develop scientific solutions of complex problems. It complements the area of theory and experiments in scientific investigations. It can be applied in molecular modelling, electronic circuit design, protein folding, atmospheric science, aerodynamic design and testing and material science. Availability of additional specialties in areas and upgraded quality of course will relate physics with CS technology. Various topics are incorporated in the syllabus to relate physics and computer science along with information technology.

2. Course Objectives

The course is designed with following objectives:

- Provide students up to date knowledge of fundamentals of computing science.
- Impart knowledge of basic view of physical principles of theoretical and experimental aspects of computers.
- Acquaint students with the recent trends in physics and electronics.
- Develop manpower in computer engineering at the tertiary level to conduct research in engineering on the basis of physics and electronics.
- Produce high level research-oriented man power in physics so that they can be able to tackle high tech IT industry.

3. Course Details

3.1. Theory (48 hrs)

Unit 1: Semiconductor and Semiconductor Devices (8 hrs)

- Intrinsic and Extrinsic Semiconductors
- Electrical conductivity of semiconductors
- Photoconductivity, Metal-Metal Junction, Contact Potential, Semiconductor Diode, Bipolar Junction Transistor (BJT), Field Effect Transistor

Unit 2: Universal Gates and Physics of Integrated Circuits (10 hrs)

- Universal Gates, RTL and TTL Gates
- Clock Circuits
- Semiconductor Purification, Zone Refining, Single Crystal Growth
- Process of IC Production, Electronic Component Fabrication on a Chip,
- CMOS Inverter
- Static Gate Characteristics
- Delay: Transistor models, RC models for delay, Drive and loads
- Power and Energy
- Scaling Theory
- Reliability

Unit 3: Sequential Machines (5 hrs)

- Introduction

- Combinational Logic: Event model, network model, Gain and reliability, Gain and delay, Delay and power
- Interconnect: Parasitic impedance, Transmission lines, Crosstalk, Wiring complexity and Rent's Rule
- Sequential Machines: Sequential modes, Registers, Clocking, Metastability

Unit 4: Processors and Systems **(10 hrs)**

- Introduction
- System Reliability
- Processors: Microprocessor characteristics, Buses and interconnect, Global communication, Clocking
- Memory: Memory structures, Memory system performance, DRAM systems, DRAM reliability
- Mass Storage: Magnetic disk drive, Flash memory, Storage and performance
- System Power Consumption: Server system, Mobile system and batteries, Power management
- Heat Transfer: Heat transfer characteristics, Heat transfer modeling, Heat and reliability, Thermal management

Unit 5: Input and Output **(5 hrs)**

- Introduction
- Displays
- Image Sensors
- Touch Sensors
- Microphones
- Accelerometers and Inertial Sensors

Unit 6: Emerging Technologies **(10 hrs)**

- Carbon Nanotubes: Nanotube transistors
- Quantum Computer
- Machine Learning
- Deep Learning
- Reinforcement Learning
- Robotics
- Natural Language Processing
- Computer Vision
- Recommender System

3.2 Tutorials

Unit-specific exercises focused on problem-solving.

3.3 Laboratory Work

The laboratory work includes the following concepts:

- Fabrication of Multi-Vibrator
- Touch Sensors
- Machine Learning
- Robotics
- Computer Vision

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	20	10
	Assignment	10		
	Seminar/project/presentation	5		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

- Marilyn Wolf (2016), The Physics of Computing (1st Edition), Morgan Kaufmann

Foundation of Information Technology

Course Code	BAI104	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course covers the basic concepts of computers and information technology including introduction, hardware, software, memory, input/output, data representation, database, networks and data communication, Internet, multimedia, and computer security.

2. Course Objectives

The course aims to achieve the following objectives:

- Identify the components of a computer system
- Familiarize with hardware/software components
- Knowledge of database management systems
- Internet, networking and mobile computing and internet security.
- Apply different tools and technique to create documents using office package software
- Use of CUI and GUI Interface to operate computer applications
- Explore concept of emerging technologies

3. Course Details

3.1 Theory (48 hrs)

Unit 1: Introduction to Computer System (3 hrs)

- Introduction to Computer
- Characteristics of Computer
- History of computer
- Generation of Computer
- Classification of Computer: Size, Work, Brand
- Anatomy of a Digital Computer
- Scope and Applications of Computer

Unit 2: Computer Hardware (6 hrs)

- Introduction to hardware
- Central Processing Unit: Definition and Role of CPU, ALU, CU, Register and their functions
- Memory Unit: Memory Hierarchy, memory representation
- Memory & Its Classification: Primary memory (RAM, ROM), Secondary memory (Magnetic Disk, Optical Disk, Magnetic Tape)
- Instruction Format: Definition, Components
- Instruction Set: Definition, Types of Instruction Set (CISC, RISC)
- Instruction Cycle: Fetch, Decode, Execute, Role of Program Counter and Instruction Register
- Interconnecting the Units of a Computer: Bus Architecture, Address Bus, Data Bus and Control Bus
- Inside the computer cabinet
- Microprocessor: Definition and Function
- Input devices: Definition and Types of Devices

- Output Devices: Definition and Types of Devices

Unit 3 Computer Software **(5 hrs)**

- Introduction to computer software
- Categories of Software
- System software: Introduction, Types (Operating System: Introduction and Objectives of Operating System, Types of Operating System, Functions of Operating System), Diver Driver, Firmware, Utility Software, Translator
- Application Software: Introduction, Types (General Purpose, Customized Software), Applications
- Software Acquisition

Unit 4: Data Representation **(4 hrs)**

- Introduction to Number System and their conversion
- Binary Arithmetic: Addition, Subtraction, Multiplication, Division
- Complement Methods: 1's Complement, 2's Complement
- Binary Data Representation: Signed and Unsigned Numbers
- Binary Coding Schemes: BCD, Gray Code, Excess-3, ASCII
- Logic Gates: OR, AND, NOT, NAND, NOR, XOR, XNOR

Unit 5: Fundamentals of Database Management System **(5 hrs)**

- Introduction to DBMS: Definition and Purpose, Key Components, Advantage of DBMS Over Flat File System, Applications of DBMS
- Database Models: Relational Database Model, Hierarchical Database Model, Network Database Model, Object-oriented Database Model, NoSQL Database Model
- Database Language: DDL, DML, DTL, DCL
- Data Security: Definition and its importance
- Data Warehouse: Definition and Purpose, Key Features
- Data Mining: Definition and Objectives, Steps, Application
- Database Users

Unit 6: Data Communication and Computer Network **(6 hrs)**

- Introduction to communication system
- Mode of Communication: Simplex, Half Duplex, Full Duplex
- Computer Network: Introduction, Importance, Basic Components
- Network Types: PAN, LAN, MAN, WAN
- Network Topologies: Bus, Ring, Star, Tree, Mesh, Hybrid
- Transmission Media: Guided Media (Twisted-Pair, Co-axial, Fiber Optics), Unguided (Microwaves, Radiowaves, Infrared)
- Network Devices: Modem, Router, Repeater, Hub, Switch, Bridge, Gateway, NIC
- TCP/IP & OSI Reference Model
- Communication Protocols: HTTP, TCP/IP, FTP, SMTP, Telnet, Gopher, POP3
- Centralized vs. Distributed System,
- Wireless Networking (Wireless PAN, Wireless LAN, Wireless MAN, Wireless WAN)

Unit 7: The Internet and Internet Services **(6 hrs)**

- Introduction: Overview, History of Internet, Advantage and Disadvantage
- Internetworking Protocol: TCP/IP
- The Internet Architecture: Client, ISP, Regional ISP, and Backbone
- Managing the Internet: Governing bodies of the Internet

- Connecting to Internet: How to connect to the Internet?
- Internet Connections: Dial-up Connection, Broadband Connection, DSL, Cable Modem, ISDN, Cellular, Wireless connection
- Internet Address: IP address and domain names
- WWW and Its Evolution
- Internet Services: Communication Services, Information Retrieval Services, Web Service, WWW
- Uses of Internet: Different uses of Internet
- Architecture of Web
- Uniform Resource Locator (URL), Browsers and It's Feature, Search Engine
- Web Servers: Apache, IIS, Proxy Server

Unit 8: Computer Security **(5 hrs)**

- Introduction: Introduction to computer security
- Security Threat and Security Attack: What is security threat? Security attack and its types.
- Malicious Software: Virus, worm, Trojan horse
- Security Services: Confidentiality, integrity, authentication, nonrepudiation
- Security Mechanism: Cryptography, Digital Signature, Firewall, User Identification and Authentication, Intrusion Detection Systems
- Cyber Security: Introduction, Types, Importance
- Security Management: Overview and Importance

Unit 9: Emerging Technologies **(8 hrs)**

- Cloud Computing: Definition and Characteristics, Cloud Service Model (IaaS, PaaS, SaaS), Deployment model (Public, Private, Community, Hybrid)
- Mobile Computing: Introduction and its Concept, Application
- Quantum computing: Introduction, Quantum Bits and Applications
- E-governance: Definition and objectives, Types, Advantage and Disadvantage
- E-commerce: Introduction, Types, Pros and Cons
- E-Learning: Introduction: Introduction, methods
- Ubiquitous Learning: Definition and characteristics, Challenges and Opportunity
- E-Banking: Online Banking, Services and security measures
- Mobile Banking: Services and Application, Security consideration
- Crypto Currency: Introduction, working Mechanism, Risk
- GIS: Basis of GIS, Application
- Artificial Intelligence: overview, Applications ethics and bias
- Machine Learning: Fundamental, Types, Features
- Deep learning: Introduction, application
- Robotics: Introduction, Types, Applications
- Virtual Reality, Augmented Reality, Ambient Reality: Introduction, Future Trends, Applications
- Internet of Things: Introduction, Architecture and Components
- Blockchain Technology: Overview, Applications

3.2 Laboratory Work

After completing this course, students should have practical knowledge of different hardware components of computer, operating systems, Word Processors, Spreadsheets, Presentation Graphics, Database Management Systems, and Internet and its services. The laboratory work includes:

(1) Basic of Computer and Network

- Basic building block of PC, hardware and peripheral introduction
- Assemble and disassemble of computer
- Learning about network cable and crimping different cable types

(2) Learn CUI and GUI Based Operating System

- Install different types of operating system
- Learning command line interface such as Command line/terminal
- Learn MS-DOS Internal command and External command

(3) Learn word processing software

- Concept of word processing, Introduction of MS Word, Features, Keyboard Shortcuts, Editing & Formatting documents, Table, Advance features of MS Word, Mail Merge, Macros

(4) Learn spreadsheets software

- Worksheet basic, working with formulas and cell referencing, Auto sum, Formatting to worksheet, Previewing and Printing worksheet, Graphs and Charts, Functions, Macros

(5) Learn presentation software

- Introduction, Use of Standard Formatting toolbar, drawing toolbar, editing slides, changing templates, slide layouts, inserting clipart & pictures into slide, slide transitions, animation, inserting sound and movies into slides

(6) Email, Internet and online tools for office management

- Learn different email system, configuring email into personal computer, using internet-based tools for office management

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	20	10
	Assignment	10		
	Seminar/project/presentation	5		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

5.1 Text Books

- A. Goel, "Computer Fundamentals." Pearson Education India.
- P. Norton, "Introduction to Computers." McGraw-Hill Education.

5.2. Reference Books

- E. Balagurusamy, "Computing Fundamentals." McGraw-Hill Education.
- P. Kumar, A. Tomar, R. Sharmila, "Emerging Technologies in Computing: Theory, Practice, and Advances." CRC Press.

C Programming

Course Code	BAI105	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course combines the principles of computational thinking with hands-on C programming to introduce students to the world of computer science and programming. Students will learn howto approach and solve problems using computational thinking concepts and apply them to implement solutions. Through hands-on programming exercises and projects, students will gain a solid foundation in both computational concepts and C programming.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce principles of computational thinking and their relevance in problem-solving
- Familiarize students with the basic syntax, data types, and control structures
- Develop students' skills in algorithm design, implementation, and analysis
- Foster logical reasoning and systematic problem-solving approaches
- Provide practical experience in developing small-scale C programs.

3. Course Details

3.1 Lecture (48 hrs)

Unit 1: Introduction to Computational Thinking (2 hrs)

Overview of computational thinking and its significance in problem-solving. Problem- solving strategies and abstraction. Algorithm design and analysis.

Unit 2: Introduction to Programming and C Programming (4 hrs)

Types of programming and language translator, algorithm and flowchart. Overview of C and its historical significance, syntax and semantics. Installing and setting up the development environment (IDE or text editor).

Unit 3: Basic Syntax and Data Types (8 hrs)

Structure of a C program: header files, main function, and comments. Data types: int, float, char, and their modifiers. C tokens and character sets. Data types, constants and variables. Expression, statements and comments. Escape sequences and delimiters I/O functions. Formatted I/O operators: arithmetic operators, relational operators; logical operators; assignment operators, increment and decrement operators; ternary operator, bitwise operator; other operators (comma, sizeof). Expression evaluation. Operator precedence and associativity, type conversions, preprocessor directives: macros and their usage.

Unit 4: Control Statements (8 hrs)

Decision-making structures: if, if-else, nested if-else. Switch statement. Looping structures: while, for, do-while. Nesting control structures for complex conditions. Jump statements: break, continue, goto.

Unit 5: Arrays and Strings (6 hrs)

Declaring and initializing arrays. Accessing array elements and multi-dimensional arrays. Strings and string manipulation functions. Pointers and Array, array of pointers. Strings and string

manipulation functions (strlen, strcpy, strcat, strcmp).

Unit 6: Functions **(6 hrs)**

Creating and calling functions. Communication between functions, passing arguments, using library functions. Recursive functions.

Unit 7: Pointers **(6 hrs)**

Introduction to pointers and memory addresses. Pointer arithmetic and dynamic memory allocation (malloc, calloc, realloc, free). Passing arguments to functions: call by value and call by reference. Pointers and arrays.

Unit 8: Structures and Unions **(3 hrs)**

Defining and using structures. Nested structures and arrays of structures. Understanding unions and their uses.

Unit 9: File Handling **(5 hrs)**

File handling concepts: opening, reading, writing, and closing files. File opening modes. Sequential and random access file operations. Handling errors during file operations.

3.2 Practical/ Lab

- Demonstrate concept of syntax, data type, variable, constant and comment
- Demonstrate concept of I/O
- Demonstrate arithmetic operations
- Demonstrate concept of operators
- Demonstrate concept of type conversion
- Use preprocessor directives: include, macros
- Demonstrate concept of decision making structure: if, if-else, if-else ladder, nested if-else
- Demonstrate concept of switch statement
- Demonstrate concept of loop: for, while, do-while and their nested structure
- Demonstrate concept to jump statements: break, continue, goto
- Demonstrate concept of array: single and multi-dimension
- Demonstrate the concept of array of string
- Demonstrate the use of string manipulation functions
- Create and call the function
- Demonstrate concept of return type and non-return type function
- Demonstrate concept of passing argument by function: call by value and call by reference
- Use recursive function
- Create and use pointer
- Demonstrate pointer arithmetic
- Demonstrate dynamic memory allocation: malloc, calloc, realloc, free
- Use of pointer in array
- Define and user of structure
- Demonstrate the concept of nested structure
- Demonstrate the concept of array of structure
- Define and use of union
- Demonstrate file handling operations: opening, reading, writing, closing
- Demonstrate sequential file access operation
- Demonstrate random file access operation
- Handle error in file handling operation.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) internal evaluation and (2) semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full	Pass

			Marks	Marks
Internal evaluation	Class attendance and performance	5	20	10
	Assignment	10		
	Seminar/project/presentation	5		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

- Y. P. Kanetkar, "Let Us C". BPB Publications.
- B. Gottfried, "Programming with C". McGraw Hill Education.
- B. Kernighan and D. Ritchie, "The C Programming Language". Pearson.

Year I Semester II

Mathematics II

Course Code	BAI151	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	0
		Total	64 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding basic concept and its uses to solving the problems of infinite series, linear programming problem, ordinary differential equations, multiple integrals, matrix and determinant and its application, vector space, orthogonality and least orthogonality.

2. Course Objectives

The course aims to achieve the following objectives:

- Students will be able to understand and formulate real world problems into mathematical statements.
- Students will be able to develop solutions to mathematical problems at the level appropriate to the course.
- Students will be able to describe or demonstrate mathematical solutions.

3. Course Details

3.1 Theory **(48 hrs)**

Unit 1: Infinite Series **(6hrs.)**

Convergence and divergence of infinite sequence, Convergence and divergence of infinite series, Necessary condition of convergence of infinite series, Geometric series, Hyper-harmonic series, Alternating series, Test for convergence of an infinite series (Without proof) (p test, n^{th} term test, Integral test, Comparison test, Ratio test, Root test, Leibnitz test), Absolute and conditional convergence, Power series, Interval and radius of convergence.

Unit 2: Linear Programming Problem **(6hrs.)**

Model formulation, Standard form, Simplex method, Simplex method: difficulties, Dual simplex method.

Unit 3: Ordinary Differential Equations **(8hrs.)**

First order differential equations, Review of separable and exact differential equation, First-order linear and Bernoulli's differential equations and its solution, Second order homogeneous and non-homogeneous linear differential equations, Solution of homogeneous equation with constant coefficient by auxiliary equation, Solution of non-homogeneous equation with constant coefficient by undetermined coefficients for polynomial, exponential and trigonometric right hand side functions.

Unit 4: Multiple Integrals **(6hrs.)**

Double integrals, Double integrals in polar coordinates, Fubini's theorems (without proof), Change of order of integration.

Unit 5: Matrix and Determinant **(6hrs.)**

Matrix, determinant, matrix transformation, Rank of a matrix, linear system of equations, Solution of the linear system of equations by Gauss elimination, eigenvalues and eigenvectors, eigenspace, Diagonalization

Unit 6: Vector Space **(4hrs.)**

Vector space and subspace, linear independence, basis of a vector space, linear transformation, orthogonal matrix and transformation by orthogonal matrix

Unit 7: Application of Matrix **(6hrs.)**

Block matrix, Partitioned Matrices, The Leontief Input Output Model, Applications to Computer Graphics (shear transformation, homogeneous coordinates in 2D & 3D, projection maps)

Unit 8: Orthogonality and Least Orthogonality **(6 hrs.)**

Inner Product, Length and Orthogonality, Orthogonal Sets, orthonormal basis, Orthogonal Projections, The Gram-Schmidt Process, Least-Squares Problems, Applications to Linear Models, least squares fitting of curves

3.2 List of Tutorials

The following tutorial activities shall be conducted to cover all the required contents of this course. This will enable the students to complete the related mathematical problems under the supervision of the subject teacher.

- To test for convergence of a series by different tests
- Finding centre of convergence, radius of convergence and interval of convergence
- Simplex method for standard problems and duality in LPP
- Simplex method for mixed inequalities (Big-M method)
- Solution of separable, homogeneous and exact differential equation Linear and Bernoulli equation
- Solve second order homogeneous ODE with constant and variable coefficients, Euler-Cauchy equation
- Solve non-homogeneous second order ODE by undetermined coefficients for polynomial, exponential and trigonometric right hand side functions
- Evaluating double integral and problems with change of order of integration
- Determining rank of a matrix and test the consistency then solve the linear equations
- Problems on solution of system of linear equations by Gauss elimination
- Finding eigenvalues, eigenvectors of a matrix and diagonalization of a matrix
- Problems on vector space and subspace
- Problems on linear dependence and independence, linear transformation
- Finding basis, test the orthogonal matrix
- Finding block matrix and partitioned matrix
- Develop and evaluate Leontief Input Output Model
- Problems based on matrix transformation - shear transformation, homogeneous coordinates in 2D & 3D, projection maps
- Evaluate inner product of vectors, testing orthogonality and finding orthogonal sets, orthonormal basis, orthogonal projections
- Find Least-Squares solution and using it finding least squares fitting of curves.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	40	20
	Assignment	15		
	Seminar/project/presentation	20		
	Practical/lab examination	-	-	-
	Internal examination	20	20	10

	Total of internal evaluation	**60**	**30**
Semester examination	**Total of semester examination**	**40**	**20**

5. Books

- James Stewart, Calculus Early Transcendentals, 7E, CENGAGE Learning.
- Kreyszig, E. Advanced Engineering Mathematics. New Delhi: John Wiley and Sons Inc.
- David C. Lay, Linear Algebra and Its Applications, 4th Edition, Addison Wesley, Pearson.

Discrete Mathematics

Course Code	BAI152	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

Discrete mathematics is a course that studies mathematical structures that are discrete or distinct from one another in contrast to Calculus which studies continuous mathematical structures. This course contains the study of discrete structures that arise frequently in various fields of computer science such as artificial intelligence, cryptography, network security, data structure and algorithm analysis etc.

2. Course Objectives

The objective of this course is to

- Give fundamental concepts of propositional and predicate logic and introduce the students to the idea of mathematical proof.
- Provide the basic concepts of number theory that will be useful for further study.
- Introduce the students to the topic of recursion and proofs by using the principle of mathematical induction.
- Provide basic tools for counting ordered and unordered arrangements and advanced tools of counting using recurrence relations.
- Introduce the students to graphs and trees and their various applications and algorithms.
- Give some basic concepts about relations and their various types, representation as well as partial ordering relations.

3. Course Details

3.1 Lecture (48hrs)

Unit 1: Logic and Proofs (8 hrs.)

Propositional Logic: Introduction, Propositions, Conditional Statements, Truth Tables of Compound Propositions, Precedence of Logical Operators, Translating English Sentences, System Specification.
Propositional Equivalences: Introduction, Logical Equivalences, Using De Morgan's Laws, Constructing New Logical Equivalences, Satisfiability, Solving Satisfiability Problems.
Predicates and Quantifiers: Introduction, Predicates, Quantifiers, Quantifiers over Finite Domains, Quantifiers with Restricted Domains, Precedence of Quantifiers, Binding Variables, Negating Quantified Expressions, Translating from English, Nested Quantifiers and its Order, Negating Nested Quantifiers, Translating English Sentences.
Rules of Inference: Introduction, Valid Arguments, Rules of Inference for Propositional Logic and its use to Build Arguments, Resolution, Fallacies, Rules of Inference for Quantified Statements and its Use.
Proofs and Proof Methods: Introduction, Methods of Proving Theorems, Direct Proofs, Proof by Contraposition, Vacuous and Trivial Proof, Proofs by Contradiction, Proofs of Equivalence, Counterexamples, Exhaustive Proof, Proof by Cases, Existence Proofs, Mistakes in Proofs.

Unit 2: Number Theory (9 hrs.)

Divisibility and Modular Arithmetic: Division, The Division Algorithm, Modular Arithmetic, Arithmetic Modulo m, Binary Representation of Integers, Modular Exponentiation.
Primes and Greatest Common Divisors: Primes, The Infinitude of Primes, GCD and LCM, The Euclidean Algorithm, gcd as Linear Combinations.

Solving Congruences: Linear Congruences, The Chinese Remainder Theorem, Fermat's Little Theorem, Pseudoprimes, Primitive Roots and Discrete Logarithms.

Unit 3: Induction and Recursion **(5 hrs.)**

Mathematical Induction: Introduction, Mathematical Induction, Examples (Proving Summation, Proving Inequalities, Proving Divisibility), Strong Induction, Well-Ordering Property
Recursive Definitions and Structural Induction: Introduction, Recursively Defined Functions, Recursively Defined Sets, Structural Induction
Recursive Algorithms: Recursive Algorithms, Proving Recursive Algorithms Correct, Recursion and Iteration

Unit 4: Counting and Recurrence Relations **(8hrs.)**

Basics of Counting and Pigeonhole Principle: Basic Counting Principles, Inclusion-Exclusion Principle, Pigeonhole Principle and Generalized Pigeonhole Principle.
Permutations and Combinations: Permutations, Combinations, Permutations and Combinations with Repetitions, Permutation with Indistinguishable Objects.
Binomial Coefficients and Identities: The Binomial Theorem, Pascal's Identity and Pascal's Triangle.
Recurrence Relations: Introduction, Modeling with Recurrence Relations, Solving Linear Homogeneous and Non-homogeneous Recurrence Relations using Characteristic Equations, Solving by Substitution.

Unit 5: Graphs and Trees **(12 hrs.)**

Graph: Graph Model, Basic Terminology, Special Simple Graphs, Bipartite Graphs, Graph Representation (Adjacency List, Adjacency Matrix, Incidence Matrix), Graph Isomorphism and its Determination
Graph Connectivity: Paths, Connectedness in Undirected Graphs, Vertex and Edge Connectivity, Connectedness in Directed Graphs, Paths and Isomorphism, Counting Paths between Vertices.
Euler and Hamiltonian Paths: Euler Paths and Circuits, Necessary and Sufficient Condition, Hamiltonian Paths and Circuits, Dirac's Theorem, Ore's Theorem.
Shortest-Path Problems: Introduction, Shortest-Path Algorithm, Traveling Salesman Problem.
Graph Coloring: Introduction, Applications of Graph Coloring.
Trees: Introduction, Rooted Trees, Properties of Trees, Binary Search Trees, Decision Trees.
Tree Traversals: Introduction, Traversal Algorithms, Infix, Prefix and Postfix Notation.
Spanning Trees: Introduction, Construction Methods (Depth-First Search and Breadth-First Search), Minimum Spanning Trees, Kruskal's Algorithm, Prim's Algorithm.

Unit 6: Relations **(6 hrs.)**

Matrices: Boolean Matrices, Boolean Matrix Operations (Complement, Join, Meet, Boolean Product).
Relations: Relations on a Set, Properties of Relations (Reflexive, Symmetric, Antisymmetric, Transitive), Combining Relations, n-ary Relations Introduction, Matrix and Digraph Representation of Relations, Reflexive, Symmetric and Transitive Closure of Relations, Equivalence Relations, Equivalence Classes, Equivalence Classes and Partitions, Partial Ordering Relation, Hasse Diagram, Maximal and Minimal Elements, Lattices.

3.2 List of Tutorials

- Propositional Logic, Propositions, Quantifiers, Rule of Inference.
- Direct, Indirect and Exhaustive Proof.
- LCM, GCD, Chinese Remainder Theorem.
- Permutation and Combination in different conditions, Pigeonhole Principle, Binomial Theorem, Pascal's Triangle.
- Mathematical Induction, Recursive and Iteration, Recursively Defined Functions.
- Different Types of Graph, Graph Representation, Isomorphism, Euler and Hamiltonian Path and Circuit, Dijkstra's Algorithm.
- Tree Traversal, Spanning Trees, Kruskal's Algorithm, Prim's Algorithm.

- Types and Properties of Relations, Equivalence Relations, Maximal and Minimal Elements, Lattices.

3.3 Laboratory Works

The laboratory work consists of implementing the algorithms and concepts discussed in the class. Students should implement problems with following concepts,

- Logical operators, truth table of tautology, contrapositive and contingency.
- Number Theory Algorithm, Operations on Integers.
- Counting and Some Recursive Algorithms.
- Algorithms for Graph and Trees.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	20	10
	Assignment	10		
	Seminar/project/presentation	5		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

- Rosen, K. H. Discrete Mathematics and its Applications, 8th Edition, McGraw Hill Education.
- Johnsonbaugh R. Discrete Mathematics, 8th Edition, Pearson Education.
- Kolman, B., Busby, R. and Ross, S. Discrete Mathematical Structures, 6th Edition, Pearson Education.
- Mott, J. L., Kandel, A. and Baker, T. P. Discrete Mathematics for Computer Scientists and Mathematicians, 2nd Edition, Pearson Education.
- Burton, D. Elementary Number Theory, 6th Edition, McGraw Hill Education.
- Liu, C. L. Elements of Discrete Mathematics, 2nd Edition, McGraw Hill Education.
- Deo, N. Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall India.

Operating System

Course Code	BAI153	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive insight to the operating systems principle, design, implementation and applications.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce Operating System, its functions, types and structure.
- Identify process management mechanisms.
- Familiarize with memory management techniques.
- Introduce file and directory structure.
- Insight IO management theory and practices.
- Identify Security management activities.

3. Course Details

3.1 Theory (48 hrs)

Unit 1: Introduction (8 hrs.)

Operating System Concepts, Functions of Operating System, Types of Operating System, System Call, Operating System Structure, Virtual Machine.

Unit 2: Process Management (12 hrs.)

Processes, Threads, Synchronization and Inter-Process Communication, Process Scheduling. deadlock characterization, handling deadlocks, deadlock prevention, avoidance, detection, recovery.

Unit 3: Memory Management (10 hrs.)

Memory Management: logical and physical addresses, swapping, contiguous allocation, paging, segmentations.
Virtual memory: demand paging, the performance of demand paging, page replacement algorithms, allocation of frames, thrashing.

Unit 4: File System (6 hrs.)

File operations, Access Methods, Directories and Levels, Directories Operations, file system mounting and sharing, protection, access control, File system layout, File system implementation.

Unit 5: Input/Output Management (8 hrs.)

I/O devices, Devices Controller, Memory Mapped I/O, Direct Memory Access (DMA), I/O software Principles: programmed I/O, Interrupt driven I/O, DMA based I/O, I/O Software Layers
Disk management: Disk structure, Disk scheduling algorithms.

Unit 6: Security (4 hrs.)

Fundamentals of Operating System Security, Controlling Access to Resources, Formal Models of Secure System, Attacks.

3.2 List of Tutorials

- Operating System Services (Class discussion)
- New Trends in Operating Systems (Oral Presentation).

- Evaluating the performance of different Process Scheduling Algorithm, Disk Scheduling Algorithm and Page Replacement Algorithm through hypothetical scenarios.
- WAP that creates multiple processes and manages their execution using different scheduling algorithms such as Round Robin, Priority Scheduling, and Shortest Job First.
- Implementing page replacement algorithms such as FIFO, LRU, and Clock, and evaluating their performance in a simulated virtual memory environment.
- WAP that simulates FCFS, SCAN and CSCAN disk scheduling algorithms.

3.3 Laboratory Works

- Simulate the process scheduling algorithms.
- Simulate the page replacement algorithms.
- Observe Disk Scheduling algorithms.
- Case Study: Unix, Linux and Android.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	20	10
	Assignment	10		
	Seminar/project/presentation	5		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

- Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, Pearson.
- Silberschatz, Galvin and Gagne, Operating System Concepts, Addition Wesley.
- P. Pal Choudhury, Operating System Principle and Design, PHI
- D. M. Dhamdhere, System Programming and Operating System, Tata McGraw-Hill.

OOP in Java

Course Code	BAI154	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding of object-oriented programming concepts using Java programming language. The course covers different aspects of programming, including fundamental concepts, different programming structures, classes and objects, inheritance, interface, package and modules, exception handling and some essential Java classes. Through a combination of lectures, practical exercises, and projects, students will gain the necessary skills to write computer programs using Java programming language.

2. Course Objectives

The course aims to achieve the following objectives:

- To introduce different concepts of object-oriented programming.
- To Provide concepts and skills of Java development environment and fundamental programming structures in Java.
- To familiarize students with concepts classes, objects, inheritance, and packages and modules and teach them how to use these concepts to write Java programs.
- To provide knowledge of exception handling and some essential Java classes.
- To teach students how to implement exception handling concepts and essential Java classes.

3. Course Details

3.1 Theory (48 hrs)

Unit 1: Introduction **(6 Hrs.)**

Java Introduction; Short History of Java; Java Buzzwords; Java in Web Development, Mobile Application Development, and Enterprise Application Development; Java Virtual Machine (JVM); Java Runtime Environment (JRE); Bytecode; Writing Simple Java Programs; Compiling and Running Java Programs using Command Line and IDE; Using Command Line Arguments; Using Scanner for Reading Input and System.out.print() for Writing Output

Unit 2: Fundamental Programming Structures **(12 Hrs.)**

Writing Comments; Primitive Data Types; Variables and Constants; Type Conversion and Casting; Operators (Arithmetic, Bitwise, Logical, Assignment, and Conditional); Precedence and Associativity of Operators; Control Statements (if, switch, for, while, do-while, for-each, nested statements, break, continue, and return); Working with Big Numbers; Arrays (One Dimensional and Multidimensional)

Unit 3: Classes and Objects **(10 Hrs.)**

Object Oriented Principles; Defining Classes; Adding Variables and Methods; Creating Objects and Accessing Class Members; Method Parameters and Return Types; Constructors; static Fields and Methods; Method Overloading; this Keyword; Access Control; Nested and Inner Classes; Recursive Methods; Garbage Collection

Unit 4: Inheritance and Interface **(6 Hrs.)**

Inheritance Basics; Defining Subclasses; Using super and final Keywords; Abstract Class; The Object Class; Dynamic Method Dispatch; Declaring, Extending, and Implementing Interfaces

Unit 5: Exception Handling **(5 Hrs.)**

Exception Handling Fundamentals; Exception Types; Uncaught Exceptions; Using try, catch, throw, throws, and finally; Java's Built-in Exceptions; Generics

Unit 6: Packages and Modules **(4 Hrs.)**

Built-in Packages; Creating and Using Packages; Modules

Unit 7: Essential Java Classes **(5 Hrs.)**

String; Primitive Type Wrappers; Math Class; File Input / Output

3.2 Java Programming Comprehensive Tutorials

- Setting Up Your Java Development Environment: A tutorial on installing and configuring the Java Development Kit (JDK) and Integrated Development Environment (IDE) for Java programming.
- Creating Your First Java Program: A step-by-step guide to writing, compiling, and running a simple Java program to understand the basic structure.
- Access Modifiers: Understanding access modifiers like public, private, protected, and default, and their implications on class members.
- Packages and Data Types: Exploring Java packages and different data types, with hands-on exercises to practice variable declarations and data manipulation.
- Working with Classes and Objects: Tutorials covering the creation of classes, instantiation of objects, and implementing constructors and methods.
- Object-Oriented Concepts: Exploring inheritance, polymorphism, encapsulation, and abstraction with practical examples.
- Advanced Inheritance and Interfaces: Tutorials on creating subclasses, implementing interfaces, and understanding multiple inheritance in Java.
- Conditional Statements: Tutorial sessions on using if-else statements, switchcase constructs, and logical operators for conditional programming.
- Exception Handling in Java: Comprehensive tutorials on try-catch blocks, checked vs. unchecked exceptions, and best practices for handling exceptions.
- Working with Files in Java: Step-by-step guidance on reading from and writing to files using Java I/O classes.

3.3 Laboratory Works

Write Java programs to implement all the concepts mentioned in each unit of the syllabus.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	20	10
	Assignment	10		
	Seminar/project/presentation	5		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

- Cay S. Horstmann (2019). Core Java Volume I – Fundamentals, Eleventh Edition, Pearson Education.
- Herbert Schildt (2022). Java: The Complete Reference, Twelfth Edition, McGraw Hill.
- Joshua Bloch (2018). Effective Java, Third Edition, Pearson Education.
- Marc Loy, Patrick Niemeyer, and Daniel Leuck (2020). Learning Java – An Introduction to Real World Programming with Java, O'Reilly, 2020.

Digital Logic

Course Code	BAI155	**Year/Semester**	I/II
Credit Weightage	3	**Lecture**	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding of basic principles of digital logic design, its implementation and applications.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce number system and their conversion
- Familiarize logic gates
- Identify and solve the Boolean algebraic expressions
- Design combinational circuits
- Design sequential circuits
- Design clocked sequential circuits

3. Course Details

3.1 Lecture (48 hrs)

Unit 1: Number Systems and Binary Code (6 hrs)

Positional Number Systems: Binary, Octal, Decimal and Hexadecimal and their conversion; Radix and diminished radix complement method Representation of Signed and Unsigned Number, Floating Point Representation, Binary Codes: Alphanumeric (ASCII, EBCDIC), BCD, Weighted Binary Codes, Non-Weighted Binary Codes (Excess-3 and Gray Codes), Error Detecting Codes: Parity

Unit 2: Boolean Algebra and Logic Gates (8 hrs)

Logic Gates: AND, OR, NOT, NAND, NOR, X-OR, X-NOR, Boolean Algebra: Postulates of Boolean Algebra, Associativity, Inverse, Closure, Commutativity, and Distributives, Basic Theorems and Properties of Boolean Algebra (DeMorgan's Law, Absorption Law, Common Identity Lay), and Principle of Duality, Boolean Variables and Constants, Algebraic Simplification, Canonical and Standard Forms: Canonical SOP, Canonical POS, Standard SOP, Standard POS, Maxterms, Minterms and Conversion between them, Realization of Boolean Expression using NAND and NOR Gate only ,IC Families

Unit 3: Simplification of Boolean Functions (6 hrs)

K-Map Method: 2, 3, and 4 variable K-Maps, The Tabulation Method – Detection and Selection of Prime Implicants, Product of Sum (POS) and Sum of Product (SOP) Simplification using K-Map, Don't Care Conditions.

Unit 4: Combinational Logic Circuits (12 hrs)

Adders: Half Adder, Full Adder; Subtractors: Half Subtractor, Full Subtractor; Binary Parallel Adders (2-bit, 4-bit, N-bit), BCD Adder; Encoders, Decoders and Priority Encoders; Multiplexers (2:1, 4:1, 8:1, 16:1) and De-Multiplexers (1:2, 1:4, 1:8, 1:16); Magnitude Comparator, Parity Generator and Checker; Combinational Circuit Analysis Procedures, Combination Circuit Realization using Universal Gates; Use of Multiplexers and Decoders for Combinational Logic Design; Introduction to ROM, PLA and PAL and their use in Combination Logic Circuits.

Unit 5: Sequential Logic Circuits **(10 hrs)**

Flip-flops: Latch, RS Flip-flop, JK Flip-flop, JK Master-Slave Flip-flop, D Flip-flop and T Flip-flop; Triggering of Flip-flop; Flip-flop Excitation Tables; Asynchronous vs Synchronous Sequential Circuits; Analysis of Sequential Circuit using State Diagrams

Unit 6: Registers and Counters **(6 hrs)**

Registers as basic Memory Blocks; Shift Registers; Ripple (Asynchronous) Counters, Synchronous Counters; Introduction to Memory Units, Memory Address Registers and Memory Buffer Registers.

3.2. List of Tutorials

- Number system conversions
- Binary code conversions
- Circuit design using basic gates
- Realization of circuit with universal gates
- Simplification of Boolean function using K-map
- Combinational circuit design (Magnitude comparator, Parity generator and checker, Combinational circuit analysis)
- Counter design (Asynchronous Up and Down counters, Synchronous Up and Down counters)
- Analysis of sequential circuit

3.3 Laboratory Work

- Implementation of Logic Gates.
- Realization of Universal Gates.
- Converting Boolean Expression into Logic Circuits.
- Implementation of Adder and Subtractor Circuits.
- Implementation of Binary Parallel Adder.
- Design and Implementation of Code Converter Circuits.
- Design and Implementation of Selector (Mux/DeMux) Circuits.
- Design and Implementation of Magnitude Comparator Circuits.
- Design and Implementation of Parity Generator and Checker Circuits.
- Implement PLA and PAL Circuits.
- Implementation of Flip-flops.
- Design and Implementation of Master-Slave Flip-flop.
- Design and Implementation of Shift Registers.
- Design Synchronous Counters (MOD-4, MOD-8, MOD-10).
- Design Synchronous Counters (Binary Counter, UP/Down Counter, BCD).
- Design a Clocked Sequential Circuit.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal assessment and (2) Semester examination with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	20	10
	Assignment	10		
	Seminar/project/presentation	5		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books

- M. Morris Mano (2016), Digital Logic and Computer Design (1st Edition), Pearson
- Thomas L. Floyd (2015), Digital Fundamentals (11th Edition), Pearson
- A. Saha & N. Manna, Digital Principles and Logic Design, Infinity Science Press
- John Patrick Hays, Introduction to Digital Logic Design

Statistics

Course Code	BAI201	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course introduces B. Tech students to the fundamental principles of statistics and its applications in engineering and scientific fields. Students will learn to analyze and interpret data, make informed decisions based on statistical evidence, and apply statistical techniques to solve real-world engineering problems. The course aims to develop students' understanding of statistical concepts and their ability to use statistical tools effectively.

2. Course Objectives

By the end of the course, students should be able to:

- Understand the basic concepts of statistics and its relevance in engineering and scientific research.
- Apply descriptive statistics to summarize and present data effectively.
- Comprehend probability theory and its role in modeling uncertain events.
- Analyze and interpret various types of probability distributions, including discrete and continuous distributions.
- Use sampling distributions to make inferences about population parameters.
- Conduct hypothesis tests to make decisions based on sample data.
- Apply regression analysis to model relationships between variables and make predictions.
- Design and analyze experiments to investigate cause-and-effect relationships.
- Apply statistical software to analyze data, perform calculations, and create visualizations.

3. Course Details

3.1 Theory **(48 hrs)**

Unit 1: Descriptive Statistics: **(10 hrs)**

Introduction to Statistics- descriptive statistics and inferential statistics; types of Data- Qualitative vs. Quantitative data, levels of measurements- nominal level data, ordinal level data, interval level data, and ratio level data and variables.

Presentations of data: Frequency Distribution-Simple frequency distribution, relative frequency distribution, and percentage frequency distribution; Rule of constructing the frequency distribution, data array, stem and leaf display; Graphical presentations; bar char and pie chart, histogram, cumulative frequency distribution and Ogive.

Descriptive measure of data: Quartiles, Deciles, and Percentiles; five number summary, box-and-Whisker plot, shape of the data-skewness and kurtosis; measure of central tendency-Mean, Median, and Mode; Measures of dispersion: Range, Average Deviation Measures, Standard Deviation, Variance, Relative Dispersion and Coefficient of Variation.

डीन

Students also practice and use the statistical software such as SPSS, STATA, R, or MS Excel etc. to calculate mean, standard deveins, skewness, kurtosis, and to visualize the data in graphical form.

Unit 2: Basic Probability: **(5 hrs)**

Introduction to basic probability theory, Approaches to assigning probabilities- classical probability, empirical probability and subjective probability; Terminology in probability, principles of counting- multiplication, permutation and combination; contingency table; Probabilities Rules- Additional and multiplication; Probabilities under conditions of Statistical Independence, Probabilities under conditions of statistical dependence; Bayes' theorem.

Unit 3: Random Variables and mathematical expectation **(3 hrs)**

Random Variables- discrete random variable and continuous random variable, mathematical expectation of random variables, additional and multiplicative rule of mathematical expectations.

Unit 4: Probability Distributions: **(9 hrs)**

Probability distribution function, joint probability function of two variables; Discrete probability distribution- Binomial Probability Distribution, Poisson Probability Distribution; Continuous probability distribution- Normal probability distributions, standard normal probability distribution function, normal distribution as an approximation of Binomial and Poisson probability distribution.

Students practice and use the statistical software such as SPSS, STATA, R, or MS Excel etc. to calculate the probabilities

Unit 5: Sampling Distributions and Estimation: **(7 hrs)**

Introduction to Sampling, reasons for sampling, defining the populations, statistics vs. parameters, sampling methods- probability sampling and non-probability sampling methods; Sampling Distributions of the sample mean and proportions, concept of the Central Limit Theorem, weak law of large number,

Estimations: Introduction of basic terms of inferential statistics such as populations, sample, parameters, significance level, confidence level, critical value, degree of freedom; standard error; method of estimations- point estimations and interval estimations for mean and proportions, determining the sample size for the mean and proportion. Relations between sample size and margin of error.

Students practice and use the statistical software such as SPSS, STATA, R, or MS Excel etc. to calculate the confidence interval and sample size

Unit 6: Hypothesis Testing: **(10 hrs)**

Introduction to hypothesis testing, Null and alternative hypotheses, Type I and Type II errors; t-tests for single mean and difference to two mean, z-tests for single mean and difference to two mean, paired t-test, Chi-Square test of goodness of fit, chi-square test for independence to two attributes. One way analysis of variance for difference of more than two means.

Students practice and use the statistical software such as SPSS, STATA, R, or MS Excel etc. to run hypothesis test the interpret the findings generated from the software to make informed decision.

Unit 7: Simple correlation and regression analysis **(4 hrs)**

Introduction, correlation analysis, regression equation using least squares method, measures of variation - computing the sum of squares; the coefficient of determination, standard error of the estimate; Inference about the population slope, correlation coefficients, and regression equation using t-test and F-test; confidence interval for the slope, estimation of mean values; Limitations and errors.

डीन

Students also practice and use the statistical software such as SPSS, STATA, R, or MS Excel etc. to run correlation and regression analysis and interpret the findings generated from the software to make informed decision.

3.2 Tutorials

Needs tutorial class for practical session for the use of software to analyze the data in unit 1, 5, 6 and 7.

3.3 Laboratory Work

In the practical work, students should be able to use the statistical software such as MS Excel, SPSS, STATA or R to calculate and run the statistical tools such as mean, standard deviations, correlations, regression, hypotheses etc. of the dataset.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	20	20	10
	Total Internal Marks		**60**	**30**
Semester End Examination			**40**	**20**

5. Books

- Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying Ye (2012). Probability and Statistics for Engineers & Scientist. 9th Ed. Printice Hall.
- Michael Baron (2013). Probability and Statistics for Computer Scientists. 2nd Ed., CRC Press, Taylor & Francis Group, A Chapman & Hall Book.
- Douglas C. Montgomery & George C. Runger (2003). Applied Statistics and Probability for Engineers. 3rd Ed. John Willey and Sons, Inc.
- Richard A. Johnson (2001). Probability and Statistics for Engineers. 6th Ed. Pearson Education, India
- Jay L. Devore (2012). Probability and Statistics for Engineering and the Sciences. 8th Ed. Cengage Learning, California Polytechnic State University, San Luis Obispo.

डीन

Database Management System

Course Code	BAI202	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding of database management system concepts. The course covers different aspects of database management including, basics of database, data modelling using ER diagram, relational data model and relational algebra, structured query language, normalization, transaction processing, concurrency control and database recovery techniques. Through a combination of lectures, practical exercises, and projects, students will gain the necessary skills to design and develop databases.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce different concepts of database management systems.
- Provide concepts and skills of designing logical data models
- Familiarize with relational algebra
- Provide concepts of structured query language
- Present concepts of normalization
- Acquaint with database transactions
- Deliver concepts of concurrency control and database recovery techniques

3. Course Details

3.1 Theory **(48 Hrs)**

Unit 1: Foundations of Database **(6 Hrs.)**

Introduction; Characteristics of the Database; DBMS; Actors and Workers in DBMS; Advantages of DBMS; Data Models; Schema and Instance; Three Schema Architecture; Data Abstraction and Independence; Database Language, Database System Environment

Unit 2: Entity-Relationship Model **(8 Hrs.)**

Entity Relationship Model; Entity Types, Entity Sets, Attributes, and Keys; Relationship Types, Relationship Sets, Mapping and Structural Constraints; Weak Entity Types; ER Diagram Notations; Naming Conventions in ER Diagram; ER-to-Relational Mapping; Extended ER Model; Specialization and Generalization; Constraints on Specialization and Generalization

Unit 3: The Relational Data Model and Relational Algebra **(6 Hrs.)**

Relational Model; Domains, Attributes, Tuples and Relations; Relational Model Constraints; Relational Database Schemas; Relational Algebra: Unary Relational Operations: SELECT, PROJECT, RENAME; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; Aggregate Functions; Outer Joins

Unit 4: SQL **(9 Hrs.)**

Data Types; SQL Constraints; DDL Statements; DML Statements: Select, Aliasing, Substring Comparison, Arithmetic Operators, Order By, Null Values, Nested Query, Correlated Nested Query,

डीन

Exists, Set Comparison, Inner Join, Outer Join, Aggregate Functions, Group By Having; INSERT, DELETE, and UPDATE Statements; Views, Stored Procedure, Triggers

Unit 5: Normalization **(6 Hrs.)**

Design Guidelines for Relational Databases; Functional Dependencies; Normalization and its importance; First, Second and Third Normal Forms; Boyce-Codd Normal Form; Multivalued Dependency; Fourth Normal Form; Properties of Relational Decomposition

Unit 6: Concepts of Transaction Processing **(6 Hrs.)**

Introduction to Transaction Processing; Transaction and System Concepts; ACID Properties of Transactions; Characterizing Schedules Based on Recoverability; Characterizing Schedules Based on Serializability

Unit 7: Concurrency Control and Database Recovery **(7 Hrs)**

Concurrency Control and its purpose; Two-Phase Locking Technique for Concurrency Control; Concurrency Control Based on Timestamp Ordering; Concepts of Recovery; NO-UNDO/REDO Recovery Based on Deferred Update; Recovery Techniques Based on Immediate Update; Shadow Paging; The ARIES Algorithm; Database Backup and Recovery from Catastrophic Failures

3.2 List of Tutorials

Unit-specific exercises focused on problem-solving.

3.3 Laboratory Works

The practical work includes modelling ER and EER diagrams using appropriate CASE tools. The practical extends writing SQL query scripts to define and manipulate databases. Real world database projects should be designed and developed as a group project under the practical work.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books:

- Fundamentals of Database Systems; Seventh Edition; Ramez Elmasri, Shamkant B. Navathe; Pearson
- Database System Concepts; Seventh Edition; Avi Silberschatz, Henry F Korth, S Sudarshan; McGraw-Hill
- An introduction to Database Systems; Eighth Edition; CJ Date, Pearson

डीन

- Database Design and Relational Theory: Normal Forms and All That Jazz; Second Edition; CJ Date; Apress
- Database Management Systems; Third Edition; Raghu Ramakrishnan, Johannes Gehrke; McGraw-Hill
- A First Course in Database Systems; Third Edition; Jaffrey D. Ullman, Jennifer Widom; Pearson
- Learning SQL: Generate, Manipulate, and Retrieve Data; Third Edition; Alan Beaulieu; O'Reilly Media

Web Technology

Course Code	BAI203	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding of web development and the underlying technologies that power the World Wide Web. The course covers various aspects of web technology, including HTML, CSS, JavaScript, server-side scripting, web servers, databases, and web application development. Through a combination of lectures, practical exercises, and projects, students will gain the necessary skills to create dynamic and interactive websites.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce students to the fundamental concepts and technologies that underpin the World Wide Web.
- Provide a solid foundation in HTML and CSS for web page design.
- Familiarize students with JavaScript and teach them how to use it to create dynamic and interactive elements on web pages.
- Introduce server-side scripting languages, such as PHP or Python, for web application development.
- Teach students the process of developing web applications using frameworks and libraries.
- Explore the integration of web applications with databases.
- Provide students with knowledge of web server configuration and deployment techniques.
- Educate students about common security vulnerabilities in web applications and teach them techniques to secure web applications against threats.
- Foster the ability to work collaboratively in a team environment, enabling students to effectively contribute to web development projects, communicate ideas, and coordinate tasks.
- Develop students' skills in identifying and resolving common issues and bugs in web applications through systematic troubleshooting and problem-solving techniques.

3. Course Details

3.1 Theory **(48 hrs)**

Unit 1: Web Essentials **(6 hrs)**

The Internet, Basic Internet Protocols, the World Wide Web, HTTP (HTTP Request and Response Messages),FTP (Request, Response and Modes), Web Clients and Web Servers, Web Page and Web Site (Static and Dynamic),Web Application architecture(Tier architecture), Client Side and Server Side Scripting; Web 2.0, Web 3.0,Domain Registration, Web Hosting, Web application deployment, Using version control tools

Unit 2: HTML and CSS **(12 hrs)**

HTML Introduction, HTML Documents, HTML Elements and Attributes, Text Formatting Elements, Layout elements, Image elements, Table elements, Form elements, HTML 5 and its features, HTML 5 Elements and Attributes, XHTML and its feature, Introduction to CSS, CSS Syntax,

डीन

Three Ways to apply CSS(Inline, Internal, External), CSS Selectors (Universal, ID, Class, Element, CSS combinators, Pseudo-classes, and Pseudo-elements, CSS Attributes selectors),Using CSS Units(Absolute and Relative),Layout related properties (Float, Display: inline, block, inline-block, flex, grid),Position properties (static, relative, fixed, absolute, sticky, z-index), Responsive Web Design, Introduction to Frontend Framework and Library

Unit 3: Client Side Scripting **(12 hrs)**

JavaScript: Introduction, Basic Syntax, Variables(let, const and var) and Data Types, Statements, Operators, Literals, Control Statements, Functions, Objects, Arrays, Built-in Objects, Strings, Forms and Regular Expression, Cookies, JavaScript Event Handling, Introduction HTML DOM, JavaScript Browser BOM, XML introduction, AJAX, JSON, ES6 and its Features (arrow function, spread operator, map, set, promises), Introduction to Client Side Libraries and Frameworks.

Unit 4: Server Side Scripting: PHP **(12 hrs)**

Introduction to server side scripting, Syntax, Variables, Data Types, Operators, Control Statements; Functions, Arrays, Strings, Superglobals, Forms, Form validations, Regular Expression for Server side validation, File Handling, Cookies, Session, Error and Exception, Database Handling(CRUD operation with DML queries), Introduction to Server Side Frameworks.

Unit 5: Web application security **(6 hrs)**

Credential stuffing, SQL Injection, Session Hijacking, Cross Site Scripting (XSS), Cross-site request forgery (CSRF), DDoS Attack, Using Client Side and Server Side technique to handle application attack, Using PDO for secure database transaction

3.2 Tutorial

Unit-specific exercises focused on problem-solving.

3.3 Laboratory Works

The practical work includes creating web pages and applications with using HTML, CSS, JavaScript, PHP, and other technologies mentioned in the syllabus. Students also learn to use client and server side frameworks for developing web applications.

- Learning to use version control tools.
- Design website for different application domain like; (Company, Portfolio, E-commerce and
- Making responsive design using media query
- Making interactive webpage using JS library and framework
- Setting up application using Client side framework
- Setting up application using Server side framework
- Implement server side script for backend CMS.
- Implement related security technique for securing web application
- Finally, student must host their project into web server and make live.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		

छाडि

डीन

	Seminar/Project/Presentation	5		
	Practical/Lab Examination	**20**	**20**	**10**
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester Examination			**40**	**20**

5. Books

- Learning PHP, MySQL & JavaScript: a step-by-step guide to creating dynamic websites, Nixon - O'Reilly – 2021
- Jackson, J. C. (2007). Web Technologies: A Computer Science Perspective.
- Responsive Web design with HTML5 and CSS3 learn responsive design using HTML5 and CSS3 to adapt websites to any browser or screen size, Frain - Packt - 2012
- PHP and MySQL Web development, Welling & Thomson - Sams - 2002
- Web application security: exploitation and countermeasures for modern web applications Hoffman - O'Reilly - 2020

डीन

Data Structure and Algorithm

Course Code	BAI204	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

The Data Structures and Algorithms course is designed to provide students with a comprehensive understanding of fundamental data structures and algorithms that are essential for efficient and effective problem-solving in computer science and programming. This course forms the backbone of computer programming and software development, as it equips students with essential problem-solving skills and strategies to tackle real-world computational challenges.

2. Course Objectives

- Understand the fundamental concepts of data structures and algorithms, their significance, and their role in computer programming.
- Develop a strong foundation in problem-solving techniques and algorithmic analysis to efficiently solve complex computational problems.
- The course will cover the concept of recursion and how it can be employed to design elegant and efficient algorithms. Students will learn how to convert iterative solutions into recursive ones and vice versa.
- Learn different data structures and their applications, such as arrays, linked lists, stacks, queues, trees, graphs, and hash tables.
- Explore fundamental algorithms for sorting, searching, and graph traversal, and understand their time and space complexities.
- Study advanced algorithmic paradigms, such as divide-and-conquer, greedy algorithms, dynamic programming, and backtracking.

3. Course Details

3.1 Theory **(48 hrs.)**

Unit 1: Introduction to Data Structures & Algorithms **(5 Hrs.)**

Overview of algorithms and their significance in computer science. Characteristics of good algorithms: correctness, efficiency, and simplicity. Algorithm Design Techniques: Greedy algorithm, Divide and conquer algorithm, Dynamic programming, Backtracking and randomized algorithm. Algorithm analysis: time complexity, space complexity, and Big Oh notation.

Introduction to data structures, Abstract data type (ADT) and their role in organizing and storing data.

Unit 2: Arrays, Pointers and Linked Lists **(5 Hrs.)**

Introduction to arrays and their properties. Multi-dimensional arrays. Pointers and Dynamic Memory Allocation: malloc, calloc, realloc, free. Singly linked lists: structure, operations, and applications. Doubly linked lists: structure, operations, and applications. Circular linked lists and their advantages.

डीन

Unit 3: Stack **(5 Hrs.)**

Basic Concept of Stack, Stack as an ADT, Stack Operations, Stack Applications, Stack implementation as Array and Linked List. Conversion from infix to postfix/prefix expression, Evaluation of postfix/ prefix expressions.

Unit 4: Queue **(5 Hrs.)**

Basic Concept of Queue, Queue as an ADT, Primitive Operations in Queue, Queue implementation as Array and Linked List. Linear Queue, Circular Queue, Priority Queue, Queue Applications. Double-ended queues (Deque) and their uses.

Unit 5: Recursion **(2 Hrs.)**

Principle of Recursion, Comparison between Recursion and Iteration. Example of recursive algorithms (factorial, Fibonacci Sequence, GCD, and Tower of Hanoi algorithms). Applications and Efficiency of Recursion.

Unit 6: Sorting **(8 Hrs.)**

Introduction; Internal and External Sorting; bubble sort, insertion sort, selection sort, merge sort, quick sort, shell sort, Heap Sort. Complexity of Sorting Algorithms.

Unit 7: Searching and Hashing **(5 Hrs.)**

Introduction to Searching, Search Algorithms: Sequential Search, Binary Search Efficiency of Search Algorithms.

Hashing: Hash Function and Hash Tables, Collision Resolution Techniques

Unit 8: Trees **(6 Hrs.)**

Concept and Definitions, Basic Operations in Binary Tree, Tree Height, Level and Depth

Binary Search Tree, Insertion, Deletion, Traversals, Search in BST, Balancing Tree: AVL tree, DSW Algorithm

Unit 9: Graphs **(7 Hrs.)**

Definition and Representation of Graphs, Graph Traversal, Minimum Spanning Trees: Kruskal and Prims Algorithm. Shortest Path Algorithms: Dijkstra's Algorithm

3.2 Tutorials

Unit-specific exercises focused on problem-solving.

3.3 Laboratory Works:

The laboratory work consists of implementing the algorithms and data structures studied in the course. Since all the units consists of their specified practicals except first unit, so students are expected to perform all the practicals according to its unit lab work.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10

डीन

	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books:

- Y Langsam, MJ Augenstein and A.M , Tanenbaum Data Structures using C and C++, Prentice Hall India, Second Edition 2015
- Leen Ammeral, Programmes and Data Structures in C, Wiley Professional Computing
- G.W Rowe, Introduction to Data Structure and Algorithms with C and C++ , Prentice Hall India
- Srivastava, S.K. & Srivastava, D. (2011). Data structure Through C in Depth, (2nd Ed.), BPB Publication

डीन

Microprocessor & Computer Architecture

Course Code	BAI205	Year/Semester	II/III
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding of microprocessor 8085 architecture, basic computer structure, central processing unit, microprogramed control, computer arithmetic, input/output, memory and pipelining mechanism.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce Microprocessor and Microprocessor based system.
- Explore features and functioning of intel 8085 microprocessor.
- Write an ALP using 8085 microprocessors.
- Familiarize basic computer.
- Identify the organization and design of basic computer.
- Explain microprogrammed control logic.
- Explore central processing unit.
- Familiarize with IO and memory organization.
- Identify computer arithmetic algorithms.
- Introduce pipelining mechanism.

3. Course Details

3.1 Theory **(48hrs.)**

Unit 1: Introduction **(4 hrs.)**

Microprocessor, Components of Microprocessor: ALU, Register Array, Control Unit, Microprocessor as a CPU, Microprocessor based system, Microcontroller, Microprocessor vs Microcontroller, Von-Neumann and Harvard Architecture

Unit 2: Microprocessor 8085 **(10 hrs.)**

Features of 8085 MPU, Operations: Microprocessor-initiated, Internal and Peripheral operations. Pinout and Signals of 8085, Bus Organization, Flags, Interrupts, Register Organization, Functional block diagram, Instruction Set, Opcode and Operands, Addressing modes, Assembly Language Programming with 8085. Instruction Cycle and Timing diagrams, Machine Cycle & T-State.

Unit 3: Basic Computer Organization and Design **(6 hrs.)**

Basic Computer, Instruction codes, Computer Registers, Computer Instructions, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt, Design of Basic Computer, Design of Accumulator Logic.

Unit 4: Microprogrammed Control **(6 hrs.)**

Control Memory, Address Sequencing, Microprogram, Microinstruction format, Symbolic and Binary microprogram, Design of Control Unit.

डीन

Unit 5: Central Processing Unit **(6 hrs.)**

Register Organization, Stack Organization, Instruction Format, Addressing modes, Data Transfer and Manipulation, Program Control, RISC and CISC architecture.

Unit 6: Computer Arithmetic **(6 hrs.)**

Fixed-point vs Floating-point arithmetic, Fixed-point: Addition, Subtraction, Multiplication and Division algorithms.

Unit 7: Input/output and Memory Organization **(6 hrs.)**

Input/output: Peripheral Devices, Input-Output Interfaces, Modes of data transfer, Direct Memory Access (DMA), DMA Controller, Pin Diagram & Internal configuration of 8237 processor. Memory: Memory Hierarchy, Auxiliary memory, Associative memory, Cache memory, Virtual Memory.

Unit 8: Pipelining **(4 hrs.)**

Pipelining, Arithmetic pipelining, Instruction pipelining, Pipelining Hazards.

3.2 List of Tutorials

The various tutorial activities that outfits this course should cover all the content of this course to give students a space to engage more actively with the course content in the presence of instructor. Students should submit tutorials as assignments or class works to the instructor for evaluation. The following tutorial activities of 15 hrs should be conducted to cover all the content of this course:

A. Discussion based Tutorials

- Comparative analysis of Von-Neumann, Harvard and modified Harvard Architecture.
- Applications of microprocessor and microcontroller based system in real world.
- Hardware and programmable module of 8 bit microprocessor.
- Interpret the concept of memory segmentation and pipelining in 16 bit microprocessor.
- Review each entity memory hierarchy for modern processors.
- Interfacing of DMA, interrupt controller and UART with 8085 microprocessor.
- Comparative analysis of different aspects of computing system as defined in Flynn's Classification.

B. Assembly level programming tutorials

- Assembly level programming illustrating data transfer instructions, arithmetic instructions, logical instructions and branching instructions in 8085 microprocessor.
- Enlighten PUSH and POP instructions using appropriate assembly level programs.
- Assembly level programming for simple sequencing program, multiplication, division, table processing and sorting.
- Timing diagram for different types of instructions and RTL of each machine cycle involved during the execution of instructions.

C. Design Thinking tutorials

- Design a CPU for any given registers set, instruction set and state diagram. Show the RTL code for each execution cycle.
- Develop a control unit for any given state diagram.
- Design a micro-sequencer control unit for any given specifications following design procedure.

3.3 Laboratory Works

Laboratory works using 8085 Microprocessor Training Kit

डीन

- Illustrate Data Transfer, Arithmetic, Logical, Branching and Machine Control Instructions.
- Demonstrate instructions from each addressing mode.
- Perform 8 – bits addition, subtraction, multiplication and division.
- Perform 16 – bits addition and subtraction.
- Transfer block of data from one memory location to another.
- Arrange data in ascending/descending order.
- Find largest/smallest datum from data array.
- Identify even/odd, prime/composite number.
- Convert number from one base (radix) to another.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	**Components**	**Weightage**	**Full Marks**	**Pass Marks**
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester End Examination			**40**	**20**

5. Books:

- Ramesh Gaonkar, Microprocessor Architecture, Programming and Applications with the 8085, Penram International Publishing India
- M. Morris Mano, Computer System Architecture, PHI.
- Ajit Pal, Microprocessor: Principles and Applications, TATA McGraw-Hill
- William Stalling, Computer Organization and Architecture, Pearson

डीन

System Analysis and Design

Course Code	BAI206	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding of System and System Development Ecosystem. The course covers different approaches of system analysis and design and various success and challenging real cases of System Development

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce students to the fundamental concepts of System and System Design
- Equip the students with different quality issues of system.
- Familiarize students' system development based upon the valid customer's requirements.

3. Course Details

3.1 Theory **(48 hrs)**

Unit 1: Introduction to System Development **(10 hrs)**

Introduction of System Analysis and Design, Understanding the system, System Diagram, Development Process, Management Process, Supporting Processes, System Structure, ,Traditional waterfall SDLC , Other Approaches: Prototyping , Spiral, RAD, Introduction to Agile Development, CASE tools.

Unit 2: System Concept Formation **(6 hrs)**

Introduction to System Concept, Identifying and Selecting the System Development Projects, Corporate and Information Systems Planning, The Problem Identification, justifying a Solution, Generating broad alternative solution, Evaluating the proposal, Feasibility and its types, Selecting an alternative, Preparing statement of a User Requirement

Unit 3: Requirement Analysis **(6 hrs)**

Introduction to Requirement Analysis, Functional and non-functional requirement, User requirements, System requirements, Interface requirements, Information Collection, Questionnaires, Electronic Data Collection, Information by Observation, Using Ethnography, From observation to design, Information by Prototyping, Interface Prototyping. Prototyping Processes, Interviewing, Software requirement document .

Unit 4: Data Flow Diagram (DFD) **(6 hrs)**

Introduction to Data Flow Diagram, DFD symbols, Describing systems by DFD, Logical and Physical DFDS, Convention for good DFDS, DFDS Levelling.

Unit 5: Process Descriptions **(6 hrs)**

डीन

Introduction to Process Description, Process descriptions method, Structured English, Decision tables, Extended Decision tables, establishing logical correctness of Decision tables, Using Karnaugh maps to detect logical errors in decision tables, Eliminating redundant specifications.

Unit 6 : Implementation and Maintenance **(8 Hrs.)**

System Implementation: Introduction, System Implementation, Software Application Testing, testing types Installation and its types Documenting the System, Training and Supporting Users, Organizational Issues in Systems Implementation; Maintaining Information Systems: Introduction, Maintaining Information Systems, Conducting Systems Maintenance

Unit 7: Object Modelling **(6 hrs)**

Introduction to Object Modelling, Object environment, Object structures, Modelling behaviour in object modelling, Unified Modelling Language (UML), Class and object diagram, Use case diagram, State diagram, Sequence diagram.

3.2 Tutorial

Unit-specific exercises focused on problem-solving.

3.3 Laboratory Work

In the practical session, students will learn to use project management, CASE, and modelling tools. They also prepare a project report that includes at least analysis, design, and implementation phases of system analysis and design.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	40	20
	Assignment	10		
	Seminar/project/presentation	05		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books :

- Igor Hawryszkiewycz (IH), "System Analysis and Design"
- Jeffrey A. Hoffer, Joey F. George, Joseph S.Valacich (JH) "Modern System Analysis and Design"
- Ian Somerville (IS) "Software Engineering"

Year II Semester IV

Research Methodology

Course Code	BAI251	Year/Semester	II/IV
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hr/wk
		Total	96 hrs

1. Course Objective

This course aims to provide students with essential research skills, covering research design, data collection, and analysis. It integrates both qualitative and quantitative methods, emphasizing practical application through case studies and real-world examples. Students will learn to craft research proposals, analyze data, and present findings, preparing them for academic and professional research in civil engineering.

2. Course Details

3. Course Detail

3.1 Theory (48 hrs)

Unit 1: Basics of research (4 hrs)

Introduction to research: Definition and importance. Types of research: Qualitative, quantitative, mixed methods. Research process: Steps in conducting research. Ethical considerations in research: Informed consent and avoiding plagiarism. Intellectual property rights: Patents, copyrights, trademarks, and protecting research work.

Unit 2: Research problem and literature review (7 hrs)

Identifying research problems: Characteristics of a good research problem, formulating research questions. Literature review: Purpose and importance, conducting a literature search, writing a literature review.

Unit 3: Research design and method (12 hrs)

Research design basics: Types of research designs (experimental, descriptive, correlational, exploratory), choosing an appropriate research design. Case study and survey research: Designing case studies, designing surveys and questionnaires. Qualitative methods: Interviews and focus groups, observation techniques. Qualitative data analysis: Coding and thematic analysis, software tools for qualitative analysis. Quantitative methods: Surveys and questionnaires, experimental data collection. Quantitative data analysis: Descriptive and inferential statistics, using software tools (e.g., SPSS, Excel). Mixed methods research: Introduction to mixed methods, integration of qualitative and quantitative data, designing mixed methods research, data collection and analysis in mixed methods (strategies for data integration).

Unit 4: Advanced topics in research methodology (9 hrs)

Introduction to meta-analysis: Purpose and importance, conducting a meta-analysis. Systematic reviews: Methodology and best practices, examples and applications. Advanced sampling methods: Probability and non-probability sampling, sample size determination. Advanced sampling designs: Stratified, cluster, and systematic sampling. Introduction to big data: Characteristics and challenges. Data mining techniques: Association rules, classification, and clustering.

Unit 5: Research proposal and report writing **(9 hrs)**

Components of a research proposal: Title, abstract, and introduction; literature review and methodology. Writing a research proposal: Tips for effective proposal writing, examples and templates. Structure of a research report: Introduction, methodology, results, discussion, conclusions, and recommendations. Writing and presenting research findings: Effective writing strategies, preparing and delivering presentations.

Unit 6: Writing and publishing research papers **(7 hrs)**

Structure of a research paper: Abstract, introduction, methodology, results, discussion, and conclusion. Writing tips and strategies: Clarity, coherence, conciseness, and common pitfalls and how to avoid them. Understanding the peer review process: Types of peer review, responding to reviewers' comments. Selecting a journal for publication: Factors to consider, submission guidelines and process.

3.2 Tutorials

Unit-specific exercises focused on case studies and applications in Computer Science and AI, leading to a research paper draft.

2.3 Laboratory Work

- Qualitative and Quantitative data analysis using appropriate tools.
- Write a publishable research paper

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) internal evaluation and (2) semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	20	20	10
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books

- C.R. Kothari. "Research Methodology - Methods and Techniques." New Age International.
- D. V. Thiel, "Research Methods for Engineers." Cambridge University Press.
- J. W. Creswell and J. D. Creswell, "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches." SAGE Publications.
- R. Kumar, "Research Methodology: A Step-by-Step Guide for Beginners." SAGE Publications.
- U. Flick, "Introducing Research Methodology: Thinking Your Way Through Your Research Project." SAGE Publications.
- G. R. Marczyk, D. DeMatteo, and D. Festinger, "Essentials of Research Design and Methodology." Wiley.

Python Programming

Course Code	BAI252	Year/Semester	II/IV
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	97 hrs

1. Course Description

This Programming with Python course is designed to provide students with a comprehensive understanding of Python programming and web development using the framework. The course starts with an introduction to Python, covering essential topics such as variables, data types, control flow, and functions. Students will learn how to handle files and gain a solid understanding of object-oriented programming (OOP) in Python.The course then delves into a popular web development framework in Python. Students will learn how to set up a python project, create models, perform database operations, and handle user authentication and authorization. Additionally, students will explore templating engine, forms, and the admin interface. Optional topics include Rest Framework for building RESTful APIs and deploying the application on web servers like Heroku or AWS.Throughout the course, students will work on a mini-project using Framework, applying the concepts they have learned to develop a small-scale web application with dynamic content rendering, user authentication, and database interactions.

2. Course Objectives

The primary objective of this course is to equip students with the knowledge and skills to become proficient Python programmers and web developers using the framework. By the end of the course, students should be able to:

- Understand the fundamentals of Python programming, including variables, data types, control flow, and functions.
- Grasp the concepts of object-oriented programming in Python and apply them to real-world scenarios.
- Develop web applications including setting up a project, creating models, and handling user authentication and authorization.
- Utilize Framework templating engine and forms to render dynamic content and handle user input.

3. Course Details

3.1 Theory **(48 hrs.)**

Unit 1: Python Basics: **(10hrs)**

Introduction to Python: Overview of Python and its applications, Installing Python and setting up the environment Running, Python scripts and interactive mode. Variables and Data Types: Numeric data types (int, float, complex), String data type and manipulation, tuples, and dictionaries, Boolean values and operators. Control Flow: If-else statements Loops (for and while), Break and continue statements, Exception handling with try-except blocks. Functions: Defining functions and function arguments, Return values and function scopes, Lambda functions and higher-order functions, Recursion. File Handling: Reading and writing text files, Working with CSV and JSON files

Unit 2: Python OOP: **(6 hrs)**

Introduction to Object-Oriented Programming (OOP): Understanding classes and objects, Encapsulation, inheritance, and polymorphism, Constructors and destructors. Working with Classes and Objects: Defining classes and creating objects Instance and class variables, Methods and properties Inheritance and Composition Extending classes through inheritance
Composition and using multiple classes together. Advanced OOP Concepts: Method overloading; overriding, Abstract classes and interfaces, Class decorators

Unit 3: Python Scripting: **(2 hrs)**

Command-Line Arguments, Parsing command-line arguments, Arg parse module for more complex argument parsing. Working with Modules: Creating and importing modules Packages and the module search path. Regular Expressions: Pattern matching using regular expressions, Replacing and searching for patterns

Unit 4: Python Database and Database Connectivity **(4 hrs)**

Introduction to Databases: Understanding relational databases and SQL Overview of popular database systems. SQL Basics: CRUD operations (Create, Read, Update, Delete) Joins, subqueries, and aggregate functions Database Connectivity in Python: Connecting to databases from Python, ExecutingSQL queries with Python

Unit 5: Introduction to Web Development **(10 hrs)**

Understanding web development concepts, Introduction to Framework and its features, setting up any Framework project and running the development server. Framework Models: Defining models and database tables, Querying data with Framework ORM (Object-Relational Mapping). Views and Templates: Creating views and handling URL patterns Using any Framework templating engine Rendering dynamic data in templates Forms and User Input: Handling forms and user input in Framework Form validation and processing Authentication and Authorization: Managing user accounts and authentication Controlling access to views and resources Static Files and Media Handling: Managing static files (CSS, JavaScript, etc.). Uploading and serving media files (images, videos, etc.). Admin Interface: Using any Framework built-in admin interface Customizing the admin interface for models. Rest Framework (optional): Introduction to RESTful APIs, Creating APIs with Rest Framework

3.4 Tutorials

Unit-specific exercises focused on problem-solving.

3.5 Laboratory Works:

- Project: Personal Blogging Platform
- Description: In this project, you will build a personal blogging platform using Framework. The platform will allow users to register, log in, create blog posts, and view posts from other users. Users can like and comment on blog posts, and admins can manage the posts and user accounts through the admin interface.

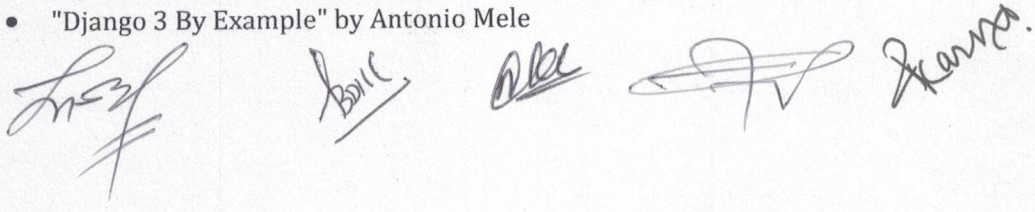
4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	20	20	10
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books:

- "Python Crash Course" by Eric Matthes (2nd Edition): ISBN-13: 978-1593279288
- "Django for Beginners" by William S. Vincent (3rd Edition): ISBN-13: 978-1099701185
- "Python Cookbook" by David Beazley and Brian K. Jones
- "Django 3 By Example" by Antonio Mele



Theory of computation

Course Code	BAI253	Year/Semester	II/IV
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hr/hrs
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding of concepts of theory of computation. The course covers different aspects including string operations, language constructs, finite automata, regular expression, context free language, push down automata, and Turing machine. Through a combination of lectures, practical exercises, and projects, students will gain the necessary skills to design and develop automata machines.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce different concepts of automata theory
- Provide concepts and skills of designing finite machines
- Familiarize with regular expressions
- Provide concepts of designing grammar
- Familiarize with formal language classes
- Present concepts of push down automata
- Acquaint with working mechanism of Turing Machines

3. Course Details

3.1 Theory (48 hrs)

Unit 1: Introduction (3 Hrs.)

Complexity Theory, Computability Theory, Automata Theory, Determinism and Non-determinism, Alphabets, Kleen and Positive of Closure Alphabets, Strings, Empty String, Suffix, Prefix and Substring of a string, Concatenation of strings, Languages, Empty Language

Unit 2: Finite Automata (6 Hrs.)

Finite Automata, Finite State Machine, Deterministic Finite Automata (DFA), Notations for DFA, Language of DFA, Extended Transition Function of DFA, Non-Deterministic Finite Automaton (NFA), Notations for NFA, Language of NFA, Extended Transition Function of NFA, Equivalence of DFA and NFA, Subset-Construction Method, Method for reduction of NFA to DFA, Theorems for equivalence of Language accepted by DFA and NFA, Finite Automaton with Epsilon Transition (ε - NFA), Notations for ε - NFA, Epsilon Closure of a State, Extended Transition Function of ε – NFA, Equivalence of NFA and ε –NFA, Equivalence of DFA and ε – NFA, Finite State Machines with output: Moore Machine and Mealy Machines

Unit 3: Regular Expressions and Regular Languages **(5 Hrs.)**

Regular Expressions, Operators in Regular Expressions (Union, Concatenation, Kleen), Algebraic Rules for Regular Expressions, Application of Regular Expressions, Equivalence of Regular Expression and Finite Automata, Reduction of Regular Expression to ε–NFA, Conversion of DFA to Regular Expression, Arden's Theorem, Regular Languages, Properties of Regular Languages, Pumping Lemma for Regular Expression, Closure Properties of Regular Languages, Minimization of Finite State Machines

Unit 4: Context Free Grammars and Context Free Languages **(8 Hrs.)**

Context Free Grammar (CFG), Components of CFG, Use of CFG, Context Free Language (CFL), Bottomup and Topdown derivation approaches, Leftmost and Rightmost derivations, Sentential Form (Left, Right), Language of a grammar, Parse tree and its construction, Ambiguous grammar, Inherent Ambiguity, Right Linear and Left Linear regular grammars, Equivalence of regular grammar and finite automata, CFG Simplification: Removal of Useless symbols, Nullable Symbols, and Unit Productions, Chomsky Normal Form (CNF), Greibach Normal Form (GNF), Backus-Naur Form (BNF), Context Sensitive Grammar, Chomsky Hierarchy(Type 0, 1, 2, 3), Pumping Lemma for CFL, Closure Properties of CFL

Unit 5: Push Down Automata **(4 Hrs.)**

Push Down Automata (PDA), Representation of PDA, Operations of PDA, Moves of a PDA, Instantaneous Description for PDA, Deterministic PDA, Non Deterministic PDA, Acceptance of strings by PDA, Language of PDA, PDA by Final State, PDA by Empty Stack, Conversion of PDA by Final State to PDA accepting by Empty Stack and vice-versa, Conversion of CFG to PDA and vice-versa

Unit 6: Turing Machine **(6 Hrs.)**

Introduction to Turing Machines (TM), Notations of Turing Machine, Language of a Turing Machine, Instantaneous Description for Turing Machine, Acceptance of a string by a Turing Machines, Turing Machine as a Language Recognizer, Turing Machine as a Computing Function, Turing Machine with Storage in its State, Turing Machine as a enumerator of stings of a language, Turing Machine as Subroutine, Turing Machine with Multiple Tracks, Turing Machine with Multiple Tapes, Equivalence of Multi-tape Turing Machine and Multi-track Turing Machine, Non-Deterministic Turing Machines, Restricted Turing Machines: With Semi-infinite Tape, Multistack Machines, and Counter Machines, Curch Turing Thesis, Universal Turing Machine, Turing Machine and Computers, Encoding of Turing Machine, Enumerating Binary Strings, Codes of Turing Machine, Universal Turing Machine for encoding of Turing Machine

3.2 Practical

The practical work includes implementing the concepts like

- string computation,
- DFA and NFA,
- Tokenizers for identifiers, keywords, etc.
- Regular expressions,
- Grammar derivations,
- Parse tree construction,
- PDA with finals state,
- PDA with empty stack
- Turing Machine.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) internal evaluation and (2) semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books

- Introduction to Automata Theory, Languages, and Computation; John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman; 3rd Edition; Pearson - Addison-Wesley.
- Automata, Computability and Complexity: Theory and Applications; Elaine Rich; Pearson
- Introduction to the Theory of Computation; Michael Sipser; 3rd Edition; Cengage Learning
- What Can Be Computed?: A Practical Guide to the Theory of Computation; John MacCormick; Princeton University Press
- Theory of Computing: A Gentle introduction; Efim Kinber, Carl Smith; Prentice- Hall.
- Introduction to Languages and the Theory of Computation; John Martin; Tata McGraw Hill.
- *Elements of the Theory of Computation;* Harry R. Lewis and Christos H. Papadimitriou; Prentice Hall.

Numerical Method

Course Code	BAI254	Year/Semester	II/IV
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

Numerical methods are those techniques that give an approximate solution of a mathematical problem that has no easy way to calculate an analytical solution. This course consists of the study of numerous tools and techniques required to find numerical solutions of various mathematical problems that frequently arise in the field of science and engineering. Together with the theory, implementation of the theory by means of practical classes will further strengthen the grasp of the subject material.

2. Course Objectives

The objective of this course is to

- Make clear why numerical techniques are needed and the conditions under which such techniques have to be implemented.
- Give an introductory idea about the errors that can arise during numerical computation.
- Introduce various numerical methods for solving a nonlinear equation and comparison of those methods in terms of efficiency.
- Provide the necessary methods of interpolation and approximation and the ability to differentiate between the two.
- Give necessary tools and techniques to solve linear algebraic equations using direct as well as iterative methods.
- Develop the skills to use various techniques of numerical differentiation and integration.
- Introduce the numerical methods of solving first and higher order ordinary differential equations that are specified using initial conditions and boundary conditions.
- Give some methods of numerical solutions of Laplace's equation and Poisson's equation.

3. Course Details

3.1 Theory **(48 hrs.)**

Unit 1: Numerical Solution of Nonlinear Equations **(9 hrs)**

Taxonomy Errors, computing error, Intermediate Value Theorem, Rolle's Theorem, Mean Value Theorem, Taylor's Series

Solving non-linear equations: Bisection Method, Secant Method, Newton Raphson Method, Fixed Point Method and False Position method and their convergence.

Unit 2: Interpolation and Least Squares Approximation **(8 hrs)**

Linear Interpolation, Lagrange's Interpolation Polynomial, Divided, Newton's Interpolation and its Derivation, Divided Difference Table, Error in Polynomial Interpolation, Interpolation for Evenly

Spaced Data: Newton's Forward and Backward Difference Interpolation and their Derivation, Cubic Spline Interpolation using Natural Cubic Splines

Least Squares Approximation, Method of Linear Least Squares and its Derivation, Method of Nonlinear Least Squares with Derivation (Exponential and Polynomial Curve Fitting)

Unit 3: Solution of Linear Algebraic Equations **(10 hrs)**

Review of System of Linear Equations, Consistent and Inconsistent Systems, Gaussian Elimination Algorithm, Ill Conditioned System and Partial Pivoting, Gauss-Jordan Algorithm, Matrix Inversion Method, Matrix Factorization Methods (Dolittle, Crout and Cholesky), Diagonally Dominant System, Iterative Methods (Jacobi, Gauss-Seidel), Convergence of Iterative Methods for Diagonally Dominant Systems, Eigenvalues and Eigenvectors, Eigenvalue Problem, Solving Eigenvalue Problem using Power Method

Unit 4: Numerical Differentiation and Integration **(7 hrs)**

Numerical Differentiation, Forward Difference, Backward Difference and Central Difference Formula for First Derivative, Interpolation Formula for Derivatives, Error Analysis, Central Difference Formula for Second Derivative

Numerical Integration, Trapezoidal Rule with Derivation and Error, Simpson's 1/3 Rule with Derivation and Error, Simpson's 3/8 Rule with Derivation and Error, Composite Rules for Trapezoidal, Simpson's 1/3 and Simpson's 3/8 Methods and their Derivation, Romberg Integration, Gaussian Quadrature Formula with Derivation, Double Integrals

Unit 5: Ordinary Differential Equations **(10 hrs)**

Review of ODE, Order and Degree, Initial and Boundary Value Problems, First-Order First-Degree Equations and their Solutions (Taylor's Series Method, Euler's Method, Modified Euler's Method, Runge-Kutta Method), Solving upto Higher Order Equations, Solving Boundary Value Problem using Shooting Method

Unit 6: Partial Differential Equations **(4 hrs)**

Review of PDE, Elliptic, Parabolic and Hyperbolic PDE, Laplace's Equation and its Difference Equation, Poisson's Equation and its Difference Equation, Solution of Laplace's and Poisson's Equation Using Finite Difference Method

3.2 Tutorial

Unit-specific exercises focused on problem-solving

3.3 Laboratory Works

The practical work will consist of the implementation of methods and algorithms using any programming language. The practical work should include the input-output analysis and demonstrate the understanding of change in accuracy, number of iterations etc. for changes in different parameters in the algorithm.

The practical work should include the implementation of

- Bisection method, Secant method, Newton Raphson method, Fixed-point iteration method and false position method.
- Lagrange's interpolation, Newton's interpolation using divided differences, Interpolating evenly-spaced data using forward difference and backward difference.

- Linear least squares approximation, Non-linear least squares approximation (Polynomial and Exponential fitting).
- Gaussian elimination algorithm, Gaussian elimination algorithm with partial pivoting, Gauss-Jordan algorithm, Jacobi method, Gauss-Seidel method, Power method.
- Central difference method for first and second derivatives, Interpolation method for first derivative.
- Composite trapezoidal method, Composite Simpson's 1/3 method and Composite Simpson's 3/8 method.
- Euler's, Modified Euler's and Runge-Kutta method for solving first order differential equations, Solving systems of differential equations and second order equations using modified Euler's method, Shooting method.
- Implementation of Laplace and Poisson's equation.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books

- Balagurusamy, E. Numerical Methods, McGraw Hill Education.
- B. S. Grewal, Numerical Methods in Engineering and Science, Khanna Publications.
- Cheney, W. and Kincaid, D. Numerical Mathematics and Computing, 7th Edition, Cengage Learning.
- Burden, R. L. and Faires, J. D. Numerical Analysis, 9th Edition, Cengage Learning.
- Chapra, S. C. and Canale, R. P. Numerical Methods for Engineers, 7th Edition, McGraw Hill Education.
- Gerald, C. F. and Wheatley, P. O. Applied Numerical Analysis, 6th Edition, Pearson Education.

Software Engineering

Course Code	BAI255	Year/Semester	II/IV
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a fundamental understanding of state of arts of Software Engineering. The course covers different aspects of software development, software maintenance and software trends in software industries.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce students to the fundamental concepts and technologies that underpin the Software Engineering
- Provide a solid basic foundation of Software Development approaches.
- Familiarize students with different software development approaches.

3. Course Detail

3.1 Theory **(48 Hrs)**

Unit 1. Introduction **(3 Hrs.)**
Professional Software Development; Software Engineering Ethics

Unit 2: Software Processes **(5 Hrs.)**
Software Process Models; Process Activities; Coping with Change; Process Improvement

Unit 3: Agile Software Development **(4 Hrs.)**
Agile Methods; Agile Development Techniques

Unit 4: Requirements Engineering **(4 Hrs.)**
Functional and Non-functional Requirements; Requirements Engineering Processes; Requirements Elicitation; Requirements Specification; Requirements Validation; Requirements Change

Unit 5: System Modeling **(6 Hrs.)**
Context Models; Interaction Models; Structural Models; Behavioral Models; Model-Driven Architecture

Unit 6: Architectural Design **(5 hrs.)**
Architectural Design Decisions; Architectural Views; Architectural Patterns; Application Architectures

Unit 7: Design and Implementation **(6 Hrs.)**

Object-oriented Design using the UML; Design Patterns; Implementation Issues; Open-source development

Unit 8: Software Testing **(4 Hrs.)**

Development Testing; Test-driven Development; Release Testing; User Testing

Unit 9: Software Evolution **(4 Hrs.)**

Evolution Processes; Legacy Systems; Software Maintenance

Unit 10: Software Management **(7 Hrs.)**

Project management (Risk Management, Managing People, Teamwork); Project Planning (Software Pricing, Plan-driven Development, Project Scheduling, Estimation Techniques, COCOMO); Quality Management (Software Quality, Software Standards, Reviews and Inspections, Quality Management and Agile Development, Software Measurement); Configuration management (Version Management, System Building, Change Management, Release Management).

3.2 List of Tutorial **(16)**

Unit-specific exercises focused on problem-solving

3.3 Laboratory Work **(32)**

The practical work includes following activities:

1. Writing the problem statement for any system.
2. Preparing software requirement specification (SRS) for any system.
3. Drawing all UML diagrams using appropriate tools.
4. Learning to use CASE and project management tools.
5. Learning to design and run the test cases using appropriate tools.

In addition to above mentioned activities, students should prepare project report for some system including different activities of software process model.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Recommended Books:

- Ian Sommerville (2016). Software Engineering, 10th Edition, Pearson.
- Roger S. Pressman and Bruce R. Maxim (2019). Software Engineering - a practitioner's approach, 9th Edition, McGraw-Hill Education.

Project I

Course Code	BAI290	Year/Semester	II/IV
Credit Weightage	2	Class Load	4 hrs/wk
		Practical	4 hrs/wk
		Total	64 hrs

1. Course Description

This is a fully practical course that expects students to implement concepts learned during the first two years of their study. Students are encouraged to go beyond the syllabus and develop realistic, technically sophisticated projects.

2. Course Objectives

The objective of this project work is to make the student able to:

- Lead a software project development
- Work in a team
- Use CASE tools
- Write programs and improve programming skills
- Write test cases for software testing and improve QA skills
- Enhance problem-solving, report writing, and presentation skills

3. Nature of Project

- It is an academic project focused on development of a desktop based, web based, or mobile based application using any programming language and technology of their expertise and comfort
- Students can develop the application containing CRUD operations or any other algorithms, if applicable. (Examples: Information Systems, E-Commerce Portals, Game Applications etc.)
- Student should use appropriate CASE tools during SDLC
- While implementing the project, students should be encouraged to write their own modules rather than relying on APIs or Plugins (except in some unavoidable circumstances)
- Emphasis on problem-solving, system design, and implementation
- Project group size: Maximum 3 students in a group

4. Focus of the Study

Each student in a group should have equal participation in every phase of the project. The students should focus on the following different software development phases during the development of their project work;

4.1. Problem Identification

4.2. System Analysis
- Feasibility Study
- System Requirement Specification (SRS)

4.3. System Design

- Architecture Design
- Interface Design
- Database/Procedure/Algorithm Design

4.4. Implementation and Testing

5. Phases of the Project Work

5.1. Proposal Submission and defense
- Timeline: 3rd - 4th week of the 4th Semester

5.2. Mid-Term Progress Report Evaluation and Defense
- Timeline: 11th –12th week of the 4th Semester

5.3. Final Submission and Defense
- Final report submission: At least 10 days before the final defense date
- Defense includes viva, presentation, project demonstration, and report evaluation

6. Supervision

- Supervised by a regular faculty member.
- One supervisor may guide up to four groups.

7. Evaluation Criteria

Total Marks: 100

Evaluation Phase	Evaluators	Marks
Proposal Defense	HoD/Coordinator (4), Supervisor (12), Internal Examiner (4)	20
Mid-Term	HoD/Coordinator (8), Supervisor (24), Internal Examiner (8)	40
Final Defense	HoD/Coordinator (5), Supervisor (15), Internal Examiner (5), External Examiner (15)	40

8. Evaluation Focus on

- Presentation Skills
- Technical Work Quality
- Documentation Quality
- Viva Performance
- Project Demo
- Individual Contributions

9. Roles & Responsibilities

Role	Responsibilities
HoD/Coordinator	Oversee project schedule, arrange defenses, evaluate at all stages
Supervisor	Guide, monitor, and evaluate the project work throughout
Internal Examiner	Evaluate at proposal, midterm, and final defense
External Examiner	Evaluate final presentation, viva, and demo of project
Student	Active in project development, documentation, and defense. Keep supervision logs

10. Report Contents

10.1. Prescribed Content Flow for the Project Proposal

1. Introduction
2. Problem Statement
3. Objectives
4. Methodology
    a. Requirement Identification
        i. Study of Existing System
        ii. Requirement Collection
    b. Feasibility Study
        i. Technical
        ii. Operational
        iii. Economic
    c. High Level System Design
        i. System Flow Chart
        ii. Methodology of Proposed System
        iii. Working Mechanism of Proposed System
5. Project Timeline (Gantt Chart)
6. Expected Outcome
7. References (IEEE Format)

10.2. Prescribed Content Flow for the Project Report

1. Cover & Title Page
2. Certificate Page
    a. Supervisor Recommendation
    b. Internal and External Examiners' Approval Letter
3. Abstract
4. Acknowledgement
5. Table of Contents
6. List of Abbreviations, List of Figures, List of Tables
7. Main Report (Chapters 1–5)
8. Appendices (Screenshots, Source Code, Supervisors Log Sheets)
9. References (IEEE Format)
10. Bibliography (if any)

10.3. Prescribed Chapters in Main Report

Chapter 1: Introduction

1.1. Introduction
1.2. Problem Statement
1.3. Objectives
1.4. Scope and Limitation
1.5. Report Organization

Chapter 2: Background Study and Literature Review

2.1. Background Study (Description of Fundamental theories and concepts of related project)
2.2. Literature Review (Review of the similar project done by other)

Chapter 3: System Analysis and Design

3.1. System Analysis
3.1.1. Requirement Analysis
  i. Functional Requirement (Use-case Diagram, Use-case Description)
  ii. Non Functional Requirement
3.1.2. Feasibility Analysis
  i. Technical
  ii. Operational
  iii. Economic
  iv. Schedule
3.1.3. System Modeling

Structural Approach	Object Oriented Approach
3.1.3.1. Data Modeling: ER Diagram 3.1.3.2. Process Modeling: DFD	3.1.3.1. Object Modeling: Class and Object Diagram 3.1.3.2. Dynamic Modeling: State and Sequence Diagram 3.1.3.3. Process Modeling: Activity Diagram

3.2. System Design

Structural Approach	Object Oriented Approach
3.2.1. Architectural Design 3.2.2. Database Schema Design 3.2.3. Interface Design (UI/UX) 3.2.4. Physical DFD	3.2.1. Refinement of Class and Object Diagram 3.2.2. Component Diagram 3.2.3. Deployment Diagram

Chapter 4: Implementation and Testing

4.1. Implementation
4.1.1. Tools Used (CASE tools, Programming Language, Database Platforms)
4.1.2. Implementation Details of Modules (Description of Procedures/Functions)
4.2. Testing
4.2.1. Test Case for Unit Testing
4.2.2. Test Case for System Testing

Chapter 5: Conclusion and Future Recommendations

5.1. Lesson Learnt and Outcome
5.2. Conclusion
5.3. Future Recommendations

Note: *While writing above chapters students should avoid basic definitions. They should relate and contextualize the above mentioned concepts with their project work.*

11. Citation and Referencing

The listing of references should be listed in the references section. The references contain the list of articles, books, url of primary source that are cited in the document. The books, articles, and others that are studied during the literature review and project work but are not cited in the document can be listed in the bibliography section.
The citation and referencing standard should be **IEEE** referencing standard.

12. Report Format Standards

- **Page size:** A4
- **Font:** Times New Roman, 12pt (body), headings bold (16pt/14pt/12pt).
- **Spacing:** 1.5 line, justified.
- **Margins:** Top/Bottom/Right = 1", Left = 1.25".
- **Page Numbers:** Roman (i, ii...) for preliminaries, numeric (1, 2...) from Chapter 1.
- **Headings:** Chapter – 16pt (Bold and center align), Section – 14pt (Bold and left align), Sub-section – 12pt (Bold and left align), Paragraph content – 12pt (Regular)
- **Figures/Tables:** Center-align, Captions – 12pt (Bold)

13. Binding & Submission

- Copies: 2 (College Library, Self)
- Binding: Plain book binding
- Final Copy: Signed final copy submit to the respective department

(Sample of Cover and Title Page)

Lumbini Technological University

Institute of Engineering and Information Technology

TITLE OF PROJECT REPORT

A PRPJECT REPORT

Submitted to

Department of B.Tech in Computer Science and Artificial Intelligence

Name of the Collage

In partial fulfillment of the requirement for the B.Tech in Computer Science and Artificial Intelligence

Submitted by

Name:

Registration No.:

Symbol No.:

Month, Year

Under the Supervision of

Supervisor Name

Designation

(Sample of Certificate)

Lumbini Technological University

Institute of Engineering and Information Technology

College Name

Supervisor's Recommendation

I, hereby recommend that this project is completed under my supervision by NAME OF THE STUDENT (Year II Semester IV) entitled **"TITLE OF THE PROJECT"** in partial fulfilment of the requirements for the degree of B.Tech in Computer Science and Artificial Intelligence is recommended for the final evaluation.

.....

Signature

Name of Supervisor

Designation of Supervisor

Department Name

Address of the college

(Sample of Approval)

Lumbini Technological University

Institute of Engineering and Information Technology

College Name

LETTER OF APPROVAL

This is to certify that this project is completed by NAME OF THE STUDENT (Year II Semester IV) entitled "**TITLE OF THE PROJECT**" in partial fulfilment of the requirements for the degree of B.Tech in Computer Science and Artificial Intelligence has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

.....
Signature
Name of Supervisor
Academic Designation
Department Name
Address of the college

.....
Signature
Name of HOD/Program Coordinator
Academic Designation
Department Name
Address of the college

.....
Signature
Name of Internal Examiner
Academic Designation

.....
Signature
Name of External Examiner
Academic Designation

UI/UX Design

Course Code	NC	Year/Semester	II/IV
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course will teach you the fundamental process of experience design as well as how to effectively evaluate your work with the people you are designing for. You'll learn fundamental design research methods that will help you understand people, the sequences of their actions, and the context in which they work. You will learn practical techniques for making sense of what you see and transforming your observations into meaningful actionable insights and unique opportunity areas for design through the assignments.

***NC** (Non-Credit) courses are additional offerings provided by the university to enhance students' professional knowledge and practical skills in their field of study. These courses do not contribute to the total degree credits but are evaluated separately. Upon completion, students receive an academic or professional certificate indicating the earned credits.*

2. Course Objectives

The course readings, activities, and assignments were designed to give students opportunities to meet the following objectives by the end of the semester:

- To create and document a portfolio-quality concept design using well-known experience design methods.
- To select appropriate methods and theories based on an understanding of the strengths and limitations of available approaches.
- To articulate your approach to design and user experience design specifically in a clear manner.
- To describe popular research and design processes used in UX as well as alternative approaches to UX design.
- To implement design prototypes in tools such as Photoshop, Illustrator, Figma, Adobe XD, HTML, CSS, Sketch, and Axure.
- To experiment with peer design critique.

3. Course Details

3.1 Theory **(48 Hrs)**

UNIT 1: Introduction to User Experience **(8 hrs)**

User Interface Design, User Experience Design, UI vs UX, User-centric Design, Key framework of UX design, Universal and inclusive design, UX design tools

UNIT 2: User Experience Design **(10 hrs)**

Users, empathy map, users story, edge cases, users journey map, storyboarding, psychology principles that influence design, Ideation: exploring ideas, business needs and competitive analysis.

UNIT 3: User Experience Research **(11 hrs)**

UX research, power of UX research, bias in UX research, choosing the right research method, research plan, goals and questions, UX research process, ethics, privacy and data in real world, usability studies, its types and process.

UNIT 4: Prototyping **(8 hrs)**

Prototypes: wireframes and low-fidelity prototypes, introduction to figma, digital wireframes, Lo-Fi prototype, usability testing and iteration on design.

Unit 5: Visual Design/ UI Design **(11 hrs)**

Introduction to Visual Design/ UI Design, Design Principles: Contrast, Visual hierarchy, Alignment, White space, Gestalt principle, Color theory, Typography, layouts, Iconography and symbols, transition from Lo-Fi to Hi-Fi Design, Design System.

3.2 List of Tutorial **(16 Hrs)**

Unit-specific exercises focused on problem-solving

3.3 Laboratory work **(32 Hrs)**

Mini Research Project

Your mini project for this course is an applied piece of user experience research.
This may be:

- An investigation of user needs for a proposed software product, information system, web site, or similar; or
- An evaluation and assessment for a prototype or an existing system or service.

The more real it is, the better the project. The best projects are for an actual client. But you can also pick a site or product with which you have no direct connection.
Given the constraints of the semester, this is likely to be more of a pilot project than an actual, completed project for a client. It is most important that you get to practice a variety of methods and the entire process from planning to analysis and delivery of a product. It is unlikely that you will have time to go into enough depth with each method to have defendable results.
This can be an individual or a group project; group projects are recommended.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Recommended Books

- Elizabeth Goodman, Mike Kuniavsky, 2012. Observing the User Experience: A Practitioner's Guide to User Research, 2nd edition.
- Designing the User Interface, Ben Schneiderman, McGraw Hill Edition, 4th Edition.
- I. Scott MacKenzie. 2013. Human-Computer Interaction: An Empirical Research Perspective. Morgan Kaufmann
- Don Norman. 2013. The Design of Everyday Things: Revised and Expanded Edition. Basic Books.
- Tomer Sharon, 2016. Validating Product Ideas.

Year III Semester V

Artificial Intelligence

Course Code	BAI301	Year/Semester	III/V
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course provides an introduction to the principles, techniques, and applications of Artificial Intelligence (AI). It covers fundamental AI concepts, including intelligent agents, problem-solving, knowledge representation, machine learning, natural language processing, and robotics. The course also explores the ethical and societal implications of AI.

2. Course Objectives

The course aims to achieve the following objectives: The main objective of the course is to introduce students to the key concepts and methodologies of Artificial Intelligence. By the end of the course, students should be able to design basic AI systems and understand the potential impact of AI on society.

3. Course Details

3.1 Theory (48 hrs)

Unit 1: Introduction to Artificial Intelligence (4 hrs)

What is AI? Definitions and perspectives; History and evolution of AI; AI in the modern world; Applications and real-world examples; Ethical considerations in AI development and deployment.

Unit 2: Intelligent Agents (4 hrs)

Introduction to intelligent agents; Components of intelligent agents: Perception, reasoning, and action, environment types, agent types; multi-agent systems: Coordination and cooperation.

Unit 3: Problem Solving and Searching (10 hrs)

Problem-solving as search: State space representation; Uninformed search algorithms: Breadth-First Search, Depth-First Search, Depth Limited Search, Iterative Deepening and Uniform-Cost Search; Informed search algorithms: Greedy Best First Search, A* Search, Hill Climbing; Constraint Satisfaction Problems and its examples; Adversarial Search: Minimax, Alpha-Beta Pruning.

Unit 4: Knowledge Representation and Reasoning (10 hrs)

Knowledge representation; Rule based, Semantic Networks, Frames; Logic-based; Propositional Logic: Syntax, Semantics, and Resolution; First-Order Logic: Syntax, Semantics, and Resolution; Uncertainty and Probabilistic Reasoning: Probability Distribution, Baye's Theorem Bayesian Networks, Markov Models; Fuzzy Logic and Fuzzy Systems: Handling imprecisions.

Unit 5: Machine Learning (8 hrs)

Introduction to Machine Learning, Supervised, Unsupervised and Reinforcement Learning; Naïve Baye's Classification, K-means Clustering; Artificial Neural Network, Activation Functions, Perceptron Learning, Back Propagation, Hopfield Network; Genetic Algorithm.

Unit 6: Application of AI (NLP) **(8 hrs)**

Expert System; Introduction to NLP and its challenges; Text preprocessing: Tokenization, POS tagging, and Named Entity Recognition; Language models and NLP tasks: Sentiment Analysis, Text Classification, and Machine Translation; Introduction to Robotics and AI applications in Robotics; Robot Perception, Robot Control; Computer Vision.

Unit 7. Future Trends and Ethical Implications **(4 hrs)**

Emerging trends in AI: Explainable AI, and AI for Good, Generative AI; Ethical considerations in AI development: Bias, Privacy, and Accountability; Societal impact and responsible AI.

3.2 List of Tutorial **(16 hrs)**

Unit-specific exercises focused on problem-solving

3.4 Laboratory Works **(32 hrs)**

The practical works must cover the implementation of following concepts;

- Intelligent agents (simple reflex, model based and utility based)
- State space graphs
- Informed and uninformed search techniques
- Game search implementation using minimax
- Rule based systems
- Frames using the concepts of classes
- Predicate logic
- Naïve Bayes Algorithm
- K-means Algorithm
- ANN for AND and OR gate simulation
- Perceptron and Backpropagation Learning
- Implementing NLP concepts using NLTK.

In addition to above mentioned concepts, the laboratory work should include project work including expert systems, question answering systems etc. The choice of programming language can be as per the skills however the concepts of predicate logic should be implemented using logic based programming.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) internal evaluation and (2) semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	40	20
	Assignment	10		
	Seminar/project/presentation	05		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Recommended Books

- Artificial Intelligence: A Modern Approach; Stuart Russell and Peter Norvig; Pearson
- Artificial Intelligence: Kevin Knight, Elaine Rich, Shivshankar B Nair, Mc Graw Hill
- Introduction to Artificial Intelligence: Intelligence and Expert Systems; Dan W. Patterson, Pearson
- Artificial Intelligence with Python: Your complete guide to building intelligent apps using Python 3.x; Alberto Artasanchez, Prateek Joshi; Packt Publishing

Design and Analysis of Algorithms

Course Code	BAI302	Year/Semester	III/V
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course introduces basic elements of the design and analysis of computer algorithms. Topics include asymptotic notations and analysis, divide and conquer strategy, greedy methods, dynamic programming, basic graph algorithms, NP-completeness, and approximation algorithms. For each topic, beside in-depth coverage, one or more representative problems and their algorithms shall be discussed.

2. Course Objective

- Analyze the asymptotic performance of algorithms.
- Demonstrate a familiarity with major algorithm design techniques
- Apply important algorithmic design paradigms and methods of analysis.
- Solve simple to moderately difficult algorithmic problems arising in applications.
- Able to demonstrate the hardness of simple NP-complete problems

3. Course Details

3.1 Theory **(48 hrs.)**

Unit 1: Foundations of Algorithm Analysis **(4 hrs.)**

Algorithm and its properties, RAM model, Time and Space Complexity, detailed analysis of algorithms (Like factorial algorithm), Concept of Aggregate Analysis, Asymptotic Notations: Big-O, Big-Ω and Big-Θ Notations their Geometrical Interpretation and Examples. Recurrences: Recursive Algorithms and Recurrence Relations, Solving Recurrences (Recursion Tree Method, Substitution Method, Application of Masters Theorem

Unit 2: Iterative Algorithms **(4 hrs.)**

Algorithm for GCD, Fibonacci Numbers and analysis of their time and space complexity. Sequential Search and its analysis, Description of Bubble Sort, Selection Sort and Insertion Sort with their complexity analysis.

Unit 3: Divide and Conquer Algorithms **(8 hrs.)**

Concept and applications of divide and conquer approach in algorithm design. Concept and detail description of Binary Search algorithms and its analysis, Finding Minimum and maximum element in a list of items(Min-Max algorithm) and their analysis. Merge Sort algorithm, examples and its time and space complexity Concepts of partitioning, Quick Sort algorithm and its analysis (Best Case,

Worst Case and Average Case). Examples, Randomized Quick Sort and its analysis. Concept of Heap Data Structures(max , min). Heap Sort Algorithm (with Build Heap and Heapify) and its complexity analysis. Concepts of Order statistics, Median order. Bruteforce approach for selection Selection in Expected Linear Time and its analysis. Selection in Worst Case Linear Time algorithm and its complexity analysis.

Unit 4. Greedy Algorithms **(6 hrs.)**

Concept of Optimization Problems and Optimal solution. Introduction of Greedy Strategy for algorithm design. Elements of Greedy Strategy(Greedy Choice Property, Optimal Substructure Property), Concept of Knapsak problem, Algorithm for Fractional Knapsack Problem examples and analysis of its complexity. Concept of Job Sequencing Problem with deadline. Algorithm for Job Sequencing with deadline and its time complexity. Kruskal's and Prim's algorithms for Minimum Spanning Tree, their examples and complexity analysis. Correctness .Dijkastra Shortest Path Algorithms , example and its time complexity, Purpose of Huffman Coding, Prefix Codes, Huffman Tree, Huffman Coding Algorithm, example and its Analysis.

Unit 5: Dynamic Programming **(8 hrs.)**

Concepts of Dynamic Programming approach for algorithm design, Greedy Algorithm vs Dynamic Programming, Recursion vs Dynamic Programming. Elements of Dynamic Programming Approach, Concept of Matrix Chain Multiplication, its Algorithm ,examples and complexity analysis, String Editing Algorithm(edit distance problem with insertion, deletion, replace operation) and its complexity analysis, 0-1 Knapsack problem and its complexity analysis, Floyd Warshall Algorithms for all pair shortest path problem, example and its complexity analysis, Travelling Salesman Problem and its analysis, Concept of Memoization. Dynamic Programming vsMemoization.

Unit 6: Backtracking **(5 hrs.)**

Concept of Backtracking Approach. Recursion vs Backtracking, Concept of Subset Sum, Algorithm for Subset-Sum, its example and Complexity Analysis, Zero-One Knapsack Problem, algorithm with backtracking approach and its analysis, N-Queen Problem and their Analysis

Unit 7: Number Theoretic Algorithms **(5 hrs.)**

Concept of Number Theoretic Notation, Concept of Modular Linear Equations. Chinese Remainder Theorem. Euclid's and Extended Euclid's Algorithms for solving Modular Linear Equations. Miller-Rabin Randomized Primility Test and Analysis

Unit 8. NP Completeness **(5 Hrs.)**

Tractable and Intractable Problems, Concept of Polynomial Time and Super Polynomial Time Complexity, Complexity Classes: P, NP, NP-Hard and NP-Complete, NP Complete Problems, NP Completeness and Reducibility, Cooks Theorem, Proofs of NP Completeness (CNF SAT, Vertex Cover and Subset Sum), Approximation Algorithms: Concept, Vertex Cover Problem, Subset Sum Problem

3.2 Tutorial

Unit-specific exercises focused on problem-solving

3.3 Laboratory Works:

For the laboratory work, students should implement the following algorithms in C/ C++ and perform their analysis for time and space complexity.

1) Basic iterative algorithms GCD algorithm, Fibonacci Sequences, Sequential and Binary Search.
2) Basic iterative sorting algorithms: Bubble Sort, selection Sort, Insertion Sort.
3) Binary Search with Divide and conquer approach.
4) Merge Sort, Heap sort, Quick Sort, Randomized Quick Sort.
5) Selection Problem with divide and Conquer approach
6) Fractional Knapsack Problem, Job sequencing with deadline, Kruskal's algorithm, Prims algorithm, Dijkstra's Algorithm
7) Implement the dynamic programming algorithms.
8) Algorithms using Backtracking approach.
9) Implement approximation Algorithm.

4 Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to algorithms", Third Edition.. The MIT Press, 2009.
- Ellis Horowitz, SartajSahni, Sanguthevar Rajasekiaran, "Computer Algorithms", Second Edition, Silicon Press, 2007.
- Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekiaran, "Computer Algorithms", Second Edition, Silicon Press, 2007.

Compiler Design

Course Code	BAI303	Year/Semester	III/V
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description:

This course is designed to understand the fundamental concepts of compiler design. The course starts with the basic concepts regarding compilation processes and its underlying backend events. The course is structured with different phases of compilation ie, lexical analysis, syntax analysis, syntax-directed translation, type checking etc. in detail.

2. Course Objectives

- Understand the fundamental principles and phases of compiler construction.
- Learn techniques for lexical, syntactic, and semantic analysis.
- Explore intermediate code generation, optimization, and target code generation.
- Gain practical experience in building compiler components.

3. Learning Outcomes

On completion of the course, students will be able to:

- Explain the phases of a compiler and their roles.
- Design and implement lexical analyzers and parsers.
- Analyze syntax and semantics of programming constructs.
- Apply intermediate code generation and basic optimizations.
- Build a working subset of a compiler for a simple language.

4. Course Details

4.1 Theory (48 hrs)

Unit-1: Introduction to Compilers **(5 hrs)**

Compilers vs Interpreters; Structure of a compiler: Front end and back end; Phases of a compiler; Cousins of Compilers; Types of compilers.

Unit-2: Lexical Analysis **(8 hrs)**

Lexical Analysis: Role of lexical analyzer, tokens, lexemes, patterns 2.2 Scanning and 2N buffering; Specification of Tokens- Regular expressions; Recognition of Tokens - Finite automata (NFA, DFA); Conversion from Regular expression to DFA; Lex tool and lexical error handling.

Unit-3: Syntax Analysis **(12 hrs)**

Context-Free Grammars (CFG), Parse Trees, Derivations; Ambiguity in grammars; Errors handling techniques; Top-Down Parsing: Backtracking, Recursive descent, Non recursive predictive parsing, LL(1) grammars and its properties; Bottom-Up Parsing: Shift-reduce, LR(0), SLR, CLR and LALR parsers; Parser generators: Yacc/Bison basics

Unit-4: Semantic Analysis and Syntax-Directed Translation **(8 Hrs)**
Syntax-directed definitions: Inherited and synthesized attributes; Construction of syntax trees; Directed Acyclic Graph; Type checking: Static vs dynamic, type systems; Symbol tables: Structure, scope, and operations; Semantic error detection and reporting.

Unit-5: Intermediate Code Generation **(8 hrs)**
Intermediate representations: Postfix, three-address code (TAC), syntax trees 5.2 Translation of expressions, control flow (if, while, switch); Boolean expressions and backpatching; Type conversion and coercion; Temporary variables and memory management.

Unit-6: Code Generation and Optimization **(7 hrs)**
Issues in code generation: Instruction selection, register allocation 6.2 Simple target machine model; Peephole optimization, loop optimization; Basic block and control flow graph (CFG); Introduction to global optimization techniques.

5. Tutorial
Unit specific problem solving exercise.

6. Practical

Tools: Lex/Flex, Yacc/Bison, Python/C/C++

Lab Experiments
1. Introduction to Lex and Yacc: Write a simple lexical analyzer.
2. Tokenization using Lex: Identifiers, keywords, numbers.
3. Parser with Yacc: Arithmetic expressions grammar and evaluation.
4. Recursive Descent Parser: Manual implementation for basic grammar.
5. Symbol Table Construction: Using hash tables or linked lists.
6. Intermediate Code Generation: Generate 3-address code from expressions. 7.
Type Checking Implementation: Implement semantic rules.
8. Simple Code Generator: Translate intermediate code to target code.

Mini Project: Implement a Top down and Bottom Up parser for a given language.

7. Evaluation Scheme
Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

8. Recommended Books

Textbooks

- Aho, Lam, Sethi, Ullman – Compilers: Principles, Techniques, and Tools (2nd Ed.) · Alfred V. Aho and Jeffrey D. Ullman – Principles of Compiler Design
- Andrew W. Appel – Modern Compiler Implementation in C
- Keith Cooper and Linda Torczon – Engineering a Compiler

Data Science

Course Code	BAI304	Year/Semester	III/V
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description:

The course "Fundamentals of Data Science" aims to provide students with a comprehensive understanding of the field of Data Science. It cater students to covering a wide range of topics, including data manipulation, statistics, machine learning, and data visualization. Through theoretical concepts, practical exercises, and real-world projects, students will gain hands-on experience and knowledge essential for a successful career in Data Science. It is a laboratory course in which students will develop programming skills in loading, cleansing, transforming, modeling and visualizing data.

2. Course Objectives:

- Introduce students to the fundamental concepts and principles of Data Science.
- Provide a solid foundation in data manipulation and analysis using Python and other essential tools.
- Familiarize students with statistical concepts and their application in data analysis.
- Teach students various machine learning algorithms and techniques for predictive modeling.
- Enable students to work with big data using distributed computing frameworks.

3. Course Details

3.1 Theory **(48 Hrs)**

Unit 1: Fundamentals **(5 Hrs)**

Python Programming and tools; Environment set-up, Jupyter Overview, Numerical computition using Numpy, Data Manipulation using pandas; Exploratory data analysis; Data Visualization using (Matplotlib, Pandas, Visdom); Text processing with nltk.

Unit 2: Introduction to Statistics **(10 Hrs)**

Descriptive Statistics and Data visualization, Network Mobilization, Sampling and estimation, Confidence Intervals, Hypothesis Testing, Analysis of Variance, Correlation Analysis, Applied Linear Algebra, Normal Distribution, Central Limit Theorem, Confidence interval, Student's T Distribution, Test Hypothesis, Type I and Type II Errors , T-test.

Unit 3: Regression and Anova **(5 Hrs)**

Regression, Anova, Correlation and Causation, R Square, Adjust4ed R Square

Unit 4: Data Analysis and Visualization **(10 Hrs)**

Introduction to Exploratory Data Analysis, Derive insights of Data with python, Missing Value Analysis, Outlier Detection Analysis, The correlation Matrix, Introduction installation and working with Tableau or Power BI, Deep diving with data and connection, creating charts, Adding calculation to workbook, Mapping data in Tableau, Dashboards and Stories.

Unit 5: Machine Learning **(10 Hrs)**

Scikit-learn, Pytorch, Introduction to Supervised Learning, Linear Regression, Logistic Regression, Naïve Bayes, Decision Tree Classifier, Random Forest Classifier , Support Vector Machine, Introduction to Neural Networks.

Unit 6: Machine Learning in the Cloud **(8 Hrs)**

Introduction to Data Science and Machine Learning on Cloud Platforms (AWS, Microsoft Azure, Google Cloud); Deploying Machine Learning Models on the Cloud

3.2 List of Tutorials

Tutorial should cover the problem solving excise of each unit.

3.3 Laboratory work

Unit 1: Fundamentals

- Python Environment Setup: Install Python and required libraries (Numpy, Pandas, Matplotlib, NLTK, Visdom) on your system.
- Create a new Jupyter Notebook and execute some basic Python code
- Numerical Computation using Numpy: Perform numerical operations using Numpy arrays.
- Practice array slicing, reshaping, and mathematical operations.
- Data Manipulation using Pandas: Load datasets using Pandas. Explore data manipulation techniques like filtering, sorting, and grouping.
- Exploratory Data Analysis and Visualization: Use Matplotlib and Pandas for data visualization. Create basic plots (line plot, scatter plot, bar plot, etc.) to analyze data.
- Text Processing with NLTK: Preprocess text data using NLTK library. Perform tokenization, stemming, and other text processing tasks.

Unit 2: Introduction to Statistics

- Descriptive Statistics and Data Visualization: Calculate descriptive statistics (mean, median, standard deviation, etc.). Visualize data distributions using histograms and box plots.
- Hypothesis Testing and Confidence Intervals: Understand hypothesis testing concepts (Null Hypothesis, p-value, significance level). Calculate confidence intervals for sample data.
- Correlation Analysis and ANOVA: Analyze correlation between variables. Perform one-way ANOVA to compare means across different groups.

Unit 3: Regression and ANOVA

- Linear Regression:Implement simple linear regression with Python. Analyze the relationship between two variables.
- Multiple Linear Regression: Extend linear regression to multiple independent variables.

- ANOVA (Analysis of Variance): Implement one-way ANOVA with Python.

Unit 4: Data Analysis and Visualization

- Exploratory Data Analysis (EDA): Perform EDA on a given dataset. Identify patterns, trends, and insights from the data.
- Missing Value Analysis and Outlier Detection: Handle missing values in a dataset. Detect and handle outliers using Python.
- Introduction to Tableau or Power BI: Install and set up Tableau or Power BI.
- Connect to a dataset and create basic visualizations.
- Data Visualization with Tableau or Power BI: Create charts, graphs, and dashboards in Tableau or Power BI.

Unit 5: Machine Learning

- Introduction to Scikit-learn and PyTorch:
- Install Scikit-learn and PyTorch libraries.
- Learn the basics of both libraries.
- Linear Regression with Scikit-learn:
- Implement linear regression using Scikit-learn. Logistic Regression and Naïve Bayes:
- Implement logistic regression and Naïve Bayes classifiers.
- Decision Tree and Random Forest:
- Implement decision tree and random forest classifiers.
- Support Vector Machine (SVM):
- Implement SVM for classification tasks.

Unit 6: Data Science and Machine Learning in the Cloud

- Introduction to Cloud Platforms (AWS, Microsoft Azure, Google Cloud):
- Create accounts on AWS, Microsoft Azure, or Google Cloud.
- Explore cloud-based data science and machine learning services.
- Deploying Machine Learning Models on the Cloud:
- Deploy a trained machine learning model on the chosen cloud platform.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) internal evaluation and (2) semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	40	20
	Assignment	10		
	Seminar/project/presentation	05		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Recommended Books:

- "Python for Data Analysis" by Wes McKinney
- https://wesmckinney.com/book/

Reference Books:

- "Introduction to Statistical Learning" by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron Geron, A. (2017),
- Hands-On Machine learning with Scikit-Learn & TensorFlow, O'Reilly. McKinney, W. (2013), Python for Data Analysis, O'Reilly.
- VanderPlas, J. (2016), Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly.

Journals and Magazines:

- IEEE Transactions on Knowledge and Data Engineering, IEEE
- ACM/IMS Transactions on Data Science, ACM
- Journal of Machine Learning Research (JMLR), Microtome
- Neural Networks, Elsevie

Best Learning Websites and Blogs:

- Python tutorials available online: https://docs.python.org/3/tutorial/
- Jupyter notebook tutorials available online: https://ipython.org/documentation.html
- Numpy tutorials available online:https://numpy.org/doc/stable/
- Pandas tutorials available online: https://pandas.pydata.org/docs/
- Nltk tutorials available online: https://www.nltk.org
- Matplotlib tutorials available online: https://matplotlib.org/contents.html
- Visdom tutorials available online: https://github.com/facebookresearch/visdom
- Scikit-learn tutorials available online: https://scikit-learn.org/stable/user_guide.html
- Pytorch tutorials available online: https://pytorch.org/tutorials/
- Kaggle (www.kaggle.com)
- Data Science Central (www.datasciencecentral.com

Computer Graphics

Course Code	BAI305	Year/Semester	III/V
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course introduces the basic concepts and techniques of interactive computer graphics and multimedia systems. The course focuses on the design and implementation of computer graphic rendering and visual multimedia systems, with an emphasis on the practical aspects of these technologies.

2. Course Objectives

The course aims to achieve the following objectives:

- To develop an understanding and awareness of how issues such as content, information architecture, motion, sound, design, and technology merge to form effective and compelling interactive experiences for a wide range of audiences and end users
- To become familiar with various software programs used in the creation and implementation of multimedia
- To appreciate the importance of technical ability and creativity within design practice
- To gain knowledge about graphics hardware devices and software used
- To understand the two-dimensional graphics and their transformations
- To understand the three-dimensional graphics and their transformations
- To understand illumination and Rendering techniques
- To become familiar with understand clipping techniques
- To become familiar with standard Three-dimensional graphics tool
- To become familiar with Multimedia system design and its file handling
- To become familiar with Hypermedia

3. Course Details

3.1 Theory **(48 Hrs)**

Unit 1: Introduction to Computer Graphics and Graphics System **(6 hrs)**

Computer Graphics and its types; Application of Computer Graphics; Video Display Devices; Raster Scan Systems; Random Scan Systems; Graphics Monitors; Workstations; Input Devices; Graphics Software

Unit 2: Two dimensional Computer Graphics **(12 hrs)**

Line Drawing Algorithm (DDA, Bresenham); Circle drawing algorithm(Midpoint); Two-dimensional geometric transformations: Matrix representations and homogeneous coordinates, composite transformations; Twodimensional viewing: viewing pipeline, viewing coordinate reference frame; window-to-viewport coordinate transformation; clipping operations: line(/cohen/Sutherland, Liang

Barsky Line clipping algorithm), and polygon clipping algorithms(Sutherland Hodgeman Polygon clipping algorithm)

Unit 3: Three Dimensional Computer Graphics **(13 hrs)**

Three-dimensional concepts; Spline representations – Bezier curves and surfaces -BSpline curves and surfaces. Transformation and Viewing: Three-dimensional geometric and modelling transformations – Translation, Rotation, Scaling, shearing, composite transformations; Three-dimensional viewing – viewing pipeline, viewing coordinates, Projections; Visible surface detection methods, Illumination Model; Polygon Rendering Techniques (Flat, Gouraud, Phong)

Unit 4: Multimedia System Design and Multimedia File Handling **(8 hrs)**

Multimedia basics; Multimedia applications; Multimedia system architecture; Evolving technologies for multimedia; Defining objects for multimedia systems; Multimedia data interface standards; Multimedia databases; Compression and decompression; Data and file format standards; Multimedia Input output technologies; Digital voice and audio; Video image and animation; Full motion video; Storage and retrieval technologies

Unit 5: Hypermedia **(9 hrs)**

Multimedia authoring and user interface; Hypermedia messaging; Mobile messaging; Hypermedia message component; Creating hypermedia message; Integrated multimedia message standards; Integrated document management; Distributed multimedia systems.

3.2 List of Tutorial **(16 Hrs)**

Unit-specific exercises focused on problem-solving

3.3 Laboratory Work **(32 Hrs)**

The practical work includes writing a program using the standard programming and graphics library for implementing the topics of two dimensions, three dimensions, and multimedia mentioned in the syllabus. Students are also encouraged to study open-source software such as blender as a case study.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Recommended Books

- Donald Hearn and Pauline Baker M (2007).Computer Graphics,
- Andleigh, P. K and KiranThakrar(2003), —Multimedia Systems and Design.
- Foley, Vandam, Feiner and Hughes (2003),Computer Graphics: Principles and Practice, 2nd Edition, Pearson Education
- Ze-NianLi and Mark S. Drew (2004), Fundamentals of Multimedia, First Edition
- Peter Shirley, Michael Ashikhmin, Michael Gleicher, Stephen R Marschner, Erik Reinhard, KelvinSung, and AK Peters(2010) —Fundamentals of Computer Graphics, CRC Press

Data Communication and Networks

Course Code	BAI306	Year/Semester	III/V
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding of data communications and computer network focusing on the protocols, services and devices in each layer of TCP/IP reference model.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce data communication and computer networks.
- Identify the functions, devices, and protocols used in physical layer.
- Identify the services, devices, and protocols used in datalink layer.
- Identify and configure the services, devices, and protocols used in network layer.
- Explore the services, devices, and protocols used in transport layer.
- Find out the services, devices, protocols and networking tools used in applications layer.

3. Course Details

3.1 Theory

(48 hrs)

Unit 1: Introduction **(6 hrs)**

Data Communications; Network criteria, Networks and Types; LAN Topologies, Networking devices, Network protocol & standards, OSI Reference model, TCP/IP Reference model, Addressing

Unit 2: Physical Layer **(12 hrs)**

Data and Signals, Periodic and Analog Signals, Digital Signals, Transmission Impairments, Data Rate Limits, Performance, Digital to Digital Conversion, Analog to Digital Conversion, Transmission Modes, Multiplexing, Transmission Media, Switching Techniques

Unit 3: Datalink Layer **(8 hrs)**

Functions, Error detection and Correction: Block Coding, Cyclic codes, Checksum; Data Link Control: Framing, Flow and Error Control, Protocols, Noiseless Channels, Noisy Channels; Multiple Access control: Random Access, Controlled Access and Channelization; Cellular telephony and Satellite network.

Unit 4: Network Layer **(8 hrs)**

Functions; IPv4 Address classes, header format, computing network and host IP addresses, subnetting; IPv6 header format, IPv4 vs IPv6, Routing: Static vs. Dynamic Routing; Routing Algorithms: Shortest-path, Flooding, Distance-vector, Link-state; Congestion control and prevention; Network layer protocols: IP, ICMP, IGMP, RIP, ARP, RARP, OSPF, IGRP, EIGRP, BGP.

Unit 5: Transport Layer **(7 hrs)**

Functions; Reliable vs Unreliable Services, Connection-oriented vs connection-less services, Overview of TCP and UDP protocols and their applications, principle of reliable data transfer, congestion control.

Unit 6: Application Layer **(7 hrs)**

Functions; Application layer protocols: DNS, DHCP, SMTP, POP, IMAP, FTP, Telnet, HTTP, WWW, SNMP; Network Simulator and analyser tools: Packet Tracer, Wireshark.

3.2 Tutorial

Unit specific exercises focused on discussion, case studies and solutions of problem.

3.3 Laboratory Works:

- Prepare a specification document for networking devices: Router, Switch etc.
- Prepare Straight and cross-over cables.
- Implement LAN topology using packet tracer.
- Configure DNS and DHCP Server.
- Configure file server.
- Implement concept of subnetting using packet tracer.
- Prepare a VLAN.
- Implement static and dynamic routing using packet tracer.
- Analyse the data packets using Wireshark.
- Connect dissimilar networks.
- Configure active directory service in windows system.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) internal evaluation and (2) semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	40	20
	Assignment	10		
	Seminar/project/presentation	05		
	Practical/lab examination	20	20	10
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

5. Books:

- Behrouz A. Forouzan; Data Communications and Networking, McGraw Hill.
- William Stallings, Data and Computer Communications, Prentice Hall of India, New Delhi.
- A.S. Tanenbaum, Computer Network, Pearson Education International.
- Kurose, Ross, Computer networking: A top-down approach, Pearson.

Year III Semester VI

Artificial Neural Networks

Course Code	BAI351	Year/Semester	III/VI
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	2 hrs/wk
		Total	80 hrs

1. Course Description

The course introduces the underlying principles and design of Neural Network. The course covers the basics concepts of Neural Network including: its architecture, learning processes, single layer and multi-layer perceptron

2. Course Objectives

- To introduce the neural networks for classification and regression
- To give design methodologies for artificial neural networks
- To demonstrate neural network applications on real-world tasks

3. Course Details

3.1 Theory **(48 hrs.)**

Unit 1: Introduction **(10hrs)**

Neural networks characteristics, History of development in neural networks principles, Artificial neural net terminology, Model of a neuron, Topology, Learning, Types of learning, Supervised, Unsupervised, Reinforcement learning, knowledge representation and acquisition
Learning Process: Competitive, Boltzmann learning, Credit Assignment Problem, Memory, Adaption, Statistical nature of the learning process.

Unit 2: Single Layer Perceptrons: **(10 hrs)**

Adaptive filtering problem, Unconstrained Organization Techniques, Linear least square filters, least mean square algorithm, learning curves, Learning rate annealing techniques, perception – convergence theorem, Relation between perception and Bayes classifier for a Gaussian Environment.

Unit 3: Multi Layer Perceptrons: **(10 hrs)**

Back propagation algorithm XOR problem, Heuristics, Output representation and decision rule, Computer experiment, feature detection, Back propagation and differentiation, Hessian matrix, Generalization, Cross validation, Network pruning Techniques, Virtues and limitations of back propagation learning, accelerated convergence, supervised learning.

Unit 4: Special networks: **(10 hrs)**

Radial basis function network: structure and working procedure, advantages, LVQ network: structure and learning approach, Hopefield network, Autoassociative memory network: general structure and Purpose, Autocorrelator, Heterocorrelator

Unit 5: Self-Organization Maps **(8 hrs)**

Two basic feature mapping models, Self-organization map, SOM algorithm, properties of feature map, computer simulations, learning vector quantization, Adaptive patter classification, Hierarchal Vector quantilizer, contexmel Maps

3.6 Laboratory Works (32 hrs)

Practical should be focused on Single Layer Perceptron, Multi-layer Perceptron, Supervised Learning, Unsupervised Learning, Recurrent Neural Network, Linear Prediction and Pattern Classification

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books:

- S. Haykin, "Neural networks and Learning Machines", Pearson Education, 3rd Edition, 2013.
- B.Vegnanarayana, "Artificial neural networks", Prentice Halll of India, 1998.
- James A. Freeman and David M. Skapura, "Neural Networks: Algorithms, Applications, and Programming Techniques", Addison-Wesley, 1991.

Cloud Computing

Course Code	BAI352	Year/Semester	III/VI
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description:

This course is to provide an introduction to cloud computing and its implementation in an emerging technological perspective. This course will cover various aspects of cloud in terms of its deployment and service models, management issues, security challenges and future research trends. It also discusses the state of the art of cloud computing and related research areas.

2. Course Objectives:

- Make students familiar with cloud computing and distributed systems.
- Students get information about deployment and service models, management issues, security challenges and future research trends.
- Student become familiar with cloud service providers and their services

3. Course Detail

3.1 Theory **(48 hrs)**

Unit-1: Introduction **(6 hrs)**

Introduction to Cloud Computing; The evolution of Cloud Computing; State of Arts of Cloud Computing; Features of Cloud Computing; Challenges of Cloud Computing.

Unit-2: Cloud Service Delivery Model **(6 hrs)**

Introduction to types of Cloud Services; Software as a Service (SaaS); Platform as a Service (PaaS); Infrastructure as a Service (IaaS); Other Services such as (DaaS, XaaS, etc).

Unit-3: Cloud Deployment Model **(5 hrs)**

Introduction to Cloud Types; Private Cloud; Public Cloud; Hybrid Cloud; Community Cloud.

Unit-4: Cloud Technologies **(6 hrs)**

Introduction to Cloud Technology; Virtualization; Service-Oriented Architecture (SOA); Cluster Computing; Grid Computing; Distributed Computing.

Unit-5: Cloud Security **(9 hrs)**

Introduction to Information Security; Cloud Security: Infrastructure Security, Data Security and Storage; Security Management in Cloud; Security -As -A Service.

Unit-6: Cloud Performance **(6 hrs)**

Service Availability; Reliability; Fault Tolerance.

Unit-7: Introduction to Big Data and the Internet of Things (IoT) **(6 hrs)**

Features of Big Data; Managing Big Data with Cloud Computing; Introduction to IoT; Relationship between IoT, Big Data and Cloud Computing.

Unit-8: Case study on open source and commercial Clouds **(4 hrs)**
Openstack; Azure; Amazon EC2

3.2 List of Tutorials **(16 Hrs)**
Unit specific exercises focused on discussion, case studies and solutions of problem.

3.3 Laboratory Work **(32 Hrs)**

- Familiarisation with web services and different protocols involved in web services. Familiarisation with basic services provided by proprietary cloud service providers and open source cloud service software.
- Application deployment from cloud platforms along with different perspectives of performance and security.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester End Examination			**40**	**20**

5. Recommended Books

- Furht Borko, Handbook of Cloud Computing , Springer, 2010, First Edition
- Mather Tim, Kumarswamy Subra, Latif Sahid, Cloud Security and Privacy , O'Reilly, 2009, First Edition
- Ricardo Puttini, Thomas Erl, and Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture,
- James F. Ransome, John W. Rittinghouse Cloud Computing Implementation Management and Security

Cryptography & Network Security

Course Code	BAI353	Year/Semester	III/VI
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is designed to provide students with a comprehensive understanding cryptography and network security concepts. The course covers different aspects of security, cryptography, classical cryptosystems, modern cryptosystems, hash function, message authentication, digital signature, user authentication, network and end point security and cloud and IoT security. Through a combination of lectures, practical exercises, and projects, students will gain the necessary skills to secure computer systems using cryptographic algorithms and network security protocols.

2. Course Objectives

The course aims to achieve the following objectives:

- Introduce different concepts computer, information and network security.
- Provide concepts of classical ciphers
- Familiarize with private key cryptography
- Familiarize with public key cryptography
- Provide concepts of hash functions and message authentication codes
- Present concepts of digital signature for message authentication
- Acquaint with user authentication
- Deliver concepts of network security protocols
- Acquaint with concepts of firewalls, malicious logics, DOS attacks
- Familiarize with Cloud and IoT Security

3. Course Details

3.1 Theory **(48 hrs)**

Unit 1: Introduction **(3 Hrs)**

Security: Computer Security, Information Security, Network Security; CIA Triad; Cryptography: Block Vs Stream, Public Vs Private; Cryptanalysis; Security Threats and Attacks; Security Services; Security Mechanisms

Unit 2: Classical and Modern Ciphers **(10 Hrs)**

Classical Cryptosystems: Ceasar, Monoalphabetic, Playfair, Hill, Polyalphabetic, Rail Fence; Modern Ciphers: Data Encryption Standard (DES), Triple DES; Finite Fields: Groups, Rings, Fields, Galois Fields (GF(p) & GF(2^n)), Polynomial Arithmetic; Advanced Encryption Standard (AES); Number Theory: Prime Numbers, Fermat's and Euler's Theorem, Primality Testing, Euclidean Algorithm, Extended Euclidean Theorem; Discrete Logarithms; Diffie-Helman Key Exchange; RSA Algorithm

Unit 3: Hash Functions, Message Authentication Codes and Digital Signatures **(8 Hrs)**

Hash Functions; Message Digests: MD5; Secure Hash Algorithms: SHA-2, SHA-3; Message Authentication, Message Authentication Functions Message Authentication Codes, HMAC; Digital Signatures: Direct Digital Signatures, Arbitrated Digital Signature; Digital Signature Algorithm; RSA as a Digital Signature

Unit 4: User Authentication **(5 Hrs)**

Authentication System; Password Based Authentication; Token Based Authentication; Biometric Authentication; Remote User Authentication; Kerberos Protocol

Unit 5: Network Security **(12 Hrs)**

Web Security; Transport Layer Security; HTTPS; Wireless Network Security: Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN Overview, IEEE 802.11i Wireless LAN Security; E-mail Security: Email Architecture, E-mail Formats, Email Threats and Comprehensive Email Security, S/MIME, Pretty Good Privacy (PGP), DNSSEC; IP Security: IP Security Overview, IP Security Policy, Authentication Header, Encapsulating Security Payload, Security Associations

Unit 6: Network Endpoint Security **(5 Hrs)**

Firewall; Intrusion Detection System; Malicious Software; Denial of Service Attack; Distributed Denial of Service Attack

Unit 7: Cloud and IoT Security **(5 Hrs)**

Cloud Computing; Cloud Security Concepts; Cloud Security Risks and Counter Measures; Cloud Security as a Service; Internet of Things; IoT Security Concepts and Objectives

3.2 Tutorial

Unit specific exercises focused on discussion, case studies and solutions of problem.

3.3 Laboratory Works

The practical work includes implementing the concepts of cryptographic algorithms, hash functions, digital signatures. The practical work extends implementation and simulation of Network Security Protocols, Intrusion Detection Systems, DDoS Attacks, Cloud Security and IoT Security Systems. Students are free to use any of the language, tools and platform as per the skills.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester End Examination			**40**	**20**

5. Recommended Books

- Cryptography and Network Security: Principles and Practice; William Stallings; Eighth Global Edition; Pearson
- Network Security Essentials: Applications and Standards; William Stallings; Sixth Edition; Pearson
- Cryptography & Network Security; Behrouz. A. Forouzan; First Edition; Mc Graw Hill Education
- Computer Security: Principles and Practice; William Stallings and Lawrie Brown; Fourth Edition; Pearson
- Network Security and Cryptography; Sarhan M. Musa; Mercury Learning and Information
- Computer Network Security Fundamentals; Joseph Migga Kizza; Fifth Edition, Springer

Image Processing

Course Code	BAI354	Year/Semester	III/VI
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course covers the investigation, creation and manipulation of digital images by computer. The course consists of theoretical material introducing the mathematics of images and imaging. Topics include representation of two-dimensional data, time and frequency domain representations, filtering and enhancement, the Fourier transform, convolution, interpolation. The student will become familiar with Image Enhancement, Image Restoration, Image Compression, Morphological Image Processing, Image Segmentation, Representation and Description, and Object Recognition.

2. Course Objectives: The objective of this course is to make students able to:

- To develop a theoretical foundation of Digital Image Processing concepts.
- To provide mathematical foundations for digital manipulation of images; image acquisition; preprocessing; segmentation; Fourier domain processing; and compression.
- To gain experience and practical techniques to write programs for digital manipulation of images; image acquisition; pre-processing; segmentation; Fourier domain processing; and compression.

3. Course Details

3.1 Theory **(48 hrs.)**

Unit 1: Introduction **(5 hrs)**

Digital image representation; Fundamental steps in digital image processing; Components of an image processing system; Elements of visual perception: structure of human eye, Image formation in eyes; Image sensing and acquisition: image acquisition using single sensor; Basic concept of sampling and quantization; Relationships between pixels: Neighbor of pixel, adjacency, connectivity, regions and boundaries.

Unit 2: Intensity Transformation **(5 hrs)**

Basic intensity transformation functions: Negative transformation, Log transformation, Gamma transformation; Histogram processing: Histogram equalization, Use of histogram statistics for image enhancement.

Unit 3: Image filtering in Spatial Domain **(6 hrs)**

Spatial filtering: Spatial correlation, convolution and padding, generating spatial filter mask; Smoothing spatial filters: Linear filters, nonlinear filters; Sharpening spatial filters: Unsharp masking and high-boost filtering, Using first order derivative (Gradient), Using second order derivative (Laplacian).

Unit 4: Image Filtering in Frequency Domain **(8 hrs)**

Concept of frequency of an image, Fourier series, Fourier transformation: 1-D and 2-D DFT, Properties of DFT, Frequency domain filters: Smoothing frequency domain filter (Ideal low pass filter, Butterworth low pass filter, Gaussian low pass filter), Sharpening frequency domain filter (Ideal high pass filter, Butterworth high pass filter, Gaussian high pass filter); Fast Fourier Transformation:

Derivation of 1-D FFT, Conversion of data from spatial domain to frequency domain using FFT, conversion of data form frequency domain to spatial domain using IFFT; Haar transformation, Hadamard transformation.

Unit 5: Image Restoration and Reconstruction **(6 hrs)**
Introduction to image restoration, Noise Models (Gaussian, Rayleigh, Impulse, Gamma, Uniform, and Exponential), Restoration filters: Mean Filters (Arithmetic, Geometric, and Harmonic), Order Statistics Filters (Median, Max and Min, Midpoint, and Alpha trimmed mean), Band pass and Band Reject Filters (Ideal, Gaussian, and Butterworth).

Unit 6: Image Compression **(5 hrs)**
Introduction to image compression, Compression ratio, Relative data redundancy, Average code length, Redundancies in Image: Coding redundancy (Huffman code), Inter-pixel redundancy (Run length coding) and Psychovisual redundancy; Image compression model.

Unit 7: Morphological Image Processing **(3 hrs)**
Introduction to mathematical morphology, Basic concept of set theory, Logic operation in binary image, Hit, Fit and Miss, Erosion and Dilation, Opening and Closing.

Unit 8: Image Segmentation **(6 hrs)**
Introduction to image segmentation; Similarity based approach (Thresholding: Global, Local and Adaptive); Discontinuity based approach (Point, Line and Edge Detection); Edge linking and boundary detection; Hough transform; Region based segmentation: Region growing algorithm, Region splitting and merging algorithm.

Unit 9: Representation, Description and Object Recognition **(4 hrs)**
Representation and Description: Chain codes, Shape numbers, and Signature; Object Recognition: Pattern and Pattern class, Recognition based on decision-theoretic methods (Matching, Neural Networks).

3.2 List of Tutorials
Unit specific exercise focused on problem solving.

3.3 Laboratory Works
Students are required to develop programs in related topics using suitable programming languages such as MatLab or Python or other similar programming languages.

4. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester Examination			**40**	**20**

5.1 Text Books

- Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", Pearson Edition, Latest Edition.

5.2 Reference Books

- I. Pitas, "Digital Image Processing Algorithms", Prentice Hall, Latest Edition.
- A. K. Jain, "Fundamental of Digital Image processing", Prentice Hall of India Pvt. Ltd., Latest Edition.
- K. Castlemann, "Digital image processing", Prentice Hall of India Pvt. Ltd., Latest Edition.
- P. Monique and M. Dekker, "Fundamentals of Pattern recognition", Latest Edition.

Year IV Semester VII

IT Project Management

Course Code	BAI401	Year/Semester	IV/VII
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course provides students with the knowledge and skills necessary to effectively plan, execute, and manage information technology (IT) projects. It covers project management principles, tools, and techniques specific to IT projects, emphasizing successful project delivery within scope, time, and budget constraints.

2. Course Objectives

- Understand the fundamentals of project management in an IT context.
- Define project scope, objectives, and requirements.
- Develop comprehensive project plans and schedules.
- Manage project resources, including human resources and technology.
- Identify, assess, and mitigate project risks.
- Monitor project progress, control changes, and ensure quality.
- Effectively communicate with project stakeholders.
- Apply various project management methodologies and frameworks.

3. Course Details

3.1 Theory **(48 Hrs)**

Unit 1: Project Management and ICT for Development **(6 Hrs)**

Introduction to IT Project Management: Importance, Project, IT Projects Vs Non-IT Projects, ICT4D Project, Project Management, Contract Management, and technical project management, Activities, Plan, Methods, and methodologies, some ways to categorizing software projects, management, Step wise – an overview of project planning: Introduction to stepwise project planning, select project, identify project scope and objectives, identify project infrastructure, analyze project characteristics, identify project products and activities, Programme management and project evaluation: Programme management, managing the allocation of resources within programmes, strategic programme management , evaluation of individual projects, technical assessment, cost benefit analysis, cash flow forecasting, cost benefit evaluation techniques, risk evaluation

Unit 2: Selection of IT Project **(10 Hrs)**

Selection of an appropriate project approach
Choosing technologies, technical plan contents list, choice of process models, structure versus speed of delivery, the waterfall model, V-process model, the spiral model, prototypes, incremental delivery Software efforts estimation, Where Estimate done? Problems with over and under estimates, the basis for software estimating, software effort estimation techniques, Expert judgment, estimating by analogy, Albrecht function point analysis, function points mark II, COSMIC full function points, COCOMO : a parametric model

Unit 3: Activity Planning **(9 Hrs)**

Introduction, Objectives of Activity Planning, When to Plan, Project Schedules, Projects and Activities, Sequencing and Scheduling Activities, Network Planning Models, Formulating a Network Model, Adding the Time Dimension, The Forward Pass, Backward Pass, Identifying the Critical Path, Activity Float, Shortening the Project Duration, Identifying Critical Activities, Activity-on-Arrow Networks.
Risk Management: Introduction, Risk, Categories of Risk, Risk Management Approaches, A Framework for Dealing with Risk, Risk Identification, Risk Assessment, Risk Planning, Risk Management, Evaluating Risks to the Schedule, Boehm"s Top 10 Risks and Counter, Measures, Applying the PERT Technique, Monte Carlo Simulation, Critical Chain Concepts.
Resource Allocation: Introduction, Nature of Resources, Identifying Resource Requirements, Scheduling Resources, Creating Critical Paths, Counting the Cost, Being Specific, Publishing the Resource Schedule, Cost Schedules, Scheduling Sequence.

Unit 4: Monitoring and Control **(6 Hrs)**

Introduction, Creating the Framework, Collecting the Data, Review, Visualizing Progress, Cost Monitoring, Earned Value Analysis, Prioritizing Monitoring, Getting the Project Back to Target, Change Control, Software Configuration Management (SCM). Managing Contracts: Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management, Acceptance. Managing People in Software Environments: Introduction, Understanding Behaviour, Organizational Behaviour: A Background, Selecting the Right Person for the Job, Instruction in the Best Methods, Motivation, The Oldham–Hackman Job Characteristics Model, Stress, Stress Management, Health and Safety, Some Ethical and Professional Concerns.

Unit 5: IT Project Team **(9 Hrs)**

Working in Teams: Introduction, becoming a Team, Decision Making, Organization and Team Structures, Coordination Dependencies, Dispersed and Virtual Teams, Communication Genres, Communication Plans, Leadership.

Unit 6: Software Quality **(8 Hrs)**

Introduction, The Place of Software Quality in Project Planning, Importance of Software Quality, Defining Software Quality, Software Quality Models, ISO 9126, Product and Process Metrics, Product versus Process Quality Management, Quality Management Systems, Process Capability Models, Techniques to Help Enhance Software Quality, Testing, Software Reliability, Quality Plans.
Project Closeout: Introduction, Reasons for Project Closure, Project Closure Process, Performing a Financial Closure, Project Closeout Report.

3.2 Tutorials **(16 Hrs)**

Unit-specific exercises focused on problem-solving.

3.3 Laboratory Works: **(32 Hrs)**

The practical work includes implementing the concepts of cryptographic algorithms, hash functions, digital signatures. The practical work extends implementation and simulation of Network Security Protocols, Intrusion Detection Systems, DDoS Attacks, Cloud Security and IoT Security Systems. Students are free to use any of the language, tools and platform as per the skills.

4. Evaluation Scheme

- Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	**Components**	**Weightage**	**Full Marks**	**Pass Marks**
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books:

- Bob Hughes, Mike Cotterell, Rajib Mall. *Software Project Management*. TMH, 6th edition, 2018.
- Shailesh Mehta. *Project Management and Tools & Technologies – An Overview*. SPD, 1st edition, 2017.
- Walker Royce. *Software Project Management*. Pearson, 2005.

Machine Learning

Course Code	BAI402	Year/Semester	IV/VII
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course is an advanced exploration of Machine Learning, designed to provide students with a deeper understanding of machine learning techniques and a wider variety of existing learning models. Building upon the foundational concepts covered in the Machine Learning course, students will delve into cutting-edge methods and applications in the field. Through a combination of lectures, practical exercises, and laboratory sessions, students will develop the skills and knowledge necessary to develop advanced machine learning applications and engage in research at a state-of-the-art level.

2. Course Objectives

The course builds on the content of Machine Learning, providing students with a deeper understanding of machine learning techniques and a wider variety of extant learning models. Students will be prepared to develop advanced machine learning applications and perform research at a state-of-the-art level.

3. Course Details

3.1 Theory **(48 hrs)**

Unit 1: Introduction **(8 hrs)**

Introduction to machine Learning, Supervised Learning: Linear regression, logistic regression and generalized linear models, Generative probabilistic models, convex optimization and quadratic programming, support vector machines, Decision Trees and ensemble models, non-parametric methods.

Unit 2: Neural Network **(8 hrs)**

Perceptron's and inspiration from neuroscience, Multilayer neural networks and backpropagation, Optimization techniques, best practices, loss curve analysis, Inception modules, Residual Layers, squeeze and excitation, Detection models, semantic segmentation models, instance aware segmentation models.

Unit 3: Unsupervised Learning **(5 hrs)**

Clustering: k-means, Gaussian mixture models, principal component analysis, independent components analysis, Autoencoder.

Unit 4: Reinforcement Learning **(7 hrs)**

Markov decision processes and Bellman equations, Value iteration, policy iteration, Q-learning, Inductive transfer learning, Transudative transfer learning, Unsupervised transfer learning. Automated feature engineering, automated model selection and automated optimization algorithm selection.

Unit 5: Deep Unsupervised Learning **(6 hrs)**

Generative adversarial networks (GANs), Cycle GANs, Wasserstein GANs, Variational autoencoders, weight initialization, Dropout, Adam optimization, Batch normalization.

Unit 6: Time Series Processing **(8 hrs)**

Hidden Markov models (HMMs), Recurrent neural Networks (RNNs) and backpropagation through time, Word embedding for natural language processing, long short time memory (LSTM) units, Gated recurrent units (GRUs), Attention mechanisms for RNNs.

Unit 7: Deep Reinforcement Learning **(6 hrs)**

Policy gradients, Actor/critic methods, limitation learning, Exploration/exploitation, Meta learning, Monte Carlo methods.

3.2 Tutorials

Unit-specific exercises focused on problem-solving.

3.3 Laboratory Works

- Speech recognition
- Speech synthesis
- Conversational agents
- Recommendation systems
- Anomaly detection
- Computer vision system

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books:

- "Pattern Recognition and Machine Learning" by Christopher M. Bishop

- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
- "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G.
- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron -
- "Pattern Classification" by Richard O. Duda, Peter E. Hart, and David G. Stork

Block Chain Technology

Course Code	BAI403	Year/Semester	IV/VII
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course provides a comprehensive introduction to Blockchain Technology, a revolutionary decentralized ledger system that underpins cryptocurrencies like Bitcoin and Ethereum. Students will delve into the foundational concepts of blockchain, its architecture, and the mechanisms that ensure security and immutability. The course covers various real-world applications of blockchain beyond cryptocurrencies, including supply chain management, voting systems, smart contracts, and decentralized finance (DeFi). Through practical examples, case studies, and hands-on exercises, students will gain a deep understanding of blockchain's potential, challenges, and the future impact it may have on various industries.

2. Course Objectives

- Understand the fundamental concepts of blockchain technology, including its history, principles, and components.
- Explore the architecture and working mechanisms of different blockchain systems.
- Analyze the security features of blockchain networks and evaluate their strengths and weaknesses.
- Examine the consensus algorithms used in blockchain networks and their impact on scalability and decentralization.
- Investigate real-world applications of blockchain in various sectors such as finance, healthcare, supply chain, and more.
- Gain practical experience in setting up and interacting with blockchain networks.
- Identify potential use cases for blockchain technology and evaluate its advantages and limitations in different scenarios.
- Discuss the legal, ethical, and regulatory challenges associated with blockchain implementation.
- Collaborate in groups to develop innovative blockchain-based solutions to real-world problems.
- Stay informed about the latest trends and developments in the ever-evolving field of blockchain technology.

3. Course Details

3.1 Theory (48 hrs)

Unit 1: Introduction to Blockchain **(3 hrs)**

Definition and characteristics of blockchain technology; Brief history and evolution of blockchain; Decentralization, transparency, and immutability.

Unit 2: Cryptography and Consensus Mechanisms **(5 hrs)**

Cryptographic principles used in blockchain (hashing, digital signatures, etc.); Public and private key cryptography; Consensus algorithms (Proof of Work, Proof of Stake, etc.).

Unit 3: Blockchain Architecture and Data Structure **(4 hrs)**

Understanding the blockchain data structure; Blocks, transactions, and Merkle trees; Blockchain nodes and their roles (miners, validators, etc.).

Unit 4: Ethereum and Smart Contracts **(5 hrs)**
Introduction to Ethereum blockchain; Ethereum Virtual Machine (EVM) and smart contract execution; Writing, deploying, and interacting with smart contracts.

Unit 5: Solidity Programming **(5 hrs.)**
Solidity basics and syntax; Handling data and variables in Solidity; Implementing control structures and functions.

Unit 6: Advanced Smart Contracts **(5 hrs.)**
Using OpenZeppelin for secure smart contract development; Design patterns for smart contracts; Developing complex smart contracts (e.g., ERC-20 token, multisig wallet).

Unit 7: Decentralized Applications (DApps) Development **(5 hrs.)**
Frontend development with web3.js or similar libraries; Integrating DApps with smart contracts; User interactions and Metamask integration.

Unit 8: Blockchain Interoperability **(4 hrs.)**
Understanding cross-chain communication; Interoperability protocols and solutions; Building interoperable applications.

Unit 9: Security and Privacy in Blockchain **(3 hrs.)**
Common security threats in blockchain networks; Best practices for secure smart contract development; Privacy considerations and solutions in blockchain.

Unit 10: Scalability and Performance **(3 hrs.)**
Challenges of blockchain scalability; Layer 2 scaling solutions (state channels, sidechains); Exploring sharding and other scalability approaches.

Unit 11: Real-world Blockchain Applications **(3 hrs.)**
Use cases in finance, supply chain, healthcare, and more; Case studies of successful blockchain implementations.

Unit 12: Future Trends and Emerging Technologies **(3 hrs.)**
Ethereum 2.0 and its impact on the blockchain ecosystem; DeFi (Decentralized Finance) and NFTs (Non-Fungible Tokens); Exploring emerging blockchain technologies.

3.2 Tutorials **(16 hrs)**

Unit-specific exercises focused on problem-solving.

3.3 Laboratory Works **(32 hrs)**

Lab 1: Setting Up the Development Environment
Objective: Install and configure the necessary tools and libraries required for blockchain development.
Tasks:

- Install a blockchain client (e.g., Geth, Parity, or Besu) on your local machine.
- Set up an Ethereum development environment using Remix IDE or Truffle.
- Configure MetaMask and connect it to your local development network.
- Create a new Ethereum account using MetaMask and fund it with test Ether.

Lab 2: Writing and Deploying Smart Contracts
Objective: Learn the basics of writing and deploying smart contracts on the Ethereum blockchain.
Tasks:

- Write a simple smart contract in Solidity that stores and retrieves data.

- Compile the smart contract using the Ethereum development environment.
- Deploy the smart contract to your local development network using Remix or Truffle.
- Interact with the deployed smart contract through Remix or a JavaScript frontend.

Lab 3: Creating a Decentralized Application (DApp)
Objective: Build a decentralized application (DApp) that interacts with a deployed smart contract.
Tasks:

- Design a simple DApp using HTML, CSS, and JavaScript.
- Integrate web3.js or ethers.js library to interact with the deployed smart contract.
- Implement functionalities such as reading data from the smart contract and sending transactions.
- Test the DApp on your local development network using MetaMask.

Lab 4: Implementing Smart Contract Security
Objective: Identify and fix common vulnerabilities in smart contracts.
Tasks:

- Explore common smart contract vulnerabilities (e.g., reentrancy, integer overflow, etc.).
- Analyze a vulnerable smart contract and identify its weaknesses.
- Implement security fixes to address the identified vulnerabilities.
- Re-deploy the updated smart contract and test its security.

Lab 5: Exploring Blockchain Interoperability
Objective: Learn about cross-chain communication and interoperability between different blockchain networks.
Tasks:

- Set up a test environment with two blockchain networks (e.g., Ethereum and Binance Smart Chain).
- Create a bridge between the two networks using a cross-chain communication protocol.
- Transfer assets (e.g., tokens) from one network to another using the bridge.
- Verify the successful transfer of assets and ensure consistency across both networks.

Lab 6: Blockchain Scalability and Layer 2 Solutions
Objective: Explore different Layer 2 scaling solutions and their impact on blockchain scalability.
Tasks:

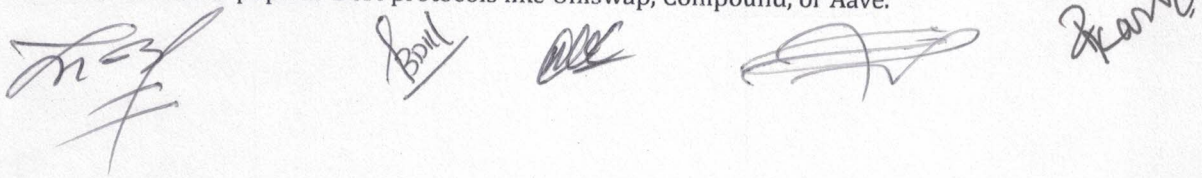
- Investigate the scalability challenges faced by the Ethereum blockchain.
- Learn about Layer 2 solutions such as state channels or sidechains.
- Implement a simple state channel for off-chain transactions.
- Measure the performance improvement in terms of transaction throughput and gas fees.

Lab 7: Building an Ethereum Token
Objective: Create a custom ERC-20 token and deploy it on the Ethereum blockchain.
Tasks:

- Design the token contract in Solidity following the ERC-20 standard.
- Deploy the token contract to the Ethereum blockchain.
- Interact with the token contract using Remix or a DApp frontend.
- Test token transfers and verify token balances using MetaMask.

Lab 8: Exploring DeFi (Decentralized Finance) Applications
Objective: Explore and interact with decentralized finance (DeFi) applications on the Ethereum blockchain.
Tasks:

- Learn about popular DeFi protocols like Uniswap, Compound, or Aave.



- Connect to the DeFi protocols using MetaMask.
- Engage in token swaps, lending, or borrowing activities within the DeFi applications.
- Analyze the risks and benefits of using DeFi applications.

Lab 9: Ethereum Smart Contract Upgrades
Objective: Understand how to upgrade smart contracts while preserving data and user interactions.
Tasks:

- Study the challenges of upgrading smart contracts.
- Implement a smart contract with upgradable features using a proxy pattern.
- Perform upgrades on the upgradable smart contract, ensuring data and functionality persistence.
- Test the upgraded contract with various scenarios.

Lab 10: Final Project
Objective: Apply the knowledge gained throughout the laboratory work to develop a full-fledged blockchain application or smart contract project.
Tasks:

- Choose a suitable project idea related to blockchain technology (e.g., NFT marketplace, decentralized Digital voting system, Transparent Budgeting, Record Management, Healthcare, Media and Entertainment, Real State, supply chain tracking, Finance, Energy).
- Design the architecture and functionalities of the project.
- Implement the project using best practices and security measures.
- Present the final project to the class, explaining its features and demonstrating its functionality.

Note: The laboratory work can be adjusted based on the specific blockchain platform or framework being used in the course. Additionally, students may work in teams for certain tasks and collaborate on larger projects for the final assignment.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	20	20	10
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Recommended Books

Textbooks:

- "Mastering Blockchain: A Hands-On Approach to Distributed Ledger Technology" by Imran Bashir
- "Blockchain Basics: A Non-Technical Introduction in 25 Steps" by Daniel Drescher

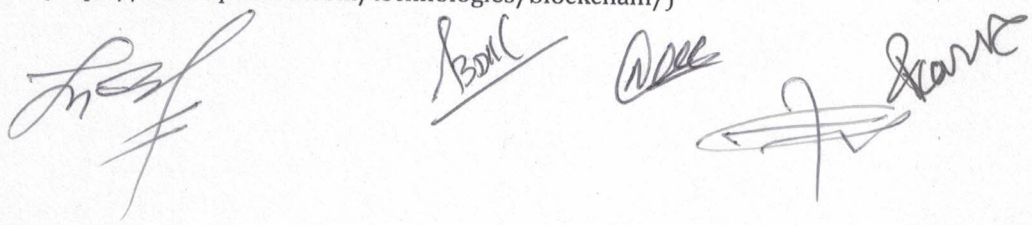
Reference Books:

- "Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World" by Don Tapscott and Alex Tapscott

- "Blockchain for Dummies" by Tiana Laurence
- "Blockchain: Blueprint for a New Economy" by Melanie Swan

Learning Websites:

- (https://www.coursera.org/browse/computer-science/blockchain)
- (https://www.edx.org/learn/blockchain)
- (https://academy.b9lab.com/)
- (https://www.blockchain-council.org/)
- (https://developer.ibm.com/technologies/blockchain/)



Capstone Project

Course Code	BAI445	Year/Semester	IV/VII
Credit Weightage	5	Class Load	10 hrs/wk
		Practical	10 hrs/wk
		Total	160 hrs

1. Course Description

This is a fully practical course that expects students to implement concepts learned during the first three years of their study. The system should not be limited to the basic CRUD operations only, they are highly recommended to implement appropriate algorithms relevant to the project. The project should include precise system analysis, design, implementation, and result analysis. The students can choose appropriate language and technologies that they have learn till seven semester.

2. Course Objectives

The objective of this project work is to make the student able to:

- Lead a software project development
- Implement appropriate algorithms in the project
- Work in a team
- Use CASE tools
- Write programs and improve programming skills
- Write test cases for software testing and improve QA skills
- Enhance problem-solving, report writing, and presentation skills

3. Nature of Project

- It is an academic project focused on development of a desktop based, web based, or mobile based application using any programming language and technology of their expertise and comfort
- Students can develop the application not limited in CRUD operations but also have implement appropriate algorithm.
- Student should use appropriate CASE tools during SDLC
- While implementing the project, students should be encouraged to write their own modules rather than relying on APIs or Plugins (except in some unavoidable circumstances)
- Emphasis on problem-solving, system design, and implementation
- Project group size: Maximum 2 students in a group but highly recommended to do individually.

4. Focus of the Study

Each student in a group (in case of group) should have equal participation in every phase of the project. The students should focus on the each phases of SDLC in detail.

5. Phases of the Project Work

5.4. Proposal Submission and defense

- Timeline: 3rd - 4th week of the 7th Semester

5.5. Mid-Term Progress Report Evaluation and Defense

- Timeline: 11th –12th week of the 7th Semester

5.6. Final Submission and Defense

- Final report submission: At least 10 days before the final defense date

- Defense includes viva, presentation, project demonstration, and report evaluation

6. Supervision

- Supervised by a regular faculty member
- One supervisor may guide up to two groups or four individual

7. Evaluation Criteria

Total Marks: 100

Evaluation Phase	Evaluators	Marks
Proposal Defense	HoD/Coordinator (4), Supervisor (12), Internal Examiner (4)	20
Mid-Term	HoD/Coordinator (8), Supervisor (24), Internal Examiner (8)	40
Final Defense	HoD/Coordinator (5), Supervisor (15), Internal Examiner (5), External Examiner (15)	40

8. Evaluation Focus on

- Presentation Skills
- Technical Work Quality
- Documentation Quality
- Project Report
- Viva Performance
- Project Demo
- Level of work understanding

9. Roles & Responsibilities

Role	Responsibilities
HoD/Coordinator	Oversee project schedule, arrange defenses, evaluate at all stages
Supervisor	Guide, monitor, and evaluate the project work throughout
Internal Examiner	Evaluate at proposal, midterm, and final defense
External Examiner	Evaluate final presentation, viva, and demo of project
Student	Active in project development, documentation, and defense. Keep supervision logs

14. Report Contents

14.1. Prescribed Content Flow for the Project Proposal

8. Introduction
9. Problem Statement
10. Objectives
11. Methodology
    a. Requirement Identification

i. Study of Existing System
ii. Literature Review
iii. Requirement Analysis

b. Feasibility Study
i. Technical
ii. Operational
iii. Economic

c. High Level System Design
i. System Flow Chart
ii. Methodology of Proposed System
iii. Working Mechanism of Proposed System

12. Project Timeline (Gantt Chart)
13. Expected Outcome
14. References (IEEE Format)

14.2. Prescribed Content Flow for the Project Report

11. Cover & Title Page
12. Certificate Page
    a. Supervisor Recommendation
    b. Internal and External Examiners' Approval Letter
13. Abstract
14. Acknowledgement
15. Table of Contents
16. List of Abbreviations, List of Figures, List of Tables
17. Main Report (Chapters 1–5)
18. References (IEEE Format)
19. Bibliography (if any)
20. Appendices (Screenshots, Source Code, Supervisors Log Sheets)

14.3. Prescribed Chapters in Main Report

Chapter 1: Introduction

1.6. Introduction
1.7. Problem Statement
1.8. Objectives
1.9. Scope and Limitation
1.10. Development Methodology
1.11. Report Organization

Chapter 2: Background Study and Literature Review

2.3. Background Study (Description of Fundamental theories and concepts of related project)
2.4. Literature Review (Review of the similar project, theories, research done by other researchers)

Chapter 3: System Analysis and Design

3.2. System Analysis
3.2.1. Requirement Analysis
iii. Functional Requirement (Use-case Diagram, Use-case Description)
iv. Non Functional Requirement
3.2.2. Feasibility Analysis
v. Technical

vi. Operational
vii. Economic
viii. Schedule

3.2.3. System Modelling
3.2.3.1. Object Modelling: Class and Object Diagram
3.2.3.2. Dynamic Modelling: State and Sequence Diagram
3.2.3.3. Process Modelling: Activity Diagram
3.3. System Design
3.3.1. Refinement of Class, Object, State, Sequence, and Activity Diagrams
3.3.2. Component Diagrams
3.3.3. Deployment Diagrams
3.4. Algorithm Details

Chapter 4: Implementation and Testing

4.3. Implementation
4.3.1. Tools Used (CASE tools, Programming Language, Database Platforms)
4.3.2. Implementation Details of Modules (Description of classes, procedures, functions, methods, algorithms)
4.4. Testing
4.4.1. Test Case for Unit Testing
4.4.2. Test Case for System Testing
4.5. Result Analysis

Chapter 5: Conclusion and Future Recommendations

5.4. Lesson Learnt and Outcome
5.5. Conclusion
5.6. Future Recommendations

Note: *While writing above chapters students should avoid basic definitions. They should relate and contextualize the above mentioned concepts with their project work.*

15. Citation and Referencing

The listing of references should be listed in the references section. The references contain the list of articles, books, url of primary source that are cited in the document. The books, articles, and others that are studied during the literature review and project work but are not cited in the document can be listed in the bibliography section.
The citation and referencing standard should be **IEEE** referencing standard.

16. Report Format Standards

- **Page size:** A4
- **Font:** Times New Roman, 12pt (body), headings bold (16pt/14pt/12pt).
- **Spacing:** 1.5 line, justified.
- **Margins:** Top/Bottom/Right = 1", Left = 1.25".
- **Page Numbers:** Roman (i, ii...) for preliminaries, numeric (1, 2...) from Chapter 1.
- **Headings:** Chapter – 16pt (Bold and center align), Section – 14pt (Bold and left align), Sub-section – 12pt (Bold and left align), Paragraph content – 12pt (Regular)
- **Figures/Tables:** Center-align, Captions – 12pt (Bold)

17. Binding & Submission

- Copies: 3 (Dean Office, College Library, Self)
- Binding: Golden Embracing with Black Book Binding
- Final Copy: Signed final copy should be submitted to the Dean Office and respective department

(Sample of Cover and Title Page)

Lumbini Technological University

Institute of Engineering and Information Technology

TITLE OF PROJECT REPORT

A PRPJECT REPORT

Submitted to

Department of B.Tech in Information Technology

Name of the Collage

In partial fulfillment of the requirement for the B.Tech in Computer Science and Artificial Intelligence

Submitted by

Name:

Registration No.:

Symbol No.:

Month, Year

Under the Supervision of

Supervisor Name

Designation

(Sample of Certificate)

Lumbini Technological University

Institute of Engineering and Information Technology

College Name

Supervisor's Recommendation

I, hereby recommend that this project is completed under my supervision by NAME OF THE STUDENT (Year IV Semester: VII) entitled **"TITLE OF THE PROJECT"** in partial fulfillment of the requirements for the degree of B.Tech in Computer Science and Artificial Intelligence is recommended for the final evaluation.

.....
Signature
Name of Supervisor
Designation of Supervisor
Department Name
Address of the college

(Sample of Approval)

Lumbini Technological University

Institute of Engineering and Information Technology

College Name

LETTER OF APPROVAL

This is to certify that this project is completed by NAME OF THE STUDENT (Year IV Semester: VII) entitled "**TITLE OF THE PROJECT**" in partial fulfillment of the requirements for the degree of B.Tech in Computer Science and Artificial Intelligence has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

.....
Signature
Name of Supervisor
Academic Designation
Department Name
Address of the college

.....
Signature
Name of HOD/Program Coordinator
Academic Designation
Department Name
Address of the college

.....
Signature
Name of Internal Examiner
Academic Designation

.....
Signature
Name of External Examiner
Academic Designation

Year IV Semester VIII

IT Entrepreneurship

Course Code	BAI451	Year/Semester	IV/VIII
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	2 hrs/wk
		Practical	N/A
		Total	80 hrs

1. Course Description

This course contains the fundamental concept of entrepreneurship, growth of entrepreneurship, entrepreneurial opportunities, entrepreneurship competencies development, business plan concept and elements of business and business plan evaluation.

2. Course Objectives

- To understand how emerging trends in information technology and innovation affect business processes and potentially create value (Emerging Technologies).
- To utilize the fundamentals of business process innovation and how to manage business process innovation initiatives and process configurations to impact business agility

3. Course Details

(48 hrs.)

3.1 Theory

Unit 1: Introduction of Business process and Information System **(7 hrs)**

Organization process, Flows in business process, Monitor process performance, Application Infrastructure, Information system and business process, Importance of information system (data and information, Functional information system), Functional organizational structure (Delay in execution the process, excess inventory, lack of visibility across processes)

Unit 2: Enterprise System **(7 hrs)**

Role of enterprise system in organization, Execute the process, Capture and store process data, Stand-alone mainframe systems, Client server architecture, Service Oriented Architecture, Types of Enterprise system, Types of Data in enterprise system (transaction data, master data, organizational data)

Unit 3: IT and Strategy **(7 hrs)**

Information revolution, Business and Strategy, Information Technology Strategy, Strategies and success, Design parameters and Strategic positioning, Evolution and development of strategy, Strategic planning and IT strategies, Evolving a dynamic nature of the Business

Unit 4: Managing IT **(7 hrs)**

IT management and Its Role, IT governance and infrastructure, IT Governance and Strategy, Technology Management Process, Steps in Technology Management, Strategic Aspects of IT and Positioning the company for change, IT and business alignment, Risk management, Implementing and Exploiting IT capabilities, Using IT in a Strategic Manner, Measuring IT with Performance measures and Balanced Score card, Implementing change in IT management

Unit 5: E – Strategy **(5 hrs)**

Introduction, E business and E-strategy, Developing an E-strategy, E-business objectives, E-commerce and E-business, Business –model and E-business model, Making E-strategy and E-economy, Best practice and Competitive advantage

Unit 6: IT strategy for IT companies **(5 hrs)**

Strategic aspects for an IT product companies and IT strategic development, IT strategy and innovation driving factor of start-up product companies, IT strategies for product life cycle and dealing with chasm, Project life cycle ad Strategies at various stages, Technology Selection and IT strategic aspects, Technology change management

Unit 7: Emerging Entrepreneurship in Nepal **(10 hrs)**

Micro, small and medium size enterprise: concept and characteristics, importance and contribution to employment generation and resource mobilization in the economic development of Nepal. Indigenous knowledge and major ethnic entrepreneurship- its problems and prospects; The emerging entrepreneurship in Nepal: Rural entrepreneurship, Tourism entrepreneurship, Agri-entrepreneurship, Transport entrepreneurship.

3.2 Project Work:

- Develop a business Strategy for a small IT company which focuses on building a software product and services and develop a IT strategy for a company
- Develop a risk management Plan
- Preparation of e- business model for an ISP company
- Develop a E strategy for a company for E commerce site
- Case study for knowledge management

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	15		
	Seminar/Project/Presentation	20		
	Practical/Lab Examination	-	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books

- Kulkarni Parag,Chande k.pradip, IT strategy for Business ,Oxford-2008
- Barringer, B.R. and Ireland, R. D. Entrepreneurship: Successfully Launching New Business. Pearson, Delhi, India.

Charantimath, P.M. Entrepreneurship Development and Small Business Enterprises. Pearson, Delhi, India

Natural Language Processing

Course Code	BAI452	Year/Semester	IV/VIII
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

The course covers the introductions, methods and approaches used in many real-world NLP applications such as Computational Linguistics, Morphology, Syntax, Semantics, Discourse.

2. Course Objectives

- To Provide the students a general overview of the basics as well as the advanced concepts of Natural Language Processing (NLP)
- To know the role of semantics of sentences and pragmatic.
- To apply the different concepts of NLP both theoretically and practically.

3. Course Details

(48 hrs.)

3.1 Theory

Unit 1: Introduction to NLP **(6hrs)**

Introduction to NLP, origins and challenges of NLP – language modeling: grammar-based lm, statistical lm – regular expressions, finite-state automata – english morphology, transducers for lexicon and rules, tokenization, detecting and correcting spelling errors, minimum edit distance

Unit 2: Word Level Analysis **(8 hrs)**

Unsmoothed n-grams, evaluating n-grams, smoothing, interpolation and backoff – word classes, part-of-speech tagging, rule-based, stochastic and transformation-based tagging, issues in pos tagging – hidden markov and maximum entropy models.

Unit 3: Syntactic Analysis **(10 hrs)**

Context free grammars, grammar rules for english, treebanks, normal forms for grammar – dependency grammar – syntactic parsing, ambiguity, dynamic programming parsing – shallow parsing – probabilistic cfg, probabilistic cyk, probabilistic lexicalized cfgs – feature structures, unification of feature structures.

Unit 4: Lexical Semantics **(6 hrs)**

Lexical Semantics, Lexeme, Lexicon, Senses, Lexical relations, WordNet (Lexical Database), Word Sense Disambiguation (WSD), Word Similarity.

Unit 5: Discourse **(8 hrs)**

Pragmatic & Discourse Analysis, Monologue and Dialogue, Reference Resolution, Coherence and Cohesion, Discourse Structure.

Unit 6: Application of NLP **(10 hrs)**

Applications of NLP, Question Answering, Machine Translation, Sentiment Analysis, Summary Generation.

3.7 Laboratory Works:

In the lab and practical works, the students will basically get practical concepts of NLP in the Python Programming Language. A lot of these would be hands-on exercises and writing the codes of NLP problem- solving.

3.3. Tutorials

Unit specific problem solving exercise.

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books:

- Daniel Jurafsky and James H. Martin (2009). Speech and Language Processing, Second Edition, Pearson Education.
- Frederick Jelinek, "Statistical Methods Of Speech Recognition", MIT Press, 1997.

Elective I
Year III Semester VI

Integrated GIS and Remote Sensing

Course Code	Elective I	Year/Semester	III/VI
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This integrated course covers fundamental principles, technologies, and applications of Geographic Information Systems (GIS) and Remote Sensing (RS). Students will learn spatial data acquisition, analysis, and visualization techniques, with hands-on experience in solving real-world environmental, urban, and resource management challenges.

2. Course Objectives

By the end of this course, students should be able to:

- Understand core concepts of GIS and RS, including data models, platforms, and sensors.
- Process and analyze geospatial data using appropriate techniques.
- Integrate GIS and RS for applications like land-use change detection and disaster management.
- Design effective cartographic outputs and spatial decision-support systems.

3. Course Details

(48 hrs.)

3.1 Theory

Unit 1: Foundations of Geospatial Technologies **(6 hrs)**

GIS Concepts: Definitions, history, components (hardware, software, data, people); RS Fundamentals: Electromagnetic spectrum, energy-matter interaction, resolutions; Integrated Applications: Smart cities, precision agriculture, climate change studies.

Unit 2: Data Models and Acquisition **(6 hrs)**

Types of data, Geographical data, major sources of GIS data; Spatial and non-spatial data, Data linkage; GIS Data Models: Vector (spaghetti, topological), raster, TIN; RS Data Sources: Satellites (Landsat, Sentinel), UAVs, LiDAR; Data Quality: Accuracy, precision, metadata (ISO 19115).

Unit 3: Coordinate Systems and Georeferencing **(7 hrs)**

Projections: Mapping concepts, coordinate systems, UTM, Lambert Conformal Conic; datum transformations; Georeferencing: Ground control points, Rectification; GPS Integration: GPS basics, components of GPS, GPS segments, Differential GPS, RTK surveys.

Unit 4: Spatial Databases **(7 hrs)**

Data concept and design, database design, data processing system, Database Table; Database Design: Hierarchical database, network database, relational database (PostgreSQL/PostGIS), NoSQL for geospatial; Normalization: 1NF to 3NF for spatial data and data security; Spatial SQL: Queries, indexing.

Unit 5: Remote Sensing Processing **(8 hrs)**

Image Preprocessing: Radiometric/atmospheric correction; Classification: Supervised (SVM), unsupervised (K-means); Change Detection: Time-series analysis (NDVI, NDWI, NDBI etc, trends).

Unit 6: Advanced Geospatial Analysis **(8 hrs)**

GIS Analysis: Network routing, hydrological modelling; RS Analysis: Spectral indices (NDVI, NDWI), object-based image analysis (OBIA); Integration: RS-to-GIS data conversion (e.g., raster to vector).

Unit 7: Map output and Information Dissemination **(6 hrs)**

Cartography and GIS; Map Layout Elements; Map Design: Color theory, visual hierarchy; Types of Maps; Cartographic output Format.

3.2 List of Tutorials:

The various tutorial activities that suits this course should cover all the content of this course to give student a space to engage more actively with the course content in the presence of instructor. Students should submit tutorials as assignments or class-works to the instructor for evaluation.

A. Discussion-based Tutorials:

1. GIS vs. Remote Sensing: Compare/contrast applications in environmental monitoring.
2. Vector vs. Raster Data Models: Debate advantages for urban planning vs. forestry.
3. Open-Source vs. Proprietary Geospatial Tools: Ethical and technical implications.
4. AI in Geospatial Analysis: Discuss ML applications in satellite image classification.

B. Problem-solving-based Tutorials:

5. Coordinate Transformations: Convert WGS84 to UTM coordinates for given points.
6. SQL for Spatial Queries: Solve queries (e.g., "Find hospitals within 1 km of flood zones").
7. Spectral Indices Calculation: Compute vegetation indices from sample spectral bands.
8. Topology Error Fixing: Correct gaps/overlaps in polygon datasets.
9. Network Analysis: Optimize emergency routes using Dijkstra's algorithm.
10. Image Classification: Classify a Landsat scene using maximum likelihood.
11. Cartographic Design: Critique map layouts based on visual hierarchy principles.

3.3 Laboratory Works

- Fundamentals of Geospatial Data Handling
- Coordinate Systems and Georeferencing
- Spatial Database Creation
- Remote Sensing Image Processing
- Vector-Based Spatial Analysis
- Raster-Based Terrain Analysis
- Integrated GIS-RS Application
- Cartographic Design and Visualization

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books

1. Chang, K. (2019). Introduction to Geographic Information Systems (9th edition). New York: McGraw-Hill Education, Penn Plaza.
2. Bolstad, P. (2016). GIS Fundamentals: A First Text on Geographic Information Systems (5th edition). United States: Eider Press.
3. Jensen, J. R., Jensen, R. R. (2013). Introductory Geographic Information Systems. Boston: Pearson.
4. Lillesand, T., Kiefer, R. W., & Chipman, J. (2015). Remote Sensing and Image Interpretation. Wiley.
5. Jensen, J. R. (2015). Introduction to Remote Sensing. CRC Press.
6. Campbell, J. B., & Wynne, R. H. (2011). Introduction to Remote Sensing. Guilford Press.
7. Good child, Loiigley, Maguire, "Geographic Information System and Science"
8. A complete coiu'se manual in Geogi'aphic Information System, nec
9. ICIMOD "Training manual on GIS application and Remote sensing"

Data Visualization and Interpretation

Course Code	Elective I	Year/Semester	III/VI
Credit Weightage	3	Lecture	2 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	80 hrs

1. Course Description

This course introduces the principles and techniques of data visualization and interpretation within the context of data science. It aims to equip students with the knowledge and skills necessary to transform data into meaningful visual narratives that facilitate analytical reasoning and decision-making. Through hands-on projects and case studies, students will learn to apply visualization tools and software to real-world data sets, gaining insights into the role of visualization in data science.

2. Course Objectives

Upon completion of this course, students will be able to:

- Select appropriate visualization techniques for various types of data and analysis goals with understanding of principles and aesthetics of efficient visualization
- Use leading data visualization tools and software to create impactful visual representations of data for different use cases
- Communicate complex data insights through compelling visual narratives.
- Apply ethical considerations in the visualization and interpretation of data.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completion of this course, students will be able to:

- Explain the importance of data visualization and Distinguish between exploratory and explanatory analysis.
- Identify and use different elements of data visualization that are used in procedural and declarative data visualization
- use tools such as matplotlib, vega-altair or ggplot, to plot different types of data plot as need
- apply Gestalt principles of visual perception for designing different visualization without producing chart junk
- Conduct univariate, bivariate and multivariate analysis using different types of plots
- Create exploratory visualization and appropriate narrative for storytelling with data
- develop interactive visualization in dashboards using certain dashboarding tools such as looker studio, Power BI, Streamlit, or other tools
- analyze and recognize bias in data and visualization and use appropriate guidelines for ethical data visualization designs.

4. Course Details

(32 hrs)

Unit 1: Introduction to Data Visualization **(2hrs)**

Importance of Data Visualization; Multiple interpretations of Data and Visualization; Exploratory vs Explanatory Analysis.

Unit 2: Visual Encoding and Graphical Foundations **(6 hrs)**

Data types; Elements of Visual Graphics; Pipeline of Graphic Generation; Procedural Graphic System: Matplotlib; Declarative system based on Grammar of Graphics: ggplot, Vega-Altair.

Unit 3: Visual Aesthetics Theory **(8 hrs)**

Design thinking in data visualization; Layout, colour, and typography; Gestalt Principles of Visual Perception; Data ink-ratio & Chart Junk.

Unit 4: Exploratory Data Analysis **(8 hrs)**

Univariate Analysis: Bar, Pie, Dot & Jitter, Histogram, KDE, Rank Order Plot, Pareto chart, CDF plot, Quantile Plot, Box-whisker Plots, Violin Plot; Bivariate Analysis: CrossTab, Scatter plot, Trend Lines, Correlations, Heatmap, Bivariate KDE, Violin; Multivariate Analysis: Faceting, Joint Plots, Plot Matrix, Non-positional encoding; Dimensionality reduction for data visualization (PCA, t-SNE).

Unit 5: Storytelling with Data **(4 hrs)**

Explanatory Analysis; Creating a Narrative for storytelling with data; Interactive Visualization; Dashboard Design Best Practices.

Unit 6: Responsible and Trustworthy Data Visualization **(4 hrs)**

Ethical considerations in Data Visualization (eg of Lying with Data); Types and source of Bias in Data and Visualization; Recognizing and avoiding bias; Case studies: Misleading visualizations and their impact; Guidelines for ethical visualization design.

5. Tutorial **(16 hrs)**

Unit specific problem solving exercise.

6. Practical **(32 hrs)**

a. Procedural Graphics Tool : Matlab and Seaborn (2hr)
b. Declarative Graphics Tool: ggplot or Vega-Altair (2hr)
c. Exploratory Data Analysis: (4hrs)
   i. Univariate Analysis
   ii. Bivariate Analysis
   iii. Multivariate Analysis
d. Interactive Dashboarding with Apache Superset or Google Looker Studio or MS Power BI or streamlit or any other dashboarding tool (2hr)
e. Advance Data Visualization Techniques (6hrs)
   i. Time series visualization with any visualization tool
   ii. Geospatial Visualization with Geoplot and geopandas
   iii. Network visualization with networkxx

7. Evaluation Scheme

Evaluation of student's performances is divided into two main parts: (1) Internal Evaluation and (2) Semester examination, with breakdowns as follows:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

8. Recommended Books

1. "The Visual Display of Quantitative Information" by Edward R. Tufte, Graphic Press 2nd Edition, 2007
2. "Storytelling with Data: A Data Visualization Guide for Business Professionals" by Cole Nussbaumer Knaflic, Wiley, 2015
3. "Data Points: Visualization That Means Something" by Nathan Yau
4. "Interactive Data Visualization for the Web" by Scott Murray (for those interested in web technologies)
5. "The Grammar of Graphics" by Leland Wilkinson, 1999

IoT and it's Application

Course Code	Elective I	Year/Semester	IV/VII
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	2 hrs/wk
		Total	96 hrs

1. Course Description

This course offers an accessible yet comprehensive introduction to the Internet of Things (IoT), designed for students with basic technical knowledge. It explores the core concepts, system components, communication technologies, and application areas of IoT. Emphasis is placed on hands-on skills using platforms such as **Arduino**, **Raspberry Pi**, and **IoT cloud services**. Through guided lectures, labs, and project-based learning, students will gain the competence to design, develop, and troubleshoot IoT systems in real-world contexts.

2. Course Objectives

The objectives of this course are:

- To understand the fundamental building blocks of an IoT system.
- To gain practical knowledge of sensors, actuators, microcontrollers, and communication protocols.
- To learn to interface hardware and software in real-time embedded systems.
- To explore IoT platforms and cloud services for data visualization and analytics.
- To apply programming logic using Python and C/C++ for IoT development.
- To develop skills in security fundamentals, including data privacy and encryption.
- To work collaboratively on real-life mini-projects or prototypes demonstrating IoT use cases.

3. Learning Outcomes

Upon successful completion, students will be able to:

- Define and explain the basic principles and architecture of IoT.
- Build and program IoT-enabled embedded devices using Arduino/Raspberry Pi.
- Develop simple IoT applications for smart environments (e.g., smart home, agriculture).
- Use cloud platforms to collect, store, analyze, and visualize data.
- Understand key network communication standards (Wi-Fi, Bluetooth, MQTT, etc.).
- Apply basic cybersecurity techniques to safeguard IoT systems.
- Collaborate in teams to prototype and debug IoT-based solutions.

4. Course Detail: **(48 hrs)**

4.1 Theory ***(6 hours)***

Unit 1: Introduction to IoT

Overview of IoT; Key concepts, history, and evolution; IoT architecture: Physical and logical views; Enabling technologies: Cloud, Big Data, AI, Embedded Systems; Communication protocols: MQTT, CoAP, HTTP, Bluetooth, Wi-Fi; IoT vs M2M vs CPS (Cyber-Physical Systems); Real-world applications: Smart Home, Health, Agriculture.

Unit 2: IoT Hardware and Platforms *(8 hours)*

Overview of popular boards: Arduino Uno/Mega, Raspberry Pi, NodeMCU; Sensor and actuator basics (temperature, humidity, IR, ultrasonic, etc.); Interfacing: LED, Push Button, Buzzer, LCD, Servo Motor; Development environments: Arduino IDE, Thonny for Python; Overview of popular IoT cloud platforms (Thingspeak, Blynk, Firebase, AWS IoT).

Unit 3: Communication & Networking in IoT *(10 hours)*

Data transfer protocols: TCP/IP, MQTT, REST; IPv4 vs IPv6, Wired vs Wireless Communication; GSM/3G/4G/5G, LoRa, Zigbee, BLE; Posting sensor data to web servers; Local data logging (SD cards) vs Cloud data storage; Simple server-client models using Python

Unit 4: Data Analytics & Visualization *(8 hours)*

Data preprocessing and cleaning for IoT; Edge vs Cloud analytics; Tools: Python (Pandas, Matplotlib), Excel, ThingSpeak, Grafana; Using real-time sensor data streams; Visualization dashboards using open-source platforms; Introduction to IoT data lifecycle

Unit 5: IoT Security Fundamentals *(8 hours)*

Overview of IoT threats: spoofing, malware, DoS; Basic encryption methods (AES, RSA); Authentication and access control; Secure communication protocols.

Case study: Securing smart home systems

Unit 6: Use Cases and Mini-Projects *(8 hours)*

Hands-on IoT design for: Smart Home Automation (lights, temperature, door lock); Smart Farming (soil moisture, water level); Smart Metering (AMR); Health monitoring (heartbeat, SpO2 sensor); Concept to prototype: design → build → test → present.

5. Practical Lab Work

Hands-on sessions covering:

- Sensor interfacing (digital and analog)
- Real-time data collection and analysis
- Arduino/Raspberry Pi programming (C/C++ and Python)
- Web dashboard creation for data visualization
- Cloud connectivity (e.g., Blynk/Thingspeak)
- Final IoT mini-project: Integrating sensors, data, UI, and security

6. Tutorial

Unit specific problem solving exercise.

7. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

8. Recommended Textbooks & Resources

- Simone Cirani, Gianluigi Ferrari, Marco Picone, Luca Veltri. Internet of Things: Architectures, Protocols and Standards. Wiley, 2019. p.383. ISBN 9781119359678.
- Perry Xiao. Designing Embedded Systems and the Internet of Things (IoT) with the ARM® Mbed™. Wiley, 2019. p.316. ISBN 9781119363996.
- ArshdeepBahga, Vijay Madisetti, " Internet of Things (A Hands-on-Approach)", University Press India Pvt. Ltd., 2015.
- David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Ro be rt Barton, Jerome Henry,"IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Pearson Education (Cisco Press Indian Reprint).
- Raj Kamal, "Internet of Things: Architecture and Design Principles", McGraw Hill Education, 2017.

Elective II
Year IV Semester VII

Application of GIS and Remote Sensing using Python

Course Code	Elective II	Year/Semester	IV/VII
Credit Weightage	3	Lecture	2 hrs/wk
		Tutorial	N/A
		Practical	3 hr/wk
		Total	80 hrs

1. Course Description

This course provides a comprehensive introduction to Python programming and its application in Geographic Information Systems (GIS). Students will learn fundamental programming concepts and how to use Python to process, analyze, and visualize spatial data. The course focuses on real-world geospatial applications including terrain modeling, spatial queries, coordinate transformations, and watershed analysis using libraries such as GeoPandas, Rasterio, Shapely, and GDAL.

2. Course Objectives

By the end of this course, students should be able to:

- Write and debug Python programs for data analysis and spatial processing.
- Apply spatial data structures and functions to manipulate and analyze geospatial data.
- Implement geoprocessing operations and raster analysis using Python libraries.
- Develop spatial workflows for advanced GIS applications such as watershed modeling, terrain analysis, and spatial statistics.
- Visualize spatial data using libraries such as Matplotlib, Seaborn, and Plotly

96 Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Remember Python syntax and core data structures (variables, lists, dictionaries, etc.).
- Understand spatial concepts such as projections, coordinate transformations, and geoprocessing.
- Apply spatial and raster analysis techniques using Python.
- Analyze terrain models, DEM data, and land-use patterns with statistical methods.
- Evaluate and optimize spatial workflows using coding best practices.
- Create GIS-integrated projects to solve real-world spatial problems (e.g., watershed delineation, urban heat island detection).

4. Course Details (32 hrs)

4.1. Theory

Unit 1: Introduction to Python Programming (4 hrs)

Variables, operators, expressions, input/output; Interactive vs. script mode, debugging; Data types: int, float, str, list, dict, conversions; String operations and basic file handling.

Unit 2: Functions, Control Flow, and Data Structures (4 hrs)

Defining functions, parameters, return types, recursion; Conditional statements (if, else), loops, list comprehensions; Exception handling; Core data structures: strings, lists, dictionaries, sets.

Unit 3: Advanced Python Programming (5 hrs)

File operations: open, read, write, file modes; Context managers (with statement); OOP: Classes, methods, inheritance, encapsulation, polymorphism.

Unit 4: Coordinate Transformations and Image Handling **(6 hrs)**

Affine and perspective transformations using matrices; Interpolation techniques; Raster data: reading bands, metadata, histograms; Libraries: NumPy, Rasterio, GDAL.

Unit 5: Spatial Data Processing and Geoprocessing **(6 hrs)**

Vector data models: points, lines, polygons; CRS, projections, spatial joins, and geometric operations; Geoprocessing tasks: buffer, clip, merge, dissolve; Libraries: GeoPandas, Shapely.

Unit 6: Spatial Analysis and Applications **(7 hrs)**

Raster processing, terrain analysis, slope/aspect/hillshade; Spatial interpolation techniques (IDW, Kriging - basic); Watershed modeling: DEM preprocessing, flow direction; Spatial statistics and network analysis (e.g., shortest path using OSMnx).

5. Tutorial

N/A

6. Practical

Lab #	Question Description
1	Basic Python scripting: variables, loops, functions
2	Vector data processing with GeoPandas (e.g., land-use shapefiles)
3	Raster analysis: DEM visualization and histogram analysis
4	Coordinate transformations and projection handling
5	Geoprocessing operations: buffer, clip, union, merge
6	Raster terrain analysis (slope, hillshade, aspect)
7	Watershed delineation using DEM and flow direction modeling
8	Spatial visualization using Matplotlib, Seaborn, Plotl

For each lab work, students are supposed to implement the algorithm with performance analysis.

7. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

8. Recommended Books

1. Bolstad, P. (2016). GIS Fundamentals: A First Text on Geographic Information Systems.
2. Campbell, J.B., & Wynne, R.H. (2011). Introduction to Remote Sensing.
3. DeMers, M.N. (2018). Fundamentals of Geographic Information Systems.
4. Jensen, J.R. (2015). Introductory Remote Sensing: Principles and Concepts.
5. Lillesand, T., Kiefer, R.W., & Chipman, J. (2015). Remote Sensing and Image Interpretation.
6. Mather, P.M. (2016). Computer Processing of Remotely-Sensed Images: An Introduction.
7. Richards, J.A., & Jia, X. (2013). Remote Sensing Digital Image Analysis: An Introduction.
8. Zhang, Y., & Xu, B. (2022). Big Data and Machine Learning in Geospatial Analysis.

Computer Vision

Course Code	Elective II	Year/Semester	IV/VII
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hr/wk
		Practical	2 hr/wk
		Total	96 hrs

1. Course Description

This course introduces fundamental and advanced concepts of Computer Vision, focusing on image processing, feature extraction, object detection, and deep learning-based approaches. The course bridges theoretical foundations with practical applications in Artificial Intelligence (AI), Robotics, Medical Imaging, and Augmented Reality (AR). Students will implement algorithms using Python, OpenCV, and modern deep learning frameworks.

2. Course Objectives

The course is designed with following objectives:

- Understand image formation, filtering, and transformations.
- Learn feature detection, matching, and segmentation techniques.
- Explore deep learning models (CNNs, R-CNN, YOLO, GANs) for vision tasks.
- Implement real-world applications (e.g., facial recognition, autonomous vehicles).
- Gain hands-on experience with OpenCV, TensorFlow/PyTorch.

3. Course Details

(48 hrs)

3.1 Theory

(5 hrs)

Unit 1: Introduction to Computer Vision

Overview of Computer Vision, applications in AI/IT; Image formation: Pinhole camera, lenses, sensors; Color spaces (RGB, HSV, LAB (Lightness-A-B)), image representations.

(6 hrs)

Unit 2: Image Processing Fundamental

Pixel operations, histograms, thresholding; Linear and nonlinear filtering (Gaussian, Sobel, Canny edge detection); Morphological operations (erosion, dilation).

(6 hrs)

Unit 3: Feature Extraction & Matching

Corner detection (Harris, Shi-Tomasi); Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB); Feature matching (Brute-Force, FLANN), RANSAC.

(8 hrs)

Unit 4: Deep Learning for Vision

Convolutional Neural Networks (CNNs), architectures (LeNet, ResNet); Transfer learning, data augmentation; Object detection (R-CNN, Fast R-CNN, YOLO, SSD); Generative models (GANs, VAEs).

(7 hrs)

Unit 5: 3D Vision & Advanced Topics

Stereo vision, depth estimation (Epipolar geometry); Structure from Motion (SfM), Simultaneous Localization and Mapping (SLAM); Optical Flow, Video Analysis.

(8 hrs)

Unit 6: Optimization & Deployment

Model Compression: Quantization (FP32→INT8), pruning, and knowledge distillation to reduce model size; Edge Deployment: TensorFlow Lite, ONNX Runtime, and hardware acceleration (NPUs/GPUs) for embedded devices; Real-Time Optimizations: Lightweight models (MobileNet), frame skipping, and multi-threading for speed; Performance Metrics: Latency, FPS, and power efficiency trade-offs in deployed systems; Tools & Debugging: Profilers (TF Profiler), logging, and visualization for edge AI optimization.

Unit 7: Applications & Emerging Trends **(8 hrs)**

Facial recognition, medical imaging, AR/VR; Ethical considerations, bias in vision systems; Case studies (self-driving cars, industrial automation).

3.2 Laboratory Works

- Image filtering & edge detection using OpenCV
- Feature extraction & matching (SIFT/SURF)
- Face detection using Haar cascades
- Image segmentation (K-means, GrabCut)
- Hands-on CNN with TensorFlow/Keras
- Object detection using YOLO
- Optical flow & motion tracking
- 3D reconstruction from stereo images
- Deploying a vision model on Raspberry Pi
- GAN-based image generation

4. Evaluation Scheme

Evaluation of students' performance is divided into two main parts: (1) Internal evaluation and (2) Semester examination, with breakdown as follows.

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	20	10
	Assignment	10		
	Seminar/Project/Presentation	5		
	Practical/Lab Examination	20	20	10
	Internal Examination	**20**	**20**	**10**
	Total Internal Marks		**60**	**30**
Semester-End Examination			**40**	**20**

5. Books

- Szeliski, R. (2022). Computer vision: algorithms and applications. Springer Nature.
- Forsyth, D. A., & Ponce, J. (2002). Computer vision: a modern approach. prentice hall professional technical reference.
- Davies, E. R. (2004). Machine vision: theory, algorithms, practicalities. Elsevier.
- Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.".
- Howse, J. (2013). OpenCV computer vision with python (Vol. 27). Birmingham, UK: Packt Publishing.

Practical book: Lakshmanan, V., Görner, M., & Gillard, R. (2021). Practical machine learning for computer vision. " O'Reilly Media, Inc.".

Ethics and Policy for Data Science

Course Code	Elective II	Year/Semester	IV/VII
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	1 hrs/wk
		Practical	N/A
		Total	64 hrs

1. Course Description:

This course explores the ethical, legal, and policy implications of data science and artificial intelligence, with a particular focus on responsible data practices, algorithmic fairness, privacy, and governance. Through local and global case studies—including from Nepal—it equips students to critically assess the societal impact of their work and make informed decisions as future data scientists and AI practitioners.

2. Course Objectives:

- Develop and Understand ethical frameworks and policy implications in data science.
- Analyze issues related to data privacy, algorithmic bias, accountability, and transparency.
- Apply ethical reasoning to real-world data science dilemmas.
- Evaluate national and international laws, including policies related to Nepal.
- Design ethical solutions to data science problems.
- To familiarize participants with Intellectual Property Rights (IPR) relevant to data science and their implications for ethical data use.
- To develop guidelines and frameworks for ethical and responsible use of data in data science practices.
- To explore ethical issues and challenges inherent in data science, including accountability, privacy, security, fairness, and bias.
- To analyze real-world case studies to understand ethical dilemmas and best practices in data science.

3. Learning Outcomes

After the completion of this course, students will be able to

- Analyze and apply ethical theories to real-world data science problems.
- Critically evaluate data privacy, ownership, and protection laws, with specific awareness of Nepal's legal context.
- Identify and mitigate bias and discrimination in AI models and data pipelines.
- Assess algorithmic transparency and accountability issues in complex data systems.
- Interpret and apply national and international policy frameworks on data governance.
- Propose ethically sound solutions to emerging issues like surveillance, cybersecurity, and autonomous systems.
- Demonstrate ethical reasoning and decision-making through case analysis and group projects.

4. Course Details **(48 hrs)**

Unit 1: Introduction to Ethics in data Science **(8 hrs)**

Moral and Morality, Ethos and Ethics, Definition and scope of data ethics; Importance of ethics in CS and AI, Moral philosophy basics: Utilitarianism, Deontology, Virtue ethics, Stakeholders in data systems, Ethical Analysis and Challenges in Data Science, Professional Ethics of Data Scientists; Case study: Ethical concerns in COVID-19 contact tracing apps (e.g., Hamro Swasthya).

Unit 2: Data Privacy and Protection **(8 hours)**

Concept of privacy and confidentiality, Goals and Principles of Data Governance; Data protection principles, Consent and anonymization techniques, Regulatory Landscape and Compliance Requirements; Data Protection and Privacy Provisions in Nepal; Data Protection Bill (Nepal), Ethical collection and usage of personal data.

Case Study: Misuse of voter data and biometric registration by local governments

Unit 3: Intellectual Property Rights (IPR) and Data Science **(8 hrs)**

Intellectual Property Rights (IPR), Copyrights, Patents, Trade Secrets, Trademark, Geographical Indicators, Industrial Designs, Licensing Models for Data, Intellectual Property Crimes, Data Ownership, Protection of Ownership Rights.

Unit 4: Framing Guidelines for Using Data Ethically and Responsibly **(6 hrs)**

The five C's

a. Consent
b. Clarity
c. Consistency and Trust
d. Control and Transparency
e. Consequences

Implementing the five C's

Unit 5: Emerging Issues: Surveillance, AI Ethics, and Cybersecurity (10 hours)

Facial recognition and mass surveillance, AI decision-making in critical sectors (health, finance, law enforcement), Ethical hacking and cybersecurity ethics, Dual-use technologies, Case analysis frameworks (e.g., Moor's Method, ACM Code of Ethics).

Unit 6: Case studies **(8 hrs)**

a) CCTV and surveillance systems in Kathmandu Valley: Ethics vs. security
b) AI usage in public safety and border security
c) Case studies with Ethical Consideration of real-world problems in the realm of Data Science
d) Review on
   i. Data policy within organizations
   ii. IT policy within organizations
   iii. Privacy policy within organizations

5. Tutorial (16 hrs)

At least one seminar presentation on Ethical issues in emerging technologies

6. Practical

N/A

7. Evaluation Scheme

The evaluation of student's performance in the course will be based on the following components:

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal evaluation	Class attendance and performance	5	40	20
	Assignment	15		
	Seminar/project/presentation	20		
	Practical/lab examination	-	-	-
	Internal examination	20	20	10
	Total of internal evaluation		**60**	**30**
Semester examination	**Total of semester examination**		**40**	**20**

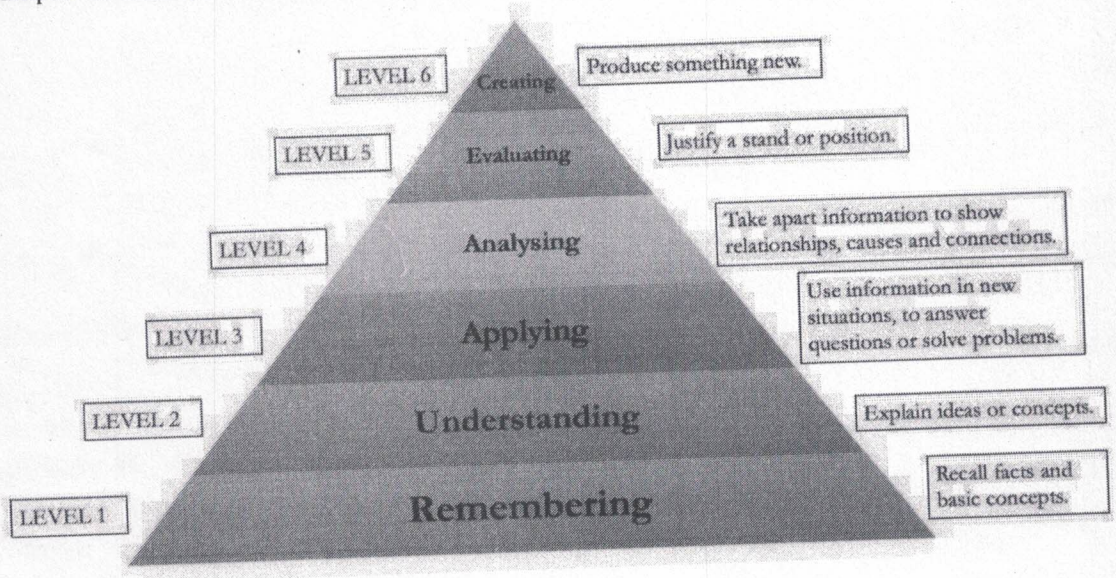
8. Recommended Books

1. Floridi, L. (2013). The Ethics of Information
2. O'Neil, C. (2016). Weapons of Math Destruction
3. Mittelstadt, B. D., & Floridi, L. (2016). *The Ethics of Big Data
4. Loukides, Mike, Hilary Mason, and D. J. Patil. *Ethics and data science*. O'Reilly Media, 2018.
5. Kelleher, John D., and Brendan Tierney. *Data science*. MIT Press, 2018.
6. Grus, Joel. *Data science from scratch: first principles with Python*. O'Reilly Media, 2019.
7. D'ignazio, Catherine, and Lauren F. Klein. *Data feminism*. MIT Press, 2023.
8. Eubanks, Virginia. *Automating inequality: How high-tech tools profile, police, and punish the poor*. St. Martin's Press, 2018.
9. Noble, Safiya Umoja. "Algorithms of oppression: How search engines reinforce racism." *Algorithms of oppression*. New York University Press, 2018.
10. Pasquale, Frank. *The black box society: The secret algorithms that control money and information*. Harvard University Press, 2015.
11. ACM Code of Ethics and IEEE Ethically Aligned Design

Annex A

Bloom's Taxonomy action verbs:

As Bloom's taxonomy is a hierarchy of progressive processes ranging from the simple to the complex, in which it is necessary to first master those lower down the pyramid before being able to master those higher up, the framework promotes what Bloom termed 'mastery learning'. In other words, by moving up the taxonomy, students become more knowledgeable, more skilled and develop an improved understanding of the content they are learning. Thus, by creating lesson plans and tasks, using the examples of verbs (in italics) provided, teachers can align with the different levels of the taxonomy.



Level 1, Remembering, is the most basic, requiring the least amount of cognitive rigour. This is about students recalling key information, for example, the meaning of a word.

Arrange	*Define*	*Describe*	*List*	*Match*	*Name*	*Order*	*Recall*	*Reproduce*

Level 2, Understanding, is to do with students demonstrating an understanding of the facts remembered. At this level, the student who recalls the definition of a word, for example, would also be able to show understanding of the word by using it in the context of different sentences.

Classify	*Discuss*	*Explain*	*Identify*	*Report*	*Summarise*

Level 3, Applying, is concerned with how students can take their knowledge and understanding, applying it to different situations. This usually involves students answering questions or solving problems.

Apply	*Calculate*	*Demonstrate*	*Interpret*	*Show*	*Solve*	*Suggest*

Level 4, Analysing, is about students being able to draw connections between ideas, thinking critically, to break down information into the sum of its parts.

Analyse	*Appraise*	*Compare*	*Contrast*	*Distinguish*	*Explore*	*Infer*	*Investigate*

Level 5, Evaluating, is reached when students can make accurate assessments or judgements about different concepts. Students can make inferences, find effective solutions to problems and justify conclusions, while drawing on their knowledge and understanding.

Argue	*Assess*	*Critique*	*Defend*	*Evaluate*	*Judge*	*Justify*

Level 6, Creating, is the ultimate aim of students' learning journey. At this final level of Bloom's taxonomy, students demonstrate what they have learnt by creating something new, either tangible or conceptual. This might include, for example, writing a report, creating a computer program, or revising a process to improve its results.

Compose	*Construct*	*Create*	*Devise*	*Generate*	*Organise*	*Plan*	*Produce*