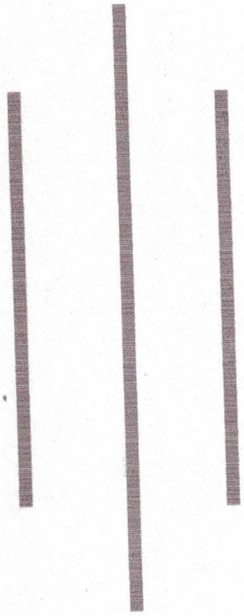
Lumbini Technological University

Institute of Engineering and Information Technology

Office of the Dean

Nepalgunj, Banke

Course Curriculum
Master of Technology in Information Technology
(M. Tech. in IT)



September, 2025

Contents

1. **Name of the program**

   Master of Technology in IT (M. Tech. in IT)

2. **Course description**

   The M. Tech. in IT program is a two-year, four-semester academic and professional program focused on computer applications and IT. This course outline offers a framework for a Master of Technology in information technology (M. Tech. IT). It includes important topics such assessment techniques, specialist areas, curriculum structure, and program objectives. This is merely a broad concept that can be modified to meet certain business, industry and university requirements.

   The M. Tech. in IT degree is primarily focused on application development, it places more of an emphasis on the newest programming languages and tools to create applications more quickly and effectively.

   Students receive domain expertise in a variety of subjects, including database administration, data mining and warehousing, computer networks, enterprise resource planning, software development, algorithms, data engineering, and system administration. Projects will be incorporated into the courses so that students can have hands-on experience on the application of IT industries.

3. **Goal of the program**

   a. To offer in-depth theoretical and practical understanding of fundamental IT topics.
   b. To gain sophisticated technical abilities in fields like as cloud computing, data science, networking, software development, and cyber security.
   c. To encourage the development of innovative and research skills to help enhance IT.
   d. To develop your communication, cooperation, and ethical standards.
   e. To get graduates ready for leadership positions in academia, research institutions, and the IT sector.

4. **Career prospects**

   This program graduates are in great demand across a range of industries. They can work as systems engineers, software architects, data scientists, research scientists, AI/ML engineers, cybersecurity specialists, and more. A solid basis for pursuing a Ph.D. is also provided by the advanced knowledge and research expertise. A graduate of this program can be the lead in an industry as:

   - IT Manager
   - Software Architect
   - Data Scientist
   - Cybersecurity Analyst
   - Cloud Architect
   - Network Engineer

- Consultant
- Teaching Faculties
- Research Assistant/Associate
- Entrepreneur

5. Pre-requisites

The students should meet following criteria:

- Basic knowledge of programming, data structures, algorithms, database management systems
- Basic knowledge of probability, statistics, discrete mathematics and linear algebra

6. Credit and duration of the program

A M. Tech. in IT program is worth 63 credits. The M. Tech. in Information Technology program lasts for two years and four semesters. The 16-week semester requires four working hours per week for instruction and learning, omitting tests and other activities.

7. Eligibility

Applicants for the M.Tech. in Information Technology must hold a bachelor's degree in Information Technology or an equivalent qualification from a recognized university. Candidates with undergraduate degrees in IT, Computer Science, Information and Communication Technology (ICT), Information Systems, Engineering, or related fields are also eligible. Additionally, students with a bachelor's degree in any discipline who have successfully completed a postgraduate program in Computer Applications or Information Technology may apply.

8. Admission process

The Dean's Office (Institute of Engineering and IT) will administer an entrance exam once a year, in the fall and spring sessions. Successful applicants who meet the requirements will be admitted based on merit. The programs academic year is divided into two semesters. Admissions for the autumn semester commence in September, while those for the spring semester begin in February. The detailed academic schedule will be announced by the Dean's Office.

9. General evaluation scheme

The program evaluation scheme is categorized into following parts:

i. Evaluation practical subjects

Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		

	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		**60**	**30**
Semester End Examination			**40**	**20**

ii. Evaluation non-practical subjects

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 Internal Marks		**60**	**30**
Semester End Examination			**40**	**20**

10. Final examination format

A student is considered to have passed a subject only if they secure a minimum of 50% marks in that subject, with mandatory separate passing in both internal and external assessments of theory and practical examinations.

11. Grading system

A students performance will be evaluated based on marks obtained in internal assessments, practicals, and semester-end examinations. These marks will be graded on an absolute basis, and the final outcome will be expressed on a four-point grading scale ranging from 0 to 4.

Grading scale:

Grade	**A+**	**A**	**A-**	**B+**	**B**	**B-**	**C+**	**C**	**F**
Marks Range	90-100	80-89	75-79	70-74	65-69	60-64	55-59	50-54	<50
Grade Point	4	3.7	3.3	3	2.7	2.3	2	1.7	0

Performance of a student in a semester is evaluated in terms of the Semester Grade Point Average

$$SGPA = \frac{\text{Total honor points earned in a Semester}}{\text{Total number of credits registered in a semester.}}$$

Similarly, performance of a student in a whole program is evaluated in terms of the Cumulative Grade Point Average

$$CGPA = \frac{\text{Total honor points earned}}{\text{Total number of credits completed.}}$$

Moreover, the students shall receive their semester grades and academic transcript grades only in letter grades and GPA scores. A student secure Semester Grade Point Average (SGPA) less than 2.7 or less than B Grade (B-, C+, C) in each subject of every semester may apply for grade improvement examination based on university guideline. Furthermore, a student who secures CGPA less than 3 may request for the opportunity to improve the grade in maximum two subjects. The concerned dean office will provide only one chance to upgrade their CGPA. The chance exam of the chosen subjects to improve grade shall be held as per the regular examination process

12. Attendance requirement

Student must attend at least 60% of the classes actually held in each course. It is recommended that the students must attend every lecture, tutorial, and practical classes. Those who will not able to fulfil attendance requirement are not eligible to attend in final examination of that course.

13. Maximum duration to complete course

The allocated time to complete the course is maximum 5 years. If any student will unable to complete the program during this maximum period of time but still he/she wants to complete the program, he/she will have to re-join from beginning following the complete admission procedure.

14. Reattempting the subject

A subject may generally be taken only once for a grade. However, if a student earns less than a B (i.e., B-, C+, or C) or receives an F (Fail), the course may be retaken. Retaking a course with a grade of B-, C+, or C is optional and provides an opportunity to improve performance, whereas a course with an F grade must be retaken and passed to fulfil the program requirements. In the case of a retake, the grade obtained will replace the earlier grade, and evaluation will follow a best-of-two system meaning the higher grade achieved, whether in the original attempt or the retake, will be recorded as the final grade.

15. Dissertation

As per the need and interest of students, a student must submit the research report of dissertation to the office of the dean of their respective institution.

16. Award of degree

Lumbini Technological University (LTU), awards of M. Tech in IT with specialization degree after completion of all the requirement of this program as prescribed in this curriculum.

17. Course structure

The four semesters course structure is provided in the table below:

SN	Course Code	Semester	Course Name	Number of Credit
1	MIT 601	I	Design and Analysis of Algorithm	3
2	MIT 602	I	Research Methodology	3
3	MIT 603	I	Application Developing using Python and Framework	3
4	MIT 604	I	Advance Operating System	3
5	MIT 605	I	Computational Mathematics	3
6	MIT 606	I	Seminar in Emerging Trends	1
7	MIT 651	II	Optimization Technique	3
8	MIT 652	II	Statistics and Probability	3
9	MIT 653	II	Software Project Management	3
10	MIT 654	II	Data Warehouse and Data Mining	3
11	MIT 655	II	Project Development	2
12	Elective I	II	Chosen Elective I Subject	
13	MIT 701	III	Machine Learning	3
14	MIT 702	III	Mobile Application Development	3
15	Elective II	III	Chosen Elective II Subjects	
16	Elective III	III	Chosen Elective III Subjects	
17	Elective IV	III	Chosen Elective IV Subjects	
18	MIT 751	IV	Thesis	15

List of Elective Subjects:

Electives	Course Code	Course Name	Numbers of Credit
Elective I	MIT 656	Knowledge Management	3
	MIT 657	e-Governance	3
	MIT 658	Cloud Computing	3
Elective II	MIT703	Web Programming	3
	MIT704	DevOps for Web Development	3
Elective III	MIT705	Big Data Analytics	3
	MIT706	Human Computer Interactions	3
	MIT707	Information Security Audit	3
	MIT708	Geographical Information System	3
Elective IV	MIT709	Linux Administration	3
	MIT710	Virtualization technique and application	3
	MIT711	Remote Sensing	3

Cre

1. Course Description

This course provides an
Students will study a range o
algorithms to solve complex
algorithm theory, including ap
structures.

2. Course Objectives

By the end of this course, students

- Analyze the efficiency of algo notations.
- Design and implement algorit ional problems.
- Evaluate and apply advanced a optimize algorithm performance.
- Understand and apply concepts completeness and computational intractability.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Justify the correctness of algorithms through inductive proofs.
- Perform various analysis of algorithms using asymptotic notations.
- Understand and apply divide-and-conquer algorithm, and develop algorithm using this approach.
- Apply the dynamic programming approach, identify suitable scenarios for its use, and create and evaluate dynamic programming algorithm.
- Comprehend greedy algorithm approach, determine when to use it, develop algorithms based on this strategy, and conduct their analysis.
- Understand major graph algorithms, and apply graphs to solve engineering challenges.
- Describe methods for analyzing randomized algorithms, including their expected running times and error probabilities.
- Define approximation algorithms and their advantages, be familiar with various such algorithms.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Introduction (6 hrs)

- Introduction to Algorithm
- Importance of Algorithm
- How to measure performance of an algorithm?
- Algorithm analysis including Best-, worst-, and average-case performance.
- Asymptotic notation: big-O, big-Ω, and big-Θ; little-o, and little-ω.

Unit 2: Divide and Conquer (8 hrs)

- Structure of Divide and Conquer algorithms
- Binary search. Fast integer multiplication
- Sorting algorithms – Merge sort and Quick Sort
- Master Theorem

4.2 Laboratory Work (16 hrs)

Lab	Practical Title	Hours
1	Implement Linear and Binary Search. Compare their performance on different input sizes.	2 hrs
2	Implement Bubble, Insertion, and Merge Sort algorithms. Analyze their time complexity.	2 hrs
3	Implement Quick Sort (including randomized version). Compare both versions.	2 hrs
4	Implement Karatsuba Algorithm for fast integer multiplication. Compare with standard method.	2 hrs
5	Implement Matrix Chain Multiplication using Dynamic Programming.	2 hrs
6	Construct Huffman Tree for a given set of character frequencies. Generate Huffman Codes.	2 hrs
7	Implement Kruskal's and Prim's Algorithms to find Minimum Spanning Tree (MST).	2 hrs
8	Implement Dijkstra's Algorithm for shortest path in a graph with non-negative weights.	2 hrs

5 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

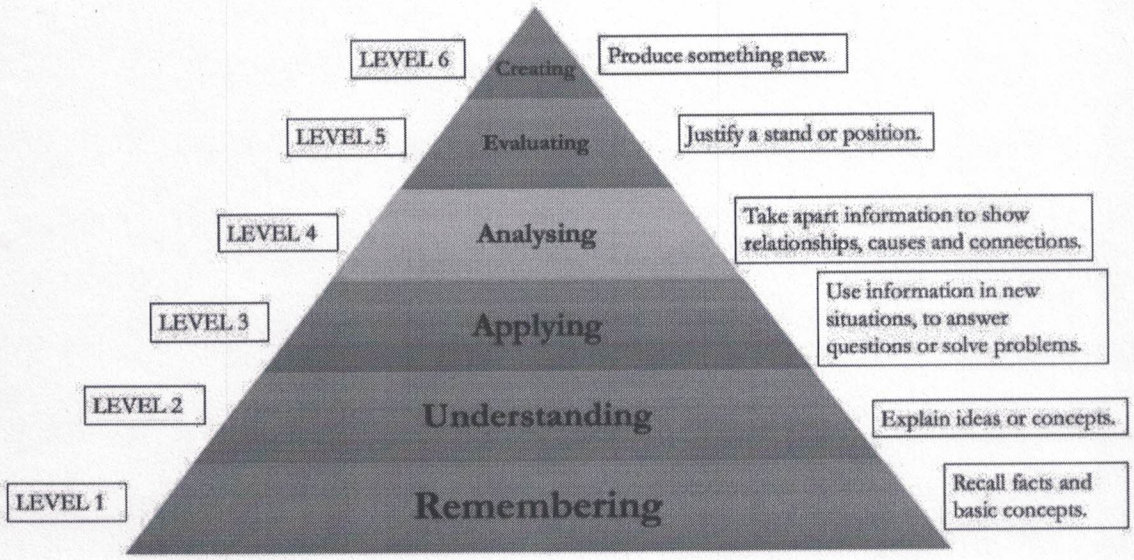
6 Books (4-10 books):

- "Introduction to Algorithms", 3rd Edition, by Cormen, Leiserson, Rivest, and Stein.
- "Fundamentals of Computer Algorithms", 2nd Edition by Sartaj Sahni and Sanguthevar Rajasekaran, and Ellis Horowitz.
- "Algorithms", 4th Edition, by Robert Sedgewick and Kevin Wayne.
- "The Algorithm Design Manual", 2nd Edition, by Steven S. Skiena.

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.

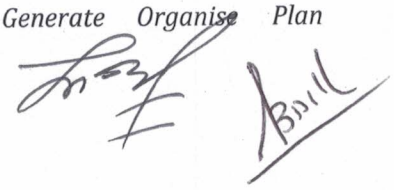
Analyse Appraise Compare Contrast Distinguish Explore Infer Investigate

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*



Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Research Methodology**

Course Code	MIT 602	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

Research Methodology in Computer Science/IT is a foundational course designed for first-year, first-semester Master's students pursuing a degree in Computer Science/ IT. This course provides students with a comprehensive understanding of research methods, techniques, and best practices relevant to data science research. Through a combination of theoretical lectures, practical exercises, and hands-on projects, students will learn how to formulate research questions, design research studies, collect and analyze data, and communicate research findings effectively.

2. Course Objectives

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

- To introduce students to the principles and concepts of research methodology in the context of data science.
- To develop students' skills in formulating research questions and hypotheses.
- To familiarize students with various research study designs and data collection techniques used in data science research.
- To equip students with the knowledge and tools necessary to analyze and interpret data using statistical methods and data visualization techniques.
- To enable students to effectively communicate research findings through written reports and presentations.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- **Knowledge (Remembering)**
  - Define key terms and concepts related to research methodology in data science. o Recall the principles of ethical conduct in research and data handling. o Identify the different types of research designs and sampling techniques.
- **Comprehension (Understanding)** o Summarize the importance of research in advancing knowledge and solving real-world problems in the context of data science.
  - Explain the process of formulating research questions and hypotheses.

- Interpret various types of research designs and their applicability to different research scenarios.

- **Application (Applying)** o Apply appropriate data collection methods and sampling techniques to design research studies in data science.
  - Utilize statistical techniques to analyze and interpret data collected from research studies.
  - Implement data visualization techniques to communicate research findings effectively.
- **Analysis (Analysing)** o Analyze the strengths and limitations of different research designs and sampling methods in data science research.
  - Evaluate the validity and reliability of the research findings based on statistical analysis.
  - Compare and contrast different data visualization techniques and their effectiveness in conveying insights from data.
- **Synthesis (Creating)** o Design research proposals and project plans for data science research studies, including research questions, objectives, and research methodologies.
  - Develop research reports and academic papers that adhere to standard formatting and citation guidelines.
  - Synthesize research findings into coherent and visually appealing presentations for diverse audiences.

- **Evaluation (Evaluating)**
  - Critically assess the ethical implications of data collection, analysis, and reporting in data science research.
  - Evaluate the appropriateness of research designs and methodologies based on research objectives and constraints.
  - Provide constructive feedback on peer-reviewed research proposals, reports, and presentations, focusing on clarity, rigor and validity.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Introduction to Research Methodology (4 hrs)

- Overview of Research Methodology in Data Science.
- Importance of Research in advancing knowledge and solving real-world problems.
- Ethical considerations in data science research.
- Formulating research questions and hypotheses.

Unit 2: Research Design and Sampling (8 hrs)

- Types of Research Design (e.g., experimental, observational, survey).
- Sampling techniques and strategies for data collection.
- Sample size determination and power analysis. - Experimental design and control variables.

Unit 3: Data Collection Methods **(10 hrs)**

- Primary data collection methods (e.g., surveys, interviews, observations).
- Secondary data sources (e.g. public datasets, data repositories).
- Data preprocessing and cleaning techniques.
- Data privacy and confidentiality considerations.

Unit 4: Data Analysis Techniques **(12 hrs)**

- Descriptive statistics and exploratory data analysis (EDA).
- Inferential statistics and hypothesis testing.
- Regression analysis and predictive modelling.
- Machine learning algorithms for data analysis (e.g., classification, clustering, regression).

Unit 5: Data Visualization and Interpretation **(6 hrs)**

- Principles of data visualization and graphical representation.
- Tools and libraries for data visualization (e.g., Matplotlib, Seaborn, ggplot2).
- Interpretation of visualization and communicating insights effectively.
- Interactive visualization techniques for exploratory analysis.

Unit 6: Research Communication and Reporting **(8 hrs)**

- Writing research proposals and project plans.
- Structuring research reports and academic papers.
- Presenting research findings orally and visually. - Peer review process and academic publishing.

4.2 Laboratory Work **(16 hrs)**

- Formulating research questions based on provided datasets.
- Collecting data from online sources or using survey tools.
- Preprocessing collected data to handle missing values and outliers.
- Performing EDA on collected datasets using Python/R libraries.
- Creating visualizations to identify patterns and insights in data.
- Implementing random sampling and stratified sampling techniques on provided datasets.
- Conducting hypothesis tests (e.g., t-tests, chi-square tests) on sample data.
- Interpreting test results and concluding.
- Building regression models to predict outcomes based on predictor variables.
- Assessing model performance and interpreting regression coefficients.
- Implementing machine learning algorithms (e.g., classification, clustering) on research datasets.
- Interpreting machine learning model outputs and evaluating model performances.

5 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6 Books (4-10 books):

i) Textbooks:

- "Research Methods for Data Science" by John W. Creswell and J. David Creswell.
- "Practical Statistics for Data Scientists" by Andrew Bruce and Peter Bruce.

ii) Online resources:

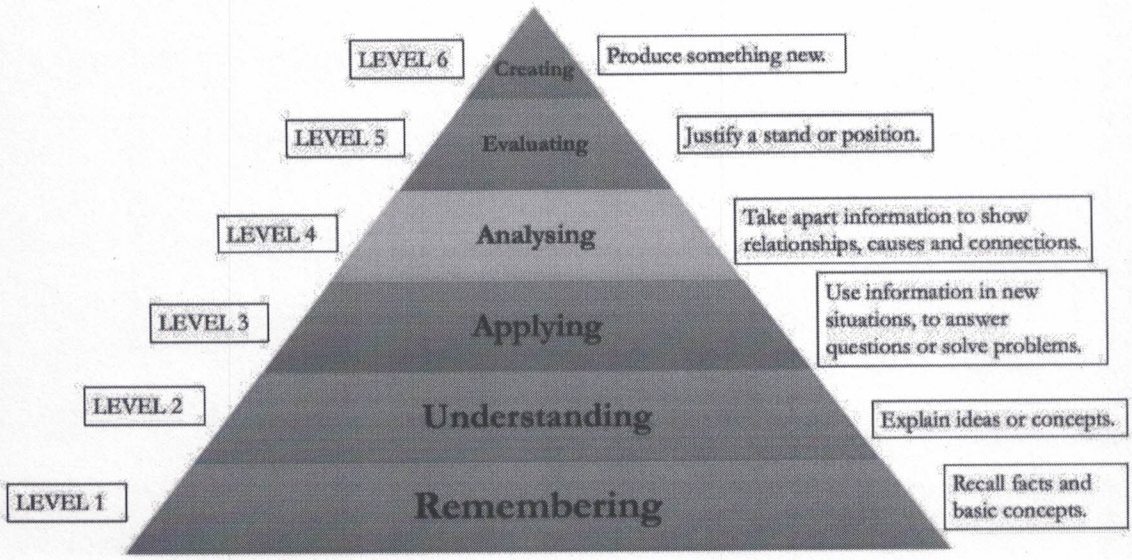
Research articles, tutorials, and documentation on research methodology and data science techniques.

iii) Software tools:

Python/R programming environments, Jupyter Notebooks, statistical analysis software (e.g., SPSS, SAS).

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

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject:Python programming**

Course Code	MIT 603	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course builds on a fundamental concept on programming in Python and leads to advanced programming techniques, covering object-oriented and high-level program design. It explores basic programming structures such as conditionals, iterations, recursions, functions, modules, files and object-oriented techniques such as classes and objects, and inheritance. The syllabus is divided into 7 units, where each chapter includes hands-on practical and tutorial sessions.

2. Course Objectives

The general objectives of this course is:

- Develop advanced-level programming skills in Python.
- Enable students to write efficient, readable, and maintainable Python code.
- Familiarize students with Python's extensive standard and third-party libraries.
- Equip students to perform data analysis, visualization, and modeling using Python tools.
- Encourage application of programming skills to solve real-world IT problems.
- Strengthen problem-solving through the use of structured, modular, and object-oriented programming.
- Prepare students for professional tasks in software development, data science, and automation.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Write a clear, effective and efficient computer program in Python
- Develop, debug and test Python program
- Use Python standard libraries and third-party modules
- Perform data analysis and visualisation
- Develop a complete application on python

4. Course Details

4.1 Theory **(48 hrs)**

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

- Computer program, programming language, software, compilers, interpreters
- Formal and natural languages, program debugging, syntax errors, runtime errors, experimental debugging, documentation, and comments.
- Program development life cycle, flowcharts, algorithms, and pseudocodes
- Python programming, variables, expressions and statements, operators, and keywords
- Variable and data types, variable naming, keywords, statements
- Evaluating expression, operators and operands, converter functions

Unit 2 Conditionals, Iterations and Functions **(8 hrs)**

- Boolean values and expressions, logical operators, conditional execution, nested conditionals,
- Iterations, assignments, updating variables, for loop, while statement, break statements, other flavours of loops, continue statements, more generalisations, nested loops for nested data.
- Basics of functions, function definitions, call, flow of executions, function arguments, return values, local variables and parameters

Unit 3: Strings, tuples, lists and dictionaries **(8 hrs)**

- Working with strings, lengths, slices, string comparison, immutable, find function, looping and counting, split method, string format method.
- Working with tuples and lists, tuple assignment, and tuple as return values
- Lists, accessing elements, membership, list operations, slices, deletion, aliasing, cloning, list methods, pure functions and modifiers, string and lists, list and range, nested lists
- Creating dictionaries, assessing values, methods, aliasing and copying, memoizations, sparse matrices
- Interpreting through dictionaries, comprehensions,

Unit 4: Modules and files **(6 hrs)**

- Concept of modules, random numbers, popular modules such as the time module, the math module, and creating your own modules.
- Namespaces, scope and lookup rules, attributes and dot operators.
- Import statements and its variants unit test and modules.
- Working with files, writing your first files, reading from file, working with binary files, directories, and fetching from the web.

Unit 5: Classes, objects and inheritance **(8 hrs)**

- Object-oriented programming, user-defined compound data types, attributes, initialisers, adding methods to a class, instance as arguments and parameters, return values
- Working with objects, rectangles, and mutable objects, sameness, and copying.
- Pure functions, modifiers, generalisations, operator overloading and polymorphism.

- Inheritance, type of inheritance, methods, self, __init_

Unit 6: Python libraries for computations **(6 hrs)**

- Working with existing libraries, os, sys, datetime, etc.
- Working with command line arguments
- NumPy arrays, slicing, operations, creating arrays with different operations, reshaping, flattening , transposing, indexing, Boolean indexing, broadcasting.
- Numpy for statistics, aggregate functions, axis operations, vectorised operations, linear algebra operations such as matmul, dot , determinant, rank, inverse, transpose, linlang etc.
- Pandas, reading csv, json, excel, data exploration, data selections, filtering, cleaning, transformation, grouping, aggregation, sorting and ranking, merging, joining, and time series data with pandas.

Unit 7: Data Visualisation **(6 hrs)**

- Why data visualisations, effective visualisation principles, Exploratory data analysis,
- Types of visualisation charts, graphs, and data importing.
- Matplotlib basics, axes, plots and customisations, lifecycle of plots, pairwise data, gridded data, statistical distribution, boxplots, 3D plots
- Seaborn basics, themes and aesthetics, categorical vs numerical plots, styling, pair plots, heatmaps, facet Grids, saving and exporting plots.

4.2 Laboratory Work **(16 hrs)**

Lab	Practical Title	Hours
1	Install Python, write a simple script, run and debug it.	2 hrs
2	Develop a program using conditionals, loops (for, while), and functions with arguments and return values.	2 hrs
3	Develop a program using strings, tuples, lists, and dictionaries demonstrating slicing, methods, and comprehensions.	2 hrs
4	Develop a program that imports modules, works with files (read/write), and fetches data from the web.	2 hrs
5	Write a program demonstrating classes, objects, methods, and inheritance (e.g., card game example).	2 hrs
6	Use Python libraries (os, sys, datetime), work with command-line arguments, NumPy arrays, and Pandas data manipulation.	2 hrs
7	Develop a data visualization project using Matplotlib and Seaborn for different types of plots and charts.	4 hrs

5 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6 Books (4-10 books):

i) Textbooks:

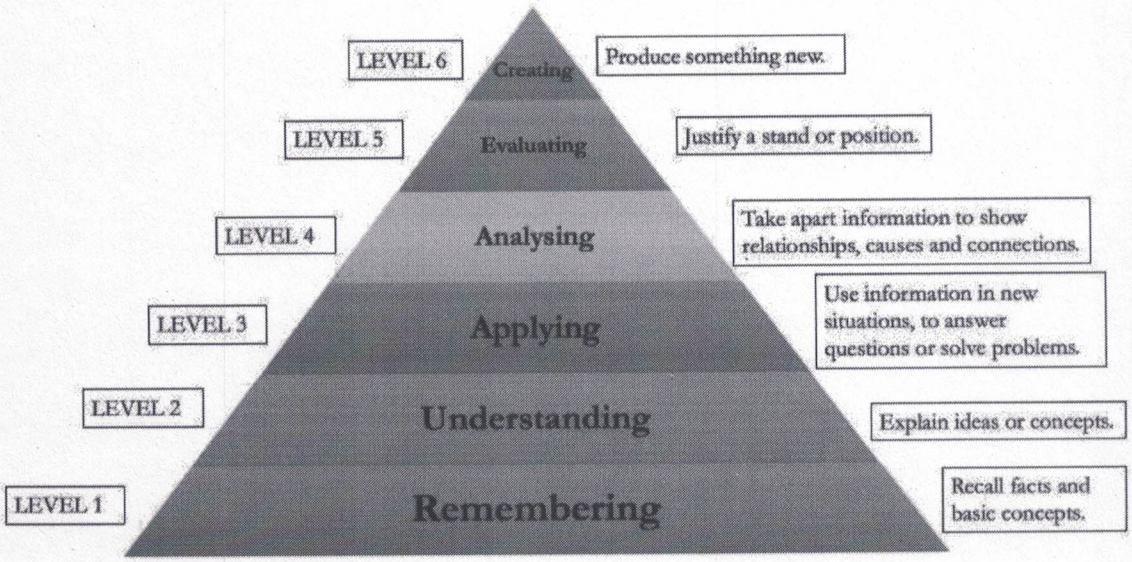
- How to Think Like a Computer Scientist: Learning with Python 3, Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers (3rd Edition)

ii) References:

- How to Think Like a Computer Scientist: Learning with Python 3rd editions
- Downey, A. (2012). Think python. " O'Reilly Media, Inc.".
- McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".
- Vasiliev, Y. (2022). Python for Data Science: A Hands-on Introduction. No Starch Press.

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

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Advanced Operating System**

Course Code	MIT 604	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course explores the advanced concepts, principles, and methodologies used in the design and operation of modern operating systems. Building on foundational knowledge of operating systems, the course emphasizes the mechanisms and policies that enable multitasking, resource management, concurrency, and system security in distributed, real-time, and virtualized environments

2. Course Objectives

The general objectives of this course is:

- To provide students with a deep understanding of the principles and technologies that underlie modern operating systems.
- To enable students to analyse and design advanced operating systems and their components.
- To prepare students to work with and develop emerging technologies in the field of operating systems, such as distributed systems, cloud computing, and mobile devices.
- To expose students to real-world case studies and examples of advanced operating systems in action.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Analyze advanced operating system structures and architectures, including kernel-level and user-level design choices in modern OS implementations.
- Explain and evaluate process synchronization, deadlock handling, and concurrency control mechanisms used in multiprocessor and distributed environments.
- Apply resource management policies for efficient CPU scheduling, memory allocation, and I/O handling in real-time and virtualized systems.
- Understand and analyze distributed system concepts, including communication, coordination, clock synchronization, and fault tolerance.
- Demonstrate the implementation and usage of virtual memory, paging, and caching mechanisms in contemporary operating systems.
- Critically examine system-level security measures such as access control, authentication, intrusion detection, and isolation in shared environments.

- Implement and simulate advanced OS features (e.g., threads, file systems, inter-process communication) using modern programming tools and techniques.
- Assess the role of virtualization and containerization technologies in system design and cloud-based deployment scenarios.
- Design and analyze algorithms for scheduling, memory management, and synchronization using real-world case studies and open-source operating systems.
- Engage in research and collaborative projects to explore cutting-edge developments in distributed, real-time, embedded, and cloud operating systems.

4. Course Details

4.1 Theory (48 hrs)

Unit-1 Introduction (3 hrs)

- Operating System Concepts
- System Calls
- Operating System Structure
- Introduction to key concepts and terminologies used in advanced operating systems, such as concurrency, virtualization, and distributed systems.

Unit-2 Processes, Threads, Deadlocks (10 hrs)

- Processes (Process model, Process Creation, State diagram, Implementation).
- Threads (Thread Model, POSIX Threads, Multi-Threading models, Thread Implementation).
- Inter-process Communication (Introduction to Critical Section Problem, Mutual Exclusion by Busy Waiting, Sleep and Wakeup, Semaphores, Mutexes, Monitors).
- Process Scheduling (Process scheduling and its categories).
- Deadlocks (Deadlock Characterization, Handling Deadlocks: Prevention, Avoidance, Detection, Recovery)

Unit-3 Memory Management (8 hrs)

- A memory abstraction
- Virtual memory
- Page replacement algorithms
- Design and implementation of issues of paging system
- Segmentation
- Advanced topics in memory management, such as persistent memory and support for high-performance computing and big data applications

Unit-4 Storage and I/O Systems (5 hrs)

- Files and Directories
- Implementing file systems

- Disk Scheduling
- RAID structure
- I/O Systems

Unit-5 Protection and Security **(3 hrs)**

- System Protection Mechanism
- Operating system security challenges
- System Security Mechanism

Unit-6 Virtualization and the cloud **(10 hrs)**

- Requirement of virtualization
- Hypervisors
- Techniques of virtualization
- Memory virtualization
- I/O virtualization
- Virtual Machines on Multicore CPUs
- Network, Distributed, Cloud file system

Unit-7 Multiple Processor Systems **(9 hrs)**

- Multiprocessors (Hardware, Operating System Types, Synchronization, Scheduling)
- Multicomputer (Hardware, Software, Remote Procedure Call, Distributed Shared Memory, Multicomputer Scheduling, Load Balancing)
- Distributed Systems (Introduction, Distributed Synchronization, Security)

4.2 Laboratory Work **(16 hrs)**

Lab	Practical Title	Hours
1	Demonstrate use of system calls like fork(), exec(), getpid(), wait() in Linux.	2 hrs
2	Implement process creation and thread management using POSIX threads and compare threading models.	2 hrs
3	Implement Inter-Process Communication (IPC) using pipes, shared memory, and semaphores.	2 hrs
4	Simulate Deadlock Detection and Avoidance using Banker's algorithm or resource allocation graph.	2 hrs
5	Simulate page replacement algorithms like FIFO, LRU, and Optimal using Python/C.	2 hrs
6	Simulate Disk Scheduling algorithms such as FCFS, SSTF, and SCAN.	2 hrs
7	Demonstrate Virtualization using tools like VirtualBox or KVM and explain memory/I/O virtualization.	2 hrs
8	Implement a basic Distributed Clock Synchronization algorithm (e.g., Lamport's algorithm).	2 hrs

5 Evaluation Scheme

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

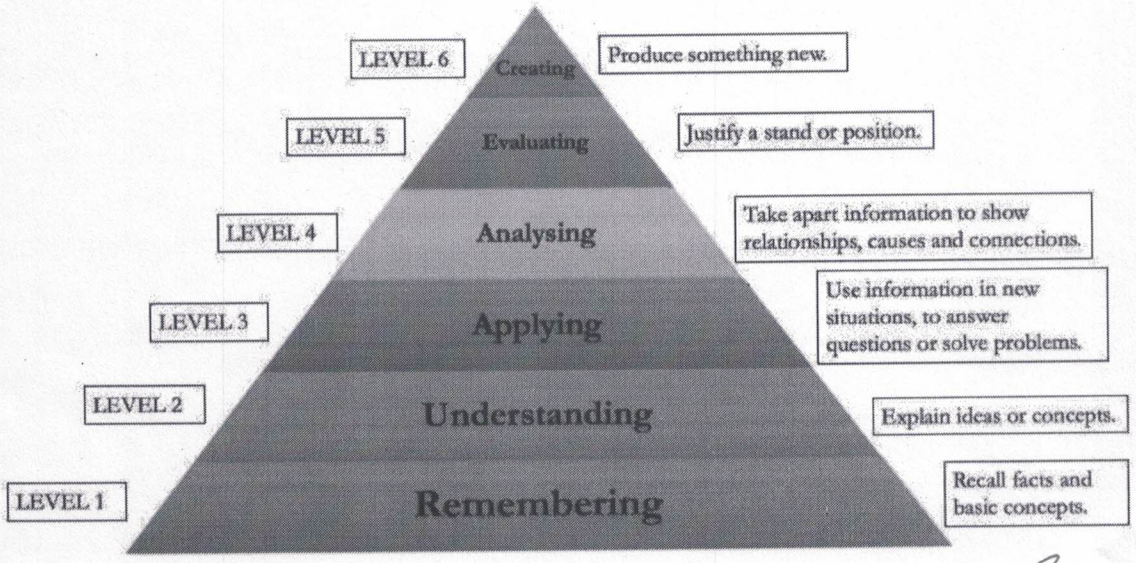
Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6 Books (4-10 books):

- Tanenbaum, A. S., & Bos, H. Modern Operating Systems, 5/e. Pearson.
- Silberschatz, A., Gagne, G., & Galvin, P. B. Operating System Concepts, 10/e. Wiley.
- Stallings, W. Operating Systems: Internals and Design Principles, 9/e. Pearson

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.

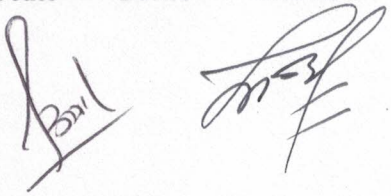
Analyse Appraise Compare Contrast Distinguish Explore Infer Investigate

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



Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Computational Mathematics**

Course Code	MIT 605	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course provides a foundation in mathematics essential for data science applications. It focuses on key concepts in scientific computing, linear algebra, and differential equations, enabling students to apply mathematical methodologies to solve complex data science problems. Students will gain hands-on experience through practical exercises and projects, enabling them to apply mathematical techniques to real-world data science problems.

2. Course Objectives

The general objectives of this course is:

- Develop a strong foundation in mathematical tools and techniques used extensively in data science.
- Enhance problem-solving skills by applying mathematical concepts to practical data science scenarios.
- Foster critical thinking and analytical abilities to effectively interpret and utilize mathematical results in data analysis.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- **Scientific Computing:**
  - Explain different types of errors encountered in scientific computing
  - Implement numerical algorithms to solve problems encountered in data science
  - Utilize interpolation and curve fitting techniques to solve practical problems in data science
- **Linear Algebra:**
  - Apply matrix operations and vector manipulations to represent and analyze data.
  - Solve systems of linear equations arising from data science applications.
  - Utilize concepts of eigenvalues and eigenvectors for dimensionality reduction and data analysis.
- **Differential Equations (ODEs):**
  - Model real-world phenomena relevant to data science using ordinary differential equations.

- Employ numerical techniques to solve ODEs and interpret the solutions in the context of data analysis.
- Understand the limitations and applications of ODEs in data science modeling.

4. Course Details

4.1 Theory (48 hrs)

Unit 1 Scientific computing in General **(8 hrs)**

- Errors : Absolute, Relative, round off, truncation
- Significant digits and Error propagation
- Solution to single non linear equation : Newton's method, Iteration method
- Solution to system non linear equations : Newton's Method, Iteration method

Unit 2 Interpolation and curve fitting **(8 hrs)**

- Curve Fitting : Linear, exponential, nth degree
- Interpolation : Lagrange and Newton
- Spline interpolation : Linear, Cubic
- Bspline interpolation : Linear, Cubic

Unit 3 Matrices and Linear system of equations **(6 hrs)**

- Vectors : Scalar Product, Norm, Angle, Distance
- Matrices : Symmetric, Skew-Symmetric, Hermitian, Orthogonal, Unitary Matrices ? The Gram-Schmidt process
- Solution to systems :
  - Direct methods :Gaussian elimination, LU decomposition, tridiagonal systems
  - Iterative methods : Gauss-Seidel, SOR, conjugate gradient

Unit 4 Eigenvalue problems **(8 hrs)**

- Diagonalization
- Power and inverse power method
- Householder's method
- The QR method
- Singular value decomposition (SVD)
- Principal component analysis (PCA)

Unit 5 Introduction to differential equations **(6 hrs)**

- Motivation with applications in data science
- Definition and terminology
- Classification of differential equations
- Geometrical meaning of first order differential equation
- Initial and boundary value problems

Unit 6 Solution of differential equations **(12 hrs)**

- Initial value problems (IVPs) : Eulers method, Runge-Kutta methods
- System and higher order ODEs : Eulers method, Runge-Kutta methods
- Boundary value problems (BVPs) : Finite difference method
- Partial differential equations (PDEs) : Laplace and Poisson equations

4.2 Laboratory Work (16 hrs)

Lab	Practical Title	Hours
1	Write a program to compute absolute, relative, round-off, and truncation errors.	2 hrs
2	Implement Newton-Raphson and Fixed-Point Iteration methods to solve a nonlinear equation.	2 hrs
3	Solve a system of nonlinear equations using Newton's method and iterative method.	2 hrs
4	Fit data using linear, exponential, and polynomial (nth degree) regression models.	2 hrs
5	Implement Lagrange and Newton interpolation for a given set of data points.	2 hrs
6	Perform cubic spline and B-spline interpolation on sample data.	2 hrs
7	Solve a system of linear equations using Gaussian elimination and LU decomposition.	2 hrs
8	Solve initial value problems using Euler's and Runge-Kutta methods.	2 hrs

5 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

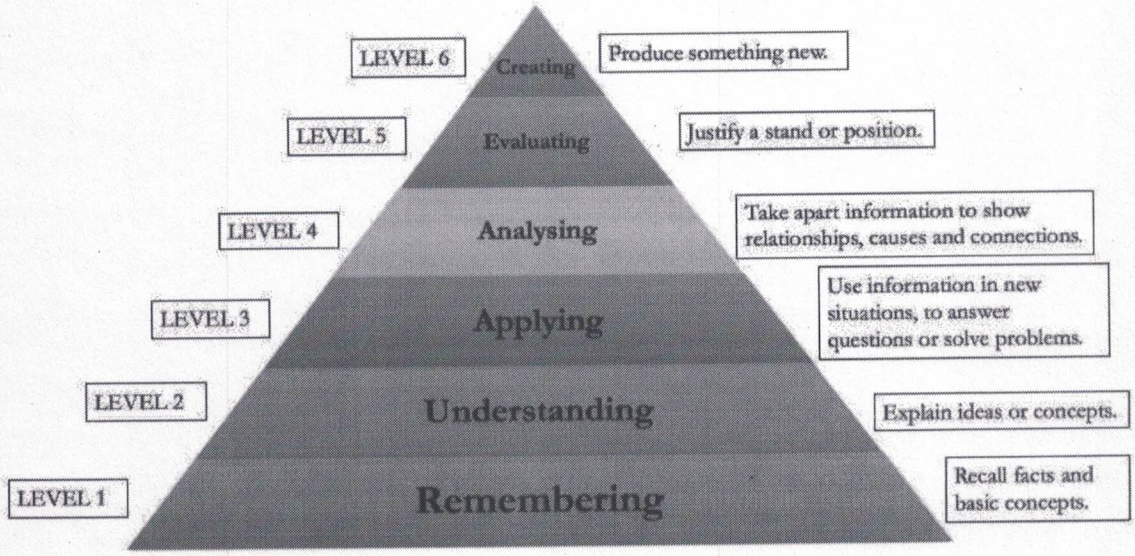
6 Books (4-10 books):

- Modelling with Differential Equations by D. U. Burghers, and M.S Borrie, Ellis Horwood Limited.
- Differential Equations and Their Applications by Zafar Ahsan, PHI Learning Pvt. Ltd.
- Numerical Methods and applications by E. Ward Cheney, and David R. Kinacaid, Cengage Learning
- Applied Numerical Analysis, C.F. Gerald and P.O. Wheately, Addison Wesely
- Linear Algebra and applications by Gilbert Strang, Cengage Learning

- Introduction to Computational linear algebra by Nabil Nassif, Jocelyne Erhel and Bernard Philippe, CRC press

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*

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Optimization Theory**

Course Code	MIT 651	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course is subjected to provide the fundamental concept of Optimization Theory necessary for data science. Students will learn the concept of Linear Programming, Integer Programming, Network Optimization and Non-Linear Programming together with the idea of Calculus in Optimization techniques.

2. Course Objectives

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

- Solve optimization problems relevant to data science using appropriate knowledge of optimization theory.
- Apply computational tools and optimization algorithms to practical problem-solving.
- Analyze, design, and implement optimization techniques for real-world applications.
- Evaluate the efficiency and effectiveness of different optimization methods in computational contexts.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Solve optimization problems relevant to data science by applying appropriate optimization theory.
- Utilize computational tools effectively and implement optimization algorithms for practical problem-solving.
- Analyze optimization techniques, design suitable approaches, and implement them for real-world applications.
- Apply dynamic optimization methods, identify suitable problem scenarios, and evaluate their effectiveness.
- Understand greedy and heuristic optimization strategies, determine appropriate use cases, and analyze their performance.
- Apply graph-based optimization algorithms to address engineering and computational challenges.
- Evaluate randomized optimization algorithms by analyzing expected performance and error probabilities.
- Define approximation methods for optimization, explain their advantages, and apply them to solve complex problems.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Introduction (4 hrs)

- Introduction to Optimization
- Types of Optimization: Linear, Non-Linear, Convex, Non-Convex
- Application of Optimization in Data Science

Unit 2: Optimization Using Calculus (6 hrs)

- Introduction to Local Extrema and Global Extrema
- Optimization Techniques for Univariate Functions
- Optimization Techniques for Multi-Variate Functions

Unit 3: Linear Programming (10 hrs)

- Concept of Linear Programming
- Simplex Method
- Duality Theorem

Unit 4: Integer Programming (6 hrs)

- Introduction to Integer Programming
- Types of Integer Programming Problem
- Gomory's All Integer Cutting Plane Method
- Branch and Bound Method

Unit 5: Network Optimization (10 hrs)

- Introduction
- Transportation Problem
- Assignment Problem
- Travelling Sales-Man Problem

Unit 6: Non-Linear Optimization (12 hrs)

- Introduction to Non-Linear Optimization
- Unconstrained Optimization (One variable and Multi-Variable)
- Convex Set and Functions
- Convex Optimization
- Non-Convex Optimization

4.2 Laboratory Work (16 hrs)

- Use of TORA Software in Optimization
- Use of Python Libraries in Optimization

5 Evaluation Scheme

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

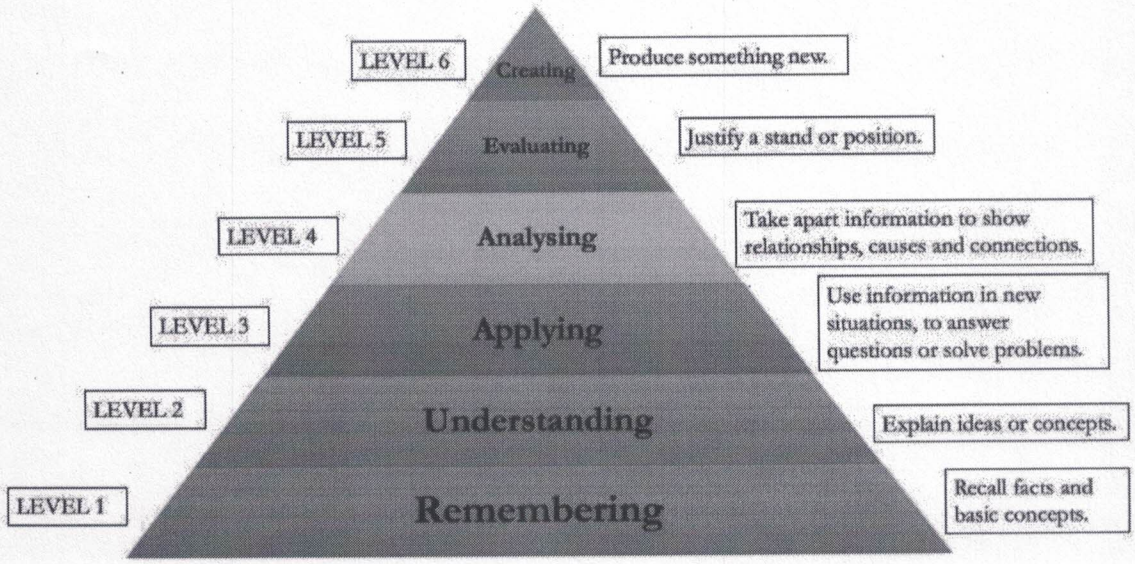
Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6 Books (4-10 books):

- Frederick S. Hillier and Gerald J. Lieberman, Introduction to Operations Research, McGraw-Hill.
- J. K. Sharma, Operations Research (Theory and Applications), Macmillan India, New Delhi.
- Michel Bierlaire, Optimization: Principles and Algorithms, EPFL Press.
- Donald A. Pierre, Optimization Theory with Applications, Dover Publications. Inc. New York
- Jorge Nocedal, Stephen Wright, Numerical Optimization, Springer.
- G. Hadley, Linear Programming, Addison Wesley Publishing Company.
- G. B. Thomas, J. Hass, C. Heil, M. D. Weir, Thomas' Calculus, Pearson.

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

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Statistics and Probability**

Course Code	MIT 652	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course introduces students to basic to advanced concepts and techniques applied in Statistics and Data Science domain. Students will gain theoretical as well as practical knowledge on various topics of Descriptive Statistics, Inferential Statistics, Statistical Modeling. This course will also familiarize students with recent data science tools for solving practical problems.

2. Course Objectives

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

- Familiarize students with random events and their probabilities
- Familiarize students with some advance knowledge of Statistical inference.
- Provide students with theoretical and practical knowledge statistical modeling.
- Allow students to explore state-of-the-art data science tools for solving problems.
- Let students gain experience of doing independent study and research.
- Prepare students for both academic and industrial career in data science domain.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Explain different types of data.
- Calculate different measures of data.
- Represent data with some graphs.
- Measure probabilities with different forms of random events.
- Describe some standard form of statistical distributions.
- Perform parameter estimation and hypothesis testing mean and variance.
- Familiarize with bi-variate probability distribution.
- Measure association and dependency of two or more variables.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Descriptive Statistics (6 hrs)

- Nominal, Ordinal, Ratio and Scale Data
- Measures of Central Values, Variation, Skewness, Kurtosis

- Graphical Representation of Scale Data using Stem-and-Leaf Plot, Frequency Curve, Histogram, Boxplot, Scatter Plot

Unit 2: Probability **(8 hrs)**

- Random Experiment, Sample Space, Events, Trials
- Classical, Empirical, Subjective and Axiomatic Approaches of Probability
- Mutual Exclusiveness of Events, Independence of Events, Addition Rule
- Conditional Probability, Multiplication Rule
- Partition of Sample Space, Total Probability Rule
- Bayes' Rule

Unit 3: Distribution of Random Variables **(10 hrs)**

- Random Variables – Discrete and Continuous
- Probability Functions of Random Variables
- Expectation and Variance of Random Variables
- Moments of Random Variables and Moment Generating Function
- Discrete Probability Distributions – Uniform, Binomial, Poisson, Geometric, Hypergeometric Distributions
- Continuous Probability Distributions – Uniform, Normal, Exponential, Gamma Distributions

Unit 4: Inferential Statistics **(12 hrs)**

- Sampling Distributions – χ^2, t and F Distributions
- Parameter Estimation – Method of Moments, Method of Maximum Likelihood
- Test of Hypothesis – Neyman-Pearson Lemma, Likelihood Ratio Test
- Interval Estimation and Test of Hypothesis Concerning Mean of a Normal Population
- Interval Estimation and Test of Hypothesis Concerning Variance of a Normal Population
- Interval Estimation and Test of Hypothesis Concerning Difference in Means of Two Normal Populations (Both Independent and Dependent Samples)
- Interval Estimation and Test of Hypothesis Concerning Ratio of Variances of Two Normal Populations
- ANOVA
- p-value of a Test

Unit 5: Bivariate Probability Distributions **(6 hrs)**

- Joint Probability Mass Function, Marginal Probability Mass Function, Conditional Probability Mass Function
- Joint Probability Density Function, Marginal Probability Density Function, Conditional Probability Density Function
- Expectation, Variance and Covariance of Bivariate Random Variables

Unit 6: Correlation and Regression **(6 hrs)**

- Measure of Correlation of Bivariate Random Variables – Pearson's Correlation Coefficient, Spearman's Rank Correlation, Test of Significance of Correlation
- Simple Linear / Non-Linear Regression

- Multiple Linear Regression
- Test of Significance of Regression Coefficients
- Test of Significance of Regression Model
- Measure of Accuracy of Regression Models – MSE, MAE, MAPE, R-Square, Adjusted R-Square

4.2 Laboratory Work (16 hrs)

Use MS-Excel/ STATA/ R/ SPSS to-

- Obtain statistical values of numerical data
- Obtain frequency distribution/ relative frequency/ cumulative-frequency distribution of categorical data
- Bin continuous data and obtain frequency distribution
- Graphing categorical data- bar plot, pie chart, line graph
- Graphing continuous data- histogram, stem-and-leaf plot, boxplot
- Develop probability distribution of discrete random variable
- Obtain probabilities related to binomial, Poisson and normal distributions
- Fitting of binomial, Poisson and normal distributions
- Test of hypothesis on mean of a normal population
- Test of hypothesis on difference in means of two normal populations (dependent and independent samples)
- Test of hypothesis on variance of a normal population
- Test of hypothesis on ratio of variance of two normal populations.
- Develop joint probability distribution and marginal distributions of two-dimensional discrete random variables.
- Obtain correlation between two samples
- Carry simple linear/ non-linear regression
- Carry multiple linear regression

5 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

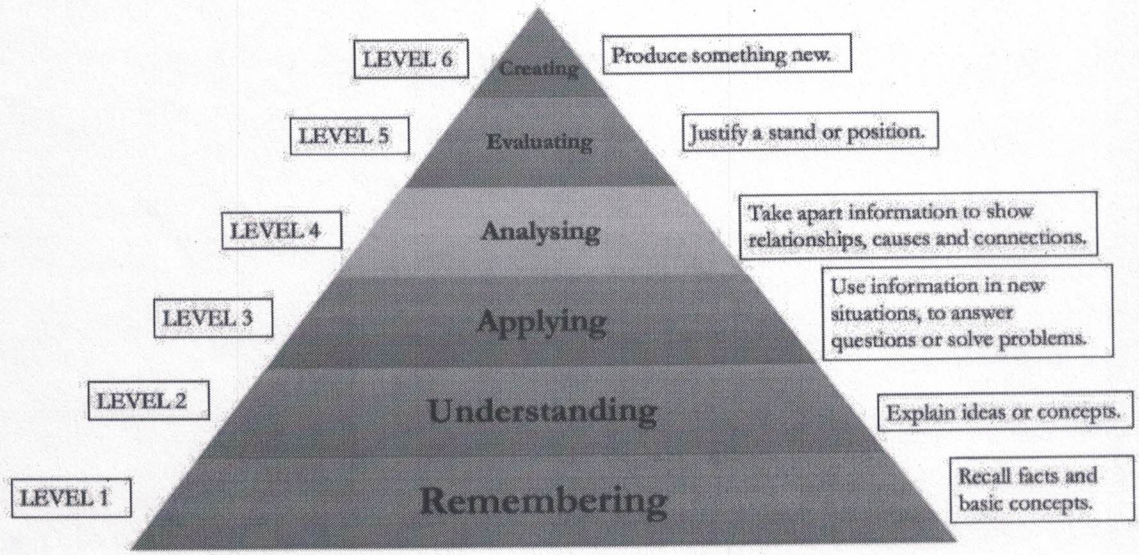
6 Books (4-10 books):

- Michael Baron Probability and Statistics for Computer Scientists,2019. CRC Press Taylor & Francis Group

- Biswas, S. (1991): Topics in Statistical Methodology, Wiley Eastern, India
- Montgomery, Douglas C.; Runger, George C.: Applied Statistics and Probability for Engineers 7th Edition, Wiley
- Bhat, B.R. (1999): Modern Probability Theory - An Introductory Textbook, New Age International, New Delhi
- Walpole, Ronald; Myers, Raymond; Myers Sharon; Ye, Keying- Probability and Statiscits for Engineers and Scientists Pearson Publication

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.

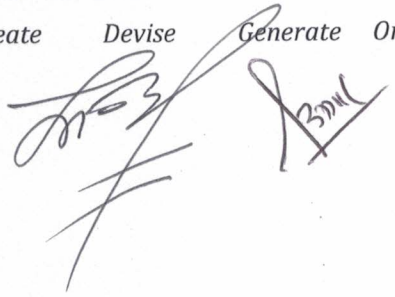
Analyse *Appraise* *Compare* *Contrast* *Distinguish* *Explore* *Infer* *Investigate*

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*



Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Software Project Management**

Course Code	MIT 653	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course, designed for M.Tech students in Information Technology and Computer Science, provides a comprehensive understanding of Software Project Management principles and practices. It covers project planning, effort estimation, risk management, human factors, leadership, and software quality assurance. Students will explore theoretical concepts and practical techniques to effectively manage software projects, addressing challenges such as project constraints, risks, and team dynamics. Through hands-on exercises and case studies, students will apply tools like Gantt charts, PERT/CPM, and COCOMO models to real-world scenarios, preparing them to lead successful software projects.

2. Course Objectives

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

- Understand the characteristics, constraints, and challenges of software projects and their management.
- Develop skills in project planning, including work breakdown structures, activity sequencing, and resource allocation.
- Master software effort estimation techniques, such as LOC, Function Point, and COCOMO models.
- Identify, analyze, and mitigate risks in software projects to ensure successful outcomes.
- Explore human factors, leadership, and team organization to foster effective collaboration and conflict resolution.
- Apply software quality assurance techniques, including Formal Technical Reviews (FTR) and Cleanroom Methodology.
- Gain practical experience in using project management tools and methodologies to monitor and control software projects.

3. Learning Outcomes

Upon successful completion of the course, students will be able to:

- Explain the key characteristics, constraints, and reasons for software project failures.
- Create comprehensive project plans, including Work Breakdown Structures (WBS), Gantt charts, and PERT/CPM diagrams.

- Apply software effort estimation techniques to predict project costs and timelines accurately.
- Identify, prioritize, and develop response strategies for software project risks.
- Demonstrate effective leadership, communication, and conflict resolution skills in managing project teams.
- Implement software quality assurance processes, including reviews and reliability measures.
- Use project management tools to monitor progress, control resources, and evaluate project performance using Earned Value Analysis.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Introduction of software Project Management (8 hrs)

- Projects and Project Characteristics
- Project Constraints
- Problems with Software Projects
- Software Project Failures & Major Reasons
- Software Project Management
- Project Management Framework Project
- Stakeholders
- Project Organization Types
- Project Charter.

Unit 2: Project Planning (8 hrs)

- Definition Planning
- Planning Tasks
- Work Breakdown Structure (WBS),
- Activity Planning
- Activity Sequencing
- Time Scheduling
- Gantt Chart
- PERT/CPM
- SQA and Test Plan
- Resource Plan
- Communication Plan
- Project Monitoring and Control
- Earned Value Analysis

Unit 3: Software Effort Estimation (8 hrs)

- Software Effort Estimation
- Need for Software Estimation
- Software Estimation Process
- Software Estimation Techniques
- Expert Judgment based Estimation
- LOC, Function Point and Object point Analysis

- COCOMO cost estimation method

Unit 4: Project Risk Management **(8 hrs)**

- Risk Identification,
- Top 10 Software Project Risks
- Risk Analysis and Prioritization
- Risk Response Planning
- Risk Resolution
- Risk Tracking and Control

Unit 5: Human Factors and Leadership Team Organization, Contract Management **(8 hrs)**

- Motivation
- Communication,
- Handling Difficult People
- Leadership and health safety
- Conflict resolution
- Introduction to contract management and types of contracts
- Stages on contract, placement, typical terms of a contract, contract management, acceptance

UNIT 6: SOFTWARE QUALITY **(8 hrs)**

- Software Quality - Quality Measures - FURPS - Software Quality Assurance - Software Reviews - Format Technical Review (FTR)
- Formal Approaches to SQA - Software Reliability
- Introduction to SQA - The Software Quality Assurance Plan - Formal approaches to SQA - Clean room Methodology

4.2 Laboratory Work **(16 hrs)**

Lab	Practical Title
1	Review of 3 fictions proposal, review of Software requirement specification (SRS). Students are asked to develop the proposal and SRS.
2	Compare and contrast the different types of SDLC
3	Develop and design the project planning.
4	Develop detailed activity networks and schedules (CPM/ PERT)
5	Testing tools.

5. Evaluation Scheme

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

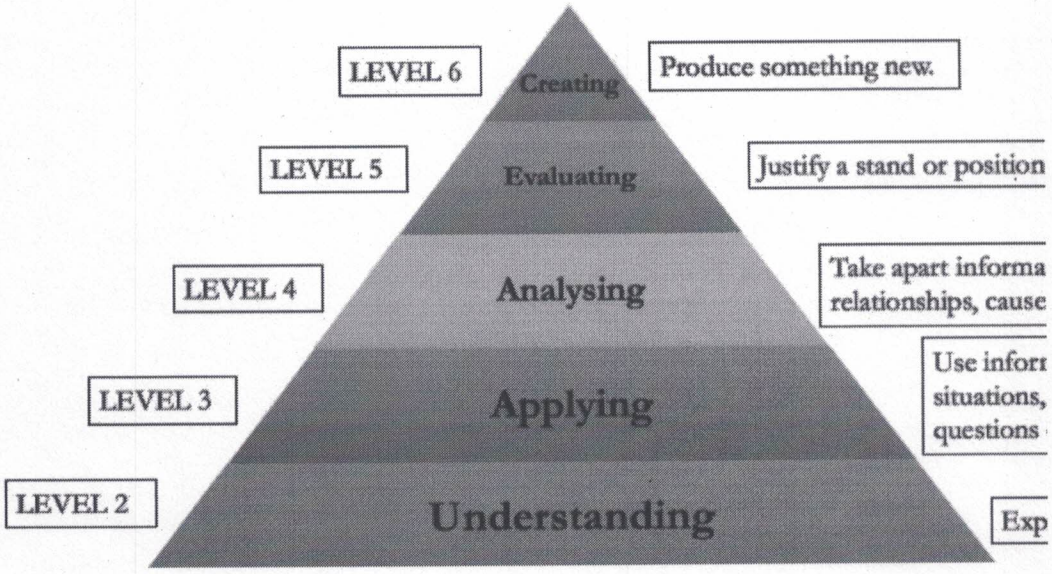
Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6. Books(4-10 books):

- Bob Hughes and Mike Cotterell: Software Project, Management latest edition, McGraw-Hill,
- Bruegge and Allen H. Dutoit (2010) Object Oriented Software Engineering – using UML, Third Edition, Prentice Hall,
- Effective Project Management: Traditional, Agile, Extreme by Robert K. Wysocki (Wiley)
- Software Quality Assurance: From Theory to Implementation by Daniel Galin (Pearson)
- Agile Project Management with Scrum by Ken Schwaber (Microsoft Press)
- Project Management: A Systems Approach to Planning, Scheduling, and Controlling by Harold Kerzner (Wiley)

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

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Data Mining and Warehouse**

Course Code	MIT 654	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course, tailored for M.Tech students in Information Technology and Computer Science, provides an in-depth study of Data Mining and Data Warehousing. It covers theoretical concepts, practical methodologies, and applications for extracting meaningful patterns from large datasets. Students will explore data preprocessing, association rule mining, classification, clustering, and advanced mining of complex data types. The course also addresses data warehousing, including multidimensional data models, OLAP technology, and data cube development. Through practical lab exercises, students will apply these concepts using tools like Python, SQL, and data mining software, preparing them to tackle real-world data challenges.

2. Course Objectives

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

- Understand the principles, architectures, and functionalities of data mining and data warehousing systems.
- Master data preprocessing techniques, including cleaning, transformation, and reduction, to prepare datasets for analysis.
- Develop proficiency in data mining techniques such as association rule mining, classification, and clustering.
- Explore advanced data mining methods for spatial, multimedia, temporal, text, and web data.
- Design and implement data warehouse systems, including multidimensional models and OLAP operations.
- Apply visualization techniques to represent input data and mined knowledge effectively.
- Evaluate the social, ethical, and technical challenges associated with data mining applications.
- Gain hands-on experience with industry-standard tools and programming languages for data mining and warehousing.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:s

- Articulate the importance, functionalities, and architectures of data mining and data warehousing systems.
- Perform data preprocessing tasks (cleaning, integration, transformation, reduction) to prepare datasets for mining.
- Implement and analyze association rules, classification models, and clustering algorithms using appropriate algorithms.
- Design and query data warehouses using multidimensional models and OLAP operations.
- Apply specialized algorithms to mine complex data types, including spatial, temporal, and text data.
- Use visualization techniques to represent data mining results and knowledge effectively.
- Evaluate the performance of data mining models using metrics like accuracy, precision, and recall.
- Develop practical solutions to real-world data problems using tools like Python, R, or SQL.
- Assess the social and ethical implications of data mining applications

4. Course Details

4.1 Theory (48 hrs)

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

- Introduction to Data Mining
- Importance of Data Mining
- Data Mining functionalities
- Classification of Data Mining Systems
- Data mining architecture
- Major Issues in Data Mining
- Applications of Data Mining
- Social Impacts of Data Mining

Unit 2: Data warehouse **(6 hrs)**

- Introduction to Data Warehouse and OLAP Technology for Data Mining
- Multidimensional data Model
- Data Warehouse Data Model
- Data Warehouse Architecture
- Data Warehouse Implementation
- Development of Data Cube Technology
- From Data warehousing to Data Mining

Unit 3: Data Cleaning **(9 hrs)**

- Matrix-Chain Multiplication
- Elements of Dynamic Programming
- Longest Common Subsequence
- 0/1 Knapsack problem
- Traveling Salesman problem
- Data Preprocessing
- Data cleaning
- Data Integration and Transformation: standardization, normalisation, smoothing, aggregation, generalization

- Data reduction: Data Cube Aggregation, Dimensionality reduction, Numerosity Reduction, Discretization, and Concept Hierarchy Generation
- Data Mining primitives
- Languages and System Architectures
- Concept Description : Characterization and Comparison, Analytical Characterization, Mining Class Comparison
- Representing input data and output knowledge: Visualization techniques
- Guidelines for Successful Data Mining

Unit 4: Association rule in data mining **(8 hrs)**

- Association Rule Mining
- Mining of single-dimensional Boolean association rules
- Multilevel association rules and Multidimensional association rules
- Correlation Analysis
- Constraint-based association Mining.

Unit 5: Classification and Predication **(7 hrs)**

- Basic issues regarding classification and predication
- Classification by Decision Tree
- Bayesian classification
- Classification by back propagation
- Associative classification
- Prediction
- Classifier accuracy

Unit 6: Cluster Analysis **(7 hrs)**

- Cluster Analysis
- Basic issues
- Clustering using partitioning methods
- Hierarchical methods,
- Density-based methods
- Grid-based methods and model-based methods,
- Algorithms for outlier analysis.

Unit 7: Mining Complex Types of Data **(5 hrs)**

- Multidimensional analysis and descriptive mining of complex data objects
- Introduction to spatial mining, multimedia mining, temporal mining, text mining and web mining with related algorithms.

4.2 Laboratory Work **(16 hrs)**

- Install and explore data mining tools (e.g., Weka, RapidMiner, or Python libraries like scikit-learn). Load a sample dataset (e.g., Iris or Titanic) and generate basic summary statistics

- Design a star schema for a retail sales data warehouse. Implement it using SQL and perform OLAP queries (e.g., roll-up, drill-down) using a database tool like PostgreSQL or MySQL.
- Preprocess a noisy dataset (e.g., Kaggle's House Prices dataset) by handling missing values, normalizing data, and discretizing continuous attributes. Use Python libraries for implementation
- Implement the Apriori algorithm in Python to mine association rules from a market basket dataset (e.g., Kaggle's Grocery dataset).
- Develop a decision tree and Bayesian classifier for a dataset (e.g., UCI's Breast Cancer dataset) using scikit-learn. Compare their performance using cross-validation
- Implement K-means and DBSCAN clustering on a dataset (e.g., UCI's Mall Customers dataset). Identify outliers using Isolation Forest. Visualize clusters using Python.
- Perform text mining on a dataset (e.g., Twitter sentiment dataset) using Python's NLTK or spaCy. Extract topics using LDA and visualize word clouds. Optionally, explore web scraping for data collection

5 Evaluation Scheme

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

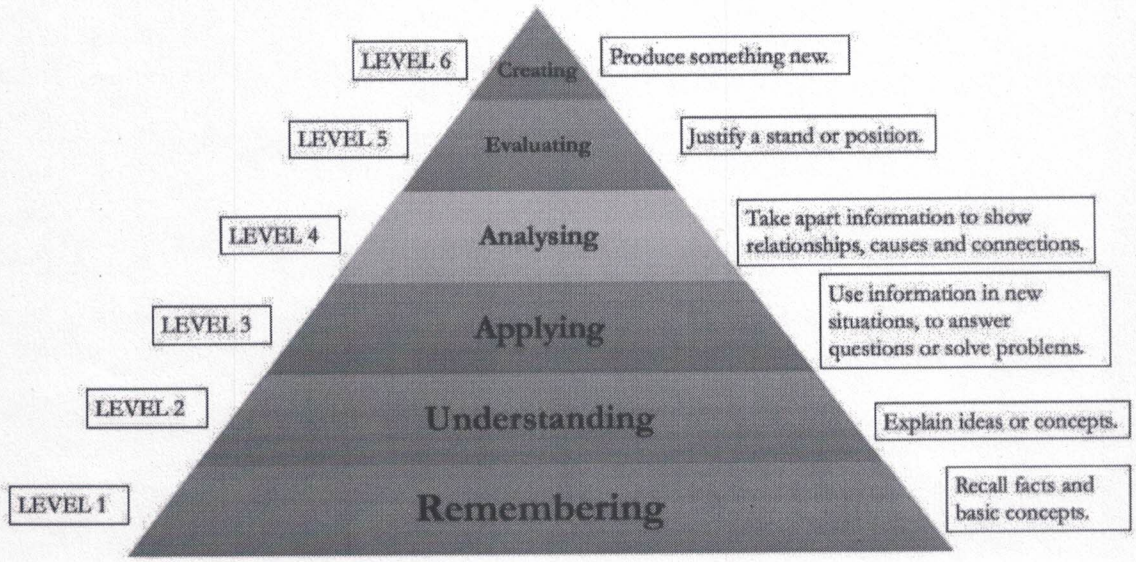
Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6 Books (4-10 books):

- Data Mining Concepts and Techniques by Jiawei Han, Micheline Kamber– Elsevier
- Data Mining by Arun K. Pujari – University Press
- Modern Data Warehousing, Data Mining and Visualization by George M. Marakas –Pearson
- Data Mining by Vikram Puri And P.Radha Krishana –Oxford Press
- Data Warehousing by Reema Theraja –Oxford Press

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

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Project Development**

Course Code	MIT 655	Year/Semester	I/II
Credit Weightage	2	Lecture	N/A
		Tutorial	N/A
		Practical	3 hrs/wk
		Total	48 hrs

1. Course Description

This course provides students with hands-on experience in designing, developing, and managing a complete software or research project. Students will learn project planning, requirement analysis, system design, implementation, testing, and documentation. The course emphasizes teamwork, professional ethics, and the application of modern tools and methodologies to solve real-world problems.

2. Course Objectives

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

- Plan, design, and execute a complete software or research project.
- Apply project management and software engineering principles to organize work effectively.
- Utilize modern development tools and technologies to develop scalable and maintainable solutions.
- Collaborate effectively in teams and communicate project outcomes professionally.
- Document, present, and defend the project results comprehensively.

3. Learning Outcomes

Upon completing this course, Students will be able to:

- Define project objectives, scope, and deliverables clearly.
- Conduct requirement analysis and feasibility studies.
- Design system architecture and select appropriate technologies.
- Develop working prototypes and full-scale project solutions.
- Implement testing strategies, including unit testing, integration testing, and user acceptance testing.
- Prepare professional project reports, presentations, and documentation.
- Apply ethical standards and professional practices in project development

4. Project Implementation Guidelines

4.1 Planning and Scope

- Clearly define project objectives, scope, and expected outcomes.
- Prepare a project proposal including problem statement, motivation, literature survey, methodology, and expected deliverables.
- Break the project into phases: requirement analysis, design, development, testing, and documentation.

4.2 Technology and Tools

- Select appropriate programming languages, frameworks, and platforms.
- Use database management systems if required (SQL/NoSQL).
- Incorporate version control (e.g., Git/GitHub) for collaborative development.
- Consider cloud services or virtualization tools if relevant for deployment or testing.

4.3 Design & Architecture

- Use UML diagrams (Class, Sequence, Activity, Use Case) to describe system design.
- Prepare data flow diagrams (DFD), ER diagrams, and database schema if applicable.
- Clearly define module responsibilities and interfaces.
- Include considerations for scalability, maintainability, and performance optimization.

4.4 Development & Coding Standards

- Follow modular coding practices with proper naming conventions.
- Comment the code adequately for clarity and maintainability.
- Implement error handling and logging mechanisms.
- Follow best practices in security, data handling, and resource management.

4.5 Testing & Quality Assurance

- Perform unit testing, integration testing, system testing, and user acceptance testing (UAT).
- Maintain a test plan with test cases, expected results, and actual results.
- Include performance testing and optimization where relevant.
- Document bugs, fixes, and improvements in a test log.

4.6 Project Deliverables

- Working software or research prototype.
- Source code, well-commented and version-controlled.
- User manual or installation guide if applicable.
- Final report and presentation slides.

5. Project Report Guidelines

The report should be professional, clear, and well-structured. Recommended sections are:

5.1 Front Matter

- Title page with project title, student names, roll numbers, supervisors, department, and university.
- Acknowledgements (optional).
- Abstract (150–250 words summarizing the project objectives, methods, results, and conclusions).

5.2 Table of Contents

- Include list of figures, tables, and abbreviations.

5.3 Introduction

- Background and context of the project.
- Motivation and significance of the problem.
- Objectives and scope of the project.

5.4 Literature Review

- Discuss previous work and related research.
- Highlight gaps or limitations that your project addresses.
- Include citations in standard formats (APA, IEEE, or university guideline).

5.5 Requirement Analysis

- Functional and non-functional requirements.
- System specifications and constraints.
- Use Use Case Diagrams or Requirement Tables where necessary.

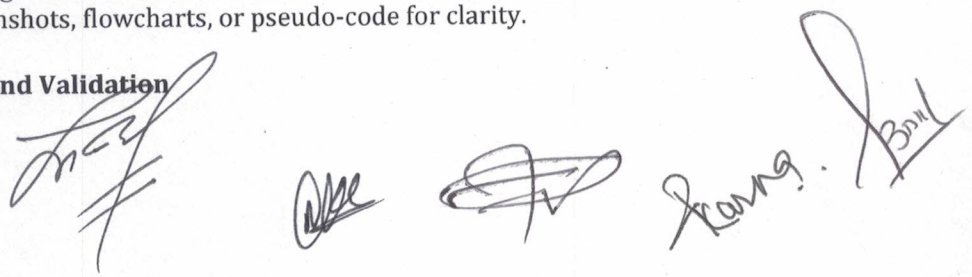
5.6 System Design

- High-level and detailed design of the system.
- Architectural diagrams (UML, DFD, ER diagrams).
- Database design, schemas, and data flow.

5.7 Implementation

- Programming languages, tools, and libraries used.
- Description of modules and workflow.
- Key algorithms, data structures, or methodologies implemented.
- Screenshots, flowcharts, or pseudo-code for clarity.

5.8 Testing and Validation



- Test plan and test cases.
- Testing methods: unit, integration, system, acceptance.
- Observations, bug fixes, and validation results.
- Performance metrics (if applicable).

5.9 Results and Discussion

- Present outcomes of the project.
- Discuss performance, accuracy, efficiency, or usability.
- Compare with baseline or expected outcomes.
- Include charts, graphs, tables, or screenshots for evidence.

5.10 Conclusion and Future Work

- Summarize achievements and lessons learned.
- Limitations of the current work.
- Recommendations and scope for future enhancements.

5.11 References

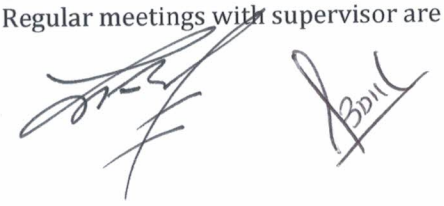
- List all cited books, papers, websites, and other resources in IEEE/APA format.

5.12 Appendices

- Source code (optional if too large), configuration files, additional diagrams.
- User manual or deployment instructions.

6. General Guidelines

- Page formatting: A4, 1.5 line spacing, 12-pt font, Times New Roman or Arial.
- Figures & Tables: Numbered and properly referenced in the text.
- Plagiarism: Ensure originality; follow university anti-plagiarism policies.
- Presentation: Prepare slides for final defense (10–15 min). Include objective, methodology, results, and conclusion.
- Supervision: Regular meetings with supervisor are mandatory for feedback and guidance.



7. 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	Project Proposal & Requirement Analysis	10	10	5
	Progress/Milestone Reports	10	10	5
	Implementation/Coding/ Prototype	15	15	7.5
	Practical Demonstration & Testing	10	10	5
	Project Report & Documentation	10	10	5
	Presentation/Seminar/Defense	5	5	2.5
	Total Internal Marks		60	30
Semester-End Examination	Final Project Evaluation	40	40	20

Notes/Guidelines for Evaluation:

- **Project Proposal & Requirement Analysis:** Assessment of clarity of objectives, feasibility, and completeness of requirements.
- **Progress/Milestone Reports:** Regular progress submissions, adherence to timelines, and supervisor feedback.
- **Implementation/Coding/Prototype:** Quality, modularity, correctness, and functionality of code or system.
- **Practical Demonstration & Testing:** Successful demonstration of working system, testing coverage, and handling of edge cases.
- **Project Report & Documentation:** Completeness, clarity, formatting, references, and alignment with implementation.
- **Presentation/Seminar/Defense:** Ability to communicate project objectives, methodology, results, and answer questions confidently.
- **Semester-End Evaluation:** Combined assessment by internal and external examiners of final deliverables and presentation.

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Knowledge Management**

Course Code	MIT 656	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course provides a comprehensive understanding of Knowledge Management (KM), focusing on the systematic processes for capturing, organizing, sharing, and utilizing knowledge within organizations to enhance efficiency, innovation, and competitive advantage. It covers theoretical foundations, practical tools, and strategies for implementing KM systems, with an emphasis on real-world applications. Students will engage in hands-on activities, case studies, and projects to develop skills in designing and managing KM initiatives.

2. Course Objectives

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

- Understand the core concepts, principles, and frameworks of Knowledge Management.
- Develop skills to design and implement effective KM strategies in organizational contexts.
- Learn to use KM tools and technologies for knowledge creation, storage, and dissemination.
- Analyze the role of organizational culture and leadership in fostering knowledge-sharing environments.
- Apply KM principles through practical projects to solve real-world organizational challenges.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Articulate the principles, theories, and frameworks of knowledge management, including the differences between tacit and explicit knowledge.
- Explain the role of KM in enhancing organizational performance, innovation, and competitive advantage.
- Evaluate knowledge creation, storage, sharing, and application processes within various organizational contexts.
- Develop strategies to foster a knowledge-sharing culture, leveraging both technological and human-centric approaches.
- Design KM initiatives, including communities of practice, knowledge repositories, and collaborative platforms, tailored to organizational goals.
- Apply knowledge management tools (e.g., content management systems, intranets, and AI-driven platforms) to facilitate knowledge capture and dissemination.

- Analyze case studies to identify best practices and challenges in KM implementation across industries.
- Propose solutions to real-world KM problems, integrating ethical considerations and stakeholder perspectives.
- Integrate KM practices with organizational learning strategies to drive adaptability and growth.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Introduction to Knowledge Management (4 hrs)

- Definition and scope of KM
- Types of knowledge (tacit vs. Explicit
- KM history and evolution, and its importance in organizations

Unit 2: Knowledge Management Frameworks and Models (6 hrs)

- KM lifecycle, Von Krogh and Roos Model of Organizational Epistemology
- Nonaka's SECI model, Wiig's KM cycle
- European Foundation for Quality Management (EFQM) KM Model and other frameworks

Unit 3: Knowledge Application and Organization Culture (10 hrs)

- Knowledge Application at the Individual Level and Bloom's Taxonomy of Learning
- Task Analysis and Modelling
- Knowledge Application at the Group and Organizational Levels
- Knowledge Reuse and Repositories, E-Learning and Knowledge Management Application
- Strategic Implications of Knowledge Application
- Introduction of Organizational Culture and Analysis
- Culture at the Foundation of KM
- Organizational Maturity Models
- KM Maturity Models
- CoP Maturity Models

Unit 4: Knowledge Creation, Capture, Storage, and Organization (9 hrs)

- Knowledge Capture and Creation Tools
- Tools and Techniques for knowledge creation, communities of practice, storytelling, and knowledge elicitation methods
- Content Creation Tools
- Data Mining and Knowledge Discovery, Blogs, Mashups, Content Management Tools, Knowledge Sharing and Dissemination Tools
- Knowledge Acquisition and Application Tools, Intelligent Filtering Tools, Adaptive Technologies, Strategic Implications of KM Tools and Techniques, Practical Implications of KM Tools and Techniques
- Knowledge repositories, taxonomies, metadata, and content management systems

Unit 5:Knowledge Sharing and Dissemination (8 hrs)

- Knowledge-sharing culture
- Barriers to sharing

- Collaboration Tools
- Groupware and Collaboration Tools, Wikis, Social Networking, Web 2.0, and KM 2.0, Networking Technologies

Unit 6: Knowledge Management Technologies and strategies **(7 hrs)**

- KM software
- AI and machine learning in KM
- Document management systems and data analytics
- Introduction of KM strategies
- Developing a Knowledge Management Strategy
- Knowledge Audit, and Gap Analysis
- The KM Strategy Road Map
- Balancing Innovation and Organizational Structure
- Types of Knowledge Assets Produced

Unit 7: Implementing and Evaluating KM systems **(4 hrs)**

- KM strategy development
- Change management, performance metrics, and ROI of KM initiatives
- Conduct a knowledge audit to identify knowledge assets in a simulated organization

4.2 Laboratory Work **(16 hrs)**

- Conduct a knowledge audit to identify knowledge assets in a simulated organization.
- Practical Component: Create a visual representation of a KM model and apply it to a case study.
- Facilitate a brainstorming session to capture tacit knowledge from team members.
- Design a knowledge database structure using a tool like Microsoft Access or a cloud-based platform.
- Implement a knowledge-sharing platform (e.g., using Microsoft Teams or Slack) for a group project.
- Practical Component: Explore and evaluate a KM tool (e.g., Confluence, SharePoint) through a hands-on demo.
- Conduct a knowledge audit to identify knowledge assets in a simulated organization.

5 Evaluation Scheme

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

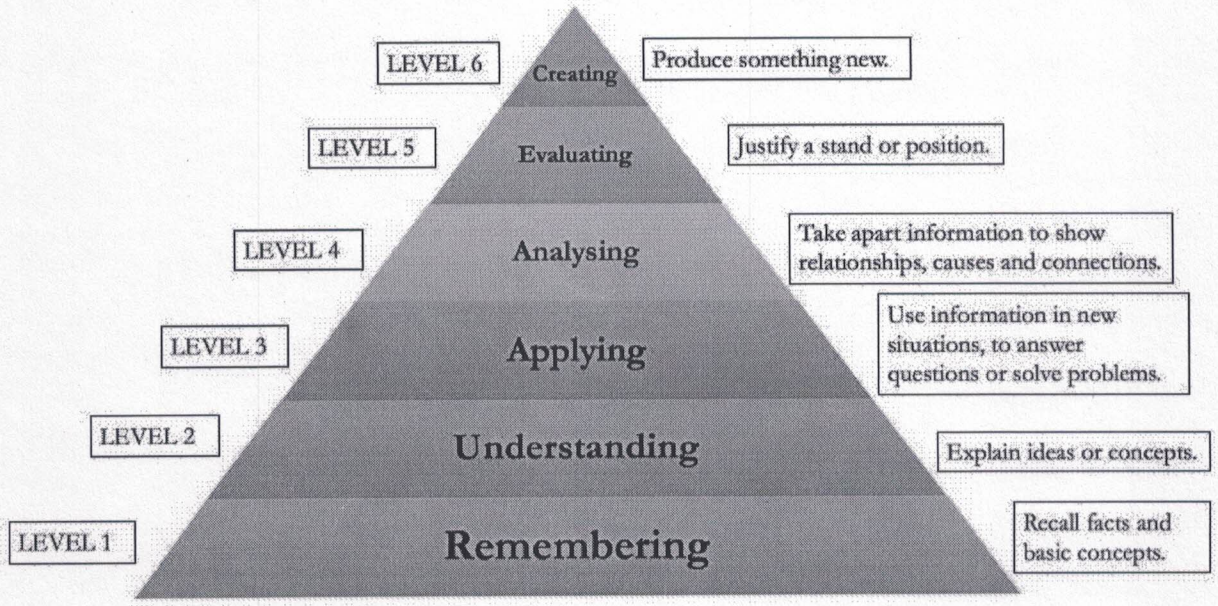
Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6 Books(4-10 books):

- Books: "Knowledge Management in Theory and Practice" by Kimiz Dalkir, "The Knowledge-Creating Company" by Ikujiro Nonaka.
- Tools: Confluence, SharePoint, Microsoft Teams, Trello.
- Articles: Harvard Business Review on KM, case studies from KMWorld.

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*

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: e-Governance**

Course Code	MIT 657	Year/Semester	I/II
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course explores the principles, frameworks, and practices of digital governance, focusing on how organizations and governments manage digital technologies, data, and policies to ensure ethical, transparent, and effective operations. Students will examine the intersection of technology, policy, and society, addressing challenges such as data privacy, cybersecurity, digital equity, and regulatory compliance in a rapidly evolving digital landscape.

2. Course Objective

To equip students with technical and strategic expertise to design, implement, and evaluate digital governance systems that enhance public service delivery, citizen engagement, and socio-economic equity in diverse contexts.

3. Learning Outcomes

Upon completion, students will be able to:

- Architect interoperable e-governance systems using cloud, AI, and data frameworks (e.g., Nepal's GIDC).
- Evaluate security, privacy, and legal compliance in digital public services (e.g., Nepal's Cyber Bylaw).
- Design inclusive solutions addressing digital divides
- Manage PPP projects and agile governance transformations (e.g., Nagarik App rollout).
- Critique ethical implications of emerging tech (AI, blockchain) in public decision-making

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Foundations & Citizen-Centric Transformation (6 hrs)

- Overview of e-Governance:
- Definition, Importance, and Objectives.
- Types of interaction of e-Governance(G-2-G, G-2-B, G-2-C, G-2-Employee),
- Theory of e-Governance (Rainbow Theory, Kytoon Theory, .3 Smog Theory, Virga Theory, Tempest Theory)
- Pillar of e-Governance,
- Impact of e-Governance,
- difference between e-Governance and e-Government,
- Case study:National ID card vs Nagarik App(Unified citizen portal for 150+ services), Relevance: Embodies "citizen focus" by integrating services from driving licenses to tax

payments, ConnectIPS (Digital payments platform), Impact: Transformed financial inclusion (8M+ users), enabling citizen-centric utility bill payments, remittances, and government fee processing.

Unit 2: Evolution and origin of e-Governance, Strategic Architecture & Interoperability **(7 hrs)**

- EVOLUTION OF e—GOVERNANCE, e-Governance master plan-2006 to Digital Nepal Framework VER 2.0, History and the Origin of e —Governance in Nepal
- e -Government Evolution in Different Parts of the World
- Evolution of Korea e —Governance, Estonia, e-Governance maturity models (World Bank, UN and others maturity model), Core Components: Data models, service layers, APIs, cloud infrastructure, Interoperability Frameworks: Standards, gateways, legacy integration , State Data Centers & Digital Infrastructure
- Case Study: Singapore's Tradenet – Cross-agency interoperability
- National e-Governance Master Plan (NeGMP) ,Framework: Aims for integrated systems across 7 pillars (e-identity, payments, data exchange),Case Study: Government Integrated Data Centre (GIDC), Role: Central hub hosting critical systems (e.g., Land Information System) with plans for cross-agency data sharing.

Unit 3: Data-Centric Governance & Emerging Tech **(7 hrs)**

- Data Management: Warehousing, metadata, open data
- AI/ML in Governance: Predictive services, chatbots, ethical risks
- Blockchain: Secure records (land, identity), smart contracts
- IoT & Smart Cities: Sensor networks, urban governance
- Cloud & Open Source in e-Governance: Cost efficiency, vendor lock-in risks
- Blockchain in e-Governance, Cloud in e-Governance

Unit 4: Security, Privacy & Cyberlaw **(6 hrs)**

- Threat Landscape: Cyberattacks, insider threats, data breaches
- Zero-Trust Architecture: Authentication, encryption
- Privacy Laws: GDPR, India's DPDP Act, consent frameworks
- Cyberlaw Essentials: Digital signatures, liability, jurisdiction
- Case Study: Estonia's X-Road – Secure data exchange
- Example: Cyber Security policy of Nepal
- Policy: Mandates incident reporting for critical infrastructure.
- Case Study: Nepal Rastra Bank's Payment System Security
- Practice: NRB-grade encryption for financial transactions amid rising phishing attacks.

Unit 5: Inclusive Governance & Digital Equity **(5 hrs)**

- Digital Divide: Access, skills, affordability (rural/urban)
- Inclusive Design: Multilingual interfaces, accessibility standards
- Shared Access Models: eSeva kiosks, mobile-first services
- Community Co-Creation: Citizen feedback loops
- E —Governance portals around the Globe 111
- Case Study: Land record management system in Nepal and Bhoomi (India) – Digitizing land records for farmers

Unit 6: Public-Private Partnerships & Agile Implementation **(5 hrs)**

- PPP Models: BOOT, ASP, co-creation ecosystems

- e-Procurement: Transparency, fraud prevention
- Agile Rollout: Pilots vs. big-bang, iterative design
- Change Management: CIO role, capacity building
- Case Study: UK's Government Digital Service (GDS) – Agile service delivery

Unit 7: Future Trends & Transformation **(4 hrs)**

- AI Governance: Algorithmic accountability, bias mitigation
- Platform Ecosystems: Unified citizen portals (India Stack, UAE's Smart Pass)
- Metaverse/AR: Virtual town halls, immersive training
- Sustainable Tech: Green data centres, digital sobriety
- Rebooting Government: AI-driven policy labs, anticipatory governance
- Case Study: Dubai Paperless Strategy – End-to-end digital government

Unit 8: E-Government portal **(8 hrs)**

- E-government portal of 3 tier government in Nepal, e —governance ranking in the un government survey, telecommunication and infrastructure index (tii), human capital index, online service index (osi), e —governance ranking across the globe and among, Nepal's e —governance ranking in the un government, the e —governance in top 10 countries - an overview
- Issues and opportunities of e —governance in Nepal
- Opportunities of e —governance, challenges for e —governance in Nepal
- Environmental and social challenges, economical challenges, technical challenges, challenges in implementation of e —governance applications, other challenges of e —governance, security drawbacks, funding, management of change

4.2 Laboratory Work **(16 hrs)**

Students will understand the practical implementation of e-governance systems and gain hands-on experience with citizen-centric digital services. They will apply emerging technologies, including AI/ML, blockchain, and IoT, in governance scenarios to enhance service delivery and efficiency. In addition, students will analyze challenges related to security, interoperability, and digital inclusion, ensuring that digital services are accessible and reliable for all citizens. Throughout the course, they will also develop presentation and reporting skills, enabling them to effectively communicate the design, implementation, and impact of e-governance projects.

5. Evaluation Scheme

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

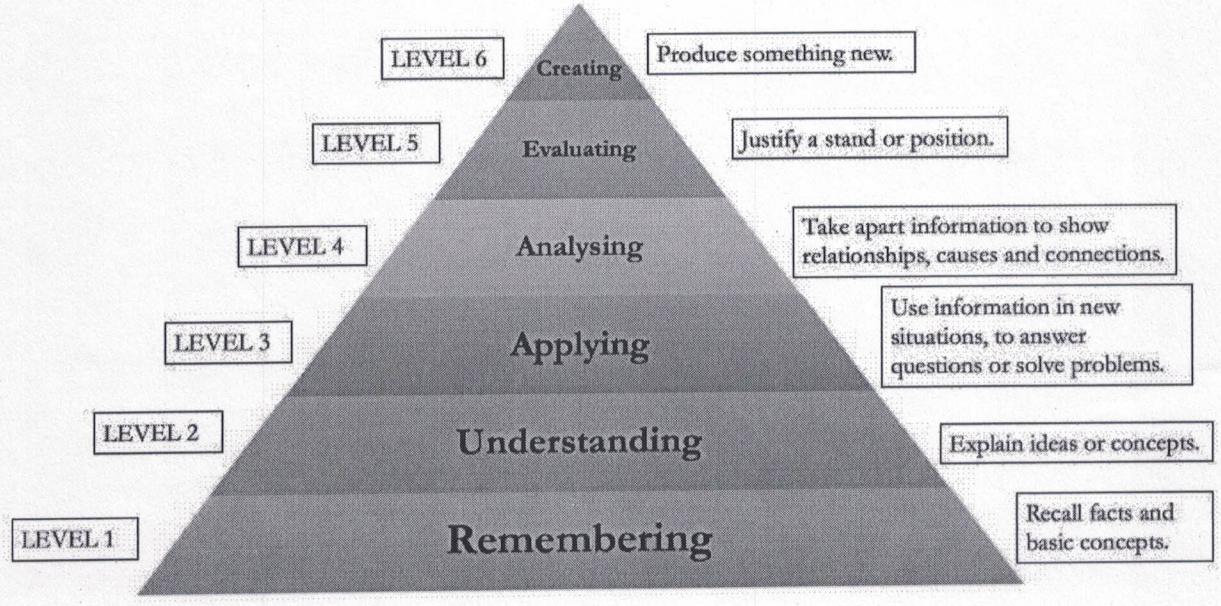
Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

4 **Books (4-10 books):**

- e-Government: The Science of the Possible by J. Satyanarayana
- e-Governance: A Comprehensive Framework by D.N. Gupta
- Research Handbook on E-Government edited by Eric Welch
- Electronic Government: Concepts, Methodologies, Tools, and Applications by Ari-Veikko Anttiroiko
- E-Governance: Styles of Political Judgment in the Information Age Polity by Perri
- Electronic Governance by Thomas B. Riley
- From Government to E-Governance: Public Administration in the Digital Age by M. Shamsul Haque and Muhammad Maroof
- E-Governance in India: Initiatives & Issues by R.P. Sinha
- Digital Solutions for Contemporary Democracy and Government edited by Kelvin Joseph Bwalya and Stephen Mutula
- A Handbook of E-Governance in India by Sumathy Mohan

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*

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Cloud Computing**

Course Code	MIT 658	**Year/Semester**	I/II
Credit Weightage	3	**Lecture**	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

The Cloud Computing Ecosystem course, designed for Master of Technology students at LTU, offers a comprehensive study of cloud technologies, architectures, and service models, including IaaS, PaaS, SaaS, and FaaS. It explores cloud infrastructure components such as virtualization, containers, storage, and networking, with hands-on experience using platforms like AWS, Azure, and Google Cloud. Students will gain practical skills through tasks like deploying virtual machines, building serverless applications, and setting up Continuous Deployment/delivery (cd), Continuous Integration (CI) pipelines. The course also covers advanced topics like cloud security, compliance, AI/ML integration, and emerging trends such as multi-cloud strategies and cloud-native development. By the end, students will be proficient in designing, securing, and optimizing scalable cloud solutions for enterprise applications.

2. Course Objectives

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

- Have in-depth knowledge of cloud computing ecosystems, including architecture, services, deployment models, security, and emerging trends, through theoretical learning and practical implementation.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Understand the fundamentals and ecosystem of cloud computing.
- Differentiate between service and deployment models.
- Analyze the roles of stakeholders in the cloud ecosystem.
- Explain cloud architecture and its components.
- Understand virtualization and containerization technologies.
- Design basic cloud infrastructure setups.
- Understand and utilize various cloud service models.
- Develop and deploy applications using PaaS and FaaS.

- Implement DevOps practices in cloud environments.
- Identify and mitigate cloud security risks.
- Implement security best practices and compliance measures.
- Design disaster recovery strategies for cloud systems.
- Explore advanced cloud technologies and their applications.
- Implement AI/ML and IoT solutions in cloud environments.
- Understand sustainability in cloud computing.
- Analyse real-world cloud implementations through case studies.
- Understand emerging trends and their impact on cloud ecosystems.
- Develop strategies for multi-cloud and cloud-native environments.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Introduction to Cloud Computing System (5 hrs)

- Definition, characteristics, and principles of cloud computing
- Evolution and history of cloud computing
- Cloud service models: IaaS, PaaS, SaaS, FaaS
- Cloud deployment models: Public, Private, Hybrid, Multi-Cloud
- Key stakeholders: Providers, users, and third-party vendors
- Benefits and challenges of cloud computing

Unit 2: Cloud Architecture and Infrastructure (9 hrs)

- Cloud computing architecture
- Cloud reference architecture: NIST and IBM models. Virtualization: Hypervisors (Type 1 and Type 2), virtual machines
- Containers: Docker, Kubernetes, and container orchestration
- Storage systems: Object, block, and file storage
- Networking: Virtual networks, load balancers, CDNs
- Data centers: Design and energy efficiency

Unit 3: Cloud Services and Development (9 hrs)

- IaaS services: Compute, storage, and networking
- PaaS services: Application development and deployment platforms
- SaaS applications: Features and use cases
- Serverless computing: FaaS architecture, AWS Lambda, Azure Functions
- APIs and microservices in cloud environments
- DevOps in the cloud: CI/CD pipelines, tools (e.g., Jenkins, GitHub Actions)

Unit 4: Security and Compliance (7 hrs)

- Cloud security challenges: Data breaches, misconfigurations
- Security mechanisms: Encryption, IAM, firewalls, VPNs
- Zero Trust architecture in cloud environments
- Compliance standards: GDPR, HIPAA, PCI-DSS, ISO 27001
- Security tools: AWS Shield, Azure Security Center, Google Cloud Armor
- Disaster recovery and business continuity in the cloud

Unit 5: Advanced Cloud Technologies **(10 hrs)**

- Big Data and analytics in the cloud: Hadoop, Spark, AWS Redshift
- AI and ML in the cloud: AWS SageMaker, Google AI Platform, Azure ML
- IoT and edge computing: AWS IoT, Azure IoT Hub
- Blockchain in the cloud: Hyperledger, Ethereum on Azure
- Green computing: Sustainable cloud practices

Unit 6: Emerging Trends and Case Studies **(8 hrs)**

- Multi-cloud and hybrid cloud strategies
- Cloud-native development: CNCF tools, service meshes (e.g., Istio)
- Quantum computing in the cloud: AWS Braket, Azure Quantum
- Case studies: Netflix, Airbnb, and Spotify's cloud adoption
- Research trends: Serverless security, edge AI, federated learning
- Future of cloud computing: Challenges and opportunities
- IaaS: Amazon Web Services, Microsoft Azure, Google Cloud Platform, OpenStack
- PaaS: Google App Engine, AWS Elastic Beanstalk, Microsoft Azure App Services, and Heroku. (Anyone)
- SaaS: Google Workspace (Gmail, Docs, Sheets), Salesforce, Zoom, Microsoft 365 (Any two)

4.2 Laboratory Work **(16 hrs)**

- Set up an account on a public cloud provider (e.g., AWS Free Tier, Google Cloud, or Azure) and explore its dashboard and services.
- Deploy a simple virtual machine (IaaS) on AWS EC2 or Google Compute Engine and access it remotely.
- Write a report comparing the features of AWS, Azure, and Google Cloud based on their free-tier offerings.
- Create and manage a virtual machine using VirtualBox or VMware to understand hypervisor functionality.
- Build a Docker container to deploy a simple web application (e.g., Flask or Node.js) and push it to Docker Hub.
- Set up a Kubernetes cluster using Minikube and deploy a multi-container application with a load balancer.
- Deploy a web application on a PaaS platform (e.g., Google App Engine or Heroku) and configure auto-scaling.
- Create a serverless function using AWS Lambda to process data from an S3 bucket (e.g., image resizing).
- Set up a CI/CD pipeline using GitHub Actions to automate deployment of a web application to AWS Elastic Beanstalk.
- Configure IAM roles and policies on AWS to restrict access to S3 buckets and EC2 instances.
- Set up a VPN connection to a virtual private cloud (VPC) on Azure or AWS for secure access.
- Simulate a disaster recovery scenario by backing up and restoring data using AWS Backup or Azure Site Recovery.
- Build a machine learning model using AWS SageMaker to predict a dataset (e.g., house prices) and deploy it as an endpoint.
- Create an IoT application using AWS IoT Core to collect and visualize sensor data in real-time.

- Analyze a large dataset using Google BigQuery and visualize results with Google Data Studio.
- Design a multi-cloud architecture using AWS and Azure for a sample application, ensuring redundancy and load balancing.
- Deploy a cloud-native application using Helm charts on a Kubernetes cluster with Istio for service mesh.
- Write a research paper or present a case study on a company's cloud adoption (e.g., Netflix's use of AWS).

5 Evaluation Scheme

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

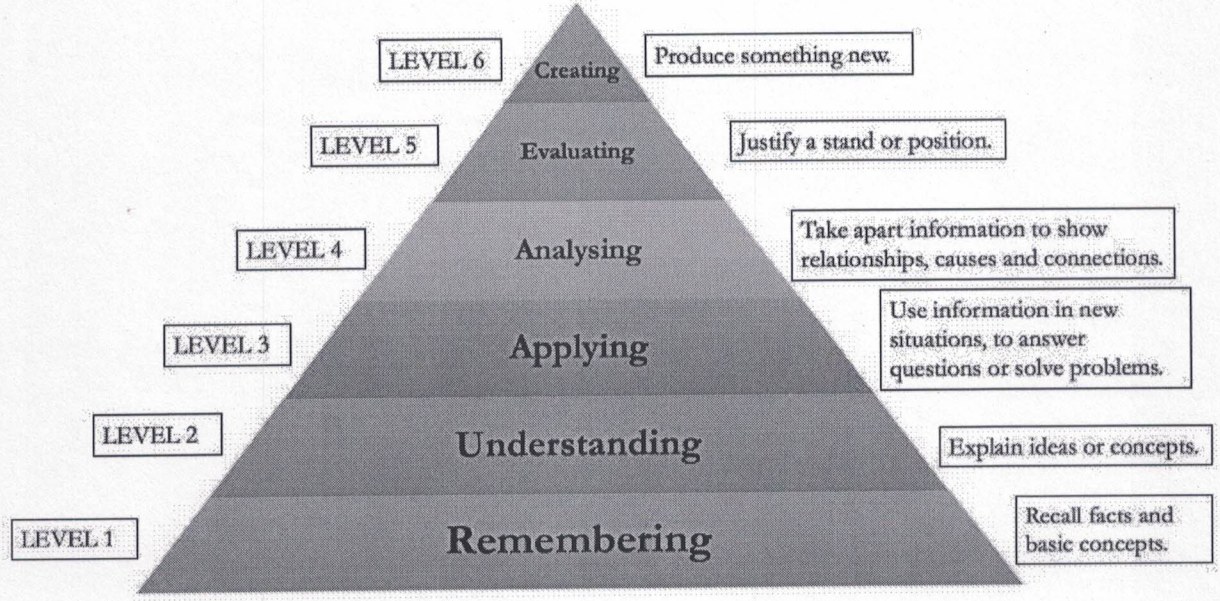
Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6 Books (4-10 books):

- Kai Hwang , Geoffrey Fox, Jack J. Dongarra, Morgan Kaufmann, Distributed and Cloud
- "" Cloud Computing: Concepts, Technology & Architecture by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, Prentice Hall, 2013
- Computing: From Parallel Processing to the Internet of Things, 1st Edition, 2011
- Gautham Shroff, "Enterprise Cloud Computing: Technology, Architecture, Applications", Cambridge press, 2010
- Rajkumar Buyya, James Broberg, Andrzej Goscinski," Cloud Computing Principles
- and Paradigms", John Wiley & Sons, 2011
- John Rhoton and Risto Haukiojal, "Cloud Computing Architectured : Solution Design Handbook", Recursive Press, 2013
- Dinkar Sitaram, Geetha Manjunathan, " Moving to the Cloud: Developing Apps in the new world of Cloud Computing", Syngress, 2012
- "Handbook on Data Centers" Samee. U. Khan, Albert. Y. Zomaya, Springer
- https://www.cloudfoundry.org/
- https://puppet.com/blog/implement-a-message-queue-your-cloud-application
- www.cloudpatterns.org
- www.IBMCloud.com
- Cloud Native DevOps with Kubernetes by John Arundel and Justin Domingus, O'Reilly Media, 2019

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*

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Machine Learning**

Course Code	MIT 701	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

The course provides a comprehensive overview of Machine Learning, aiding graduate students in grasping fundamental concepts. It starts with foundational knowledge essential for understanding various machine learning algorithms, including Supervised and Unsupervised methods, the Bias-Variance Trade-Off, Overfitting, and Underfitting. The course delivers an example-based approach, offering insight into the practical application of machine learning techniques to real-world problems through case studies and related projects.

2. Course Objectives

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

- To introduce fundamental concepts and techniques for prediction and classification.
- To enhance understanding of different problem-solving techniques with appropriate machine learning algorithms.
- To provide insight into the practical application of machine learning algorithms in real-world problem-solving scenarios.

3. Learning Outcomes

Upon completing this course, Students will be able to:

- Students will acquire foundational knowledge, understanding key principles such as the BiasVariance Trade-Off and the concepts of overfitting and underfitting.
- Students will apply their knowledge to solve complex problems, using case studies and projects as vehicles for practical application.
- Students will evaluate the effectiveness of various machine-learning techniques, critically assessing their performance and suitability for different tasks.
- Students will develop a comprehensive understanding of statistical and machine learning principles and the ability to apply them proficiently in diverse real-world contexts.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Introduction (8 hours)

- Introduction to Machine Learning
- Machine Learning Methods
- The Bias-Variance Trade-Off
- Overfitting and Underfitting

Unit 2: Linear Regression (10 hours)

- Linear Regression
- Multiple Linear Regression
- Qualitative Predictors
- Gradient Descent

Unit 3: Resampling and Linear Model Selection (10 hours)

- The validation Set Approach
- Leave-One-Out Cross Validation
- K-Fold Cross Validation
- Bias-Variance Trade-off for k-Fold Cross Validation

Unit 4: Machine Learning Algorithms (10 hours)

- Naïve Bayes
- Random Forest
- Support Vector Machines
- K-means and Fuzzy C-means

Unit 5: Machine Learning Applications and Case Studies (10 Hours)

- Machine learning Applications and Case studies
- Project Works

4.2 Laboratory Work (16 hrs)

- Real-life applications for statistical learning.

  (a) Describe three real-life applications in which *classification* might be useful. Describe the response, as well as the predictors.
  (b) Describe three real-life applications in which *regression* might be useful. Describe the response, as well as the predictors.
  (c) Describe three real-life applications in which *cluster analysis* might be useful.

- Download the Auto dataset from the textbook website, as this question involves the use of simple linear regression on the Auto dataset. .

  a. Use the sm.OLS() function to perform a simple linear regression with mpg as the response and horsepower as the predictor. Use the summarize () function to print the results. Comment on the
  b. output. For example:

     i. Is there a relationship between the predictor and the response?
     ii. How strong is the relationship between the predictor and the response?
     iii. Is the relationship between the predictor and the response positive or negative?
     iv. What is the predicted mpg associated with a horsepower of 98?
     v. What are the associated 95 % confidence and prediction intervals?

  c. Plot the response and the predictor in a new set of axes ax. Use the ax.axline() method or the abline () function defined in the lab to display the least squares regression line.
  d. Produce some of diagnostic plots of the least squares regression fit as described in the lab. Comment on any problems you see with the fit.
- Explore Cross-Validation Techniques Using Machine Learning Tools, Focusing on Chapter 3.
- Implement a machine-learning algorithm covered in Chapter 4.

5. 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6. Books(4-10 books):

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Jonathan Taylor.
- An Introduction to Statistical Learning: with Applications in Python. New York :Springer, 2023

- Jason Brownlee: Master Machine Learning Algorithms: Discover How they Work and Implement Them From Scratch. 2016

- Machine Learning. Tom M. Mitchell. McGraw-Hill International Editions. Computer Science Series, 1997.
- Machine Learning : The Art of Science and Algorithms that Make Sense of Data by Peter Flach

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Mobile Application Development**

Course Code	MIT 702	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

Android app development offers a wealth of opportunities for building innovative and feature-rich mobile applications. As Android continues to evolve, developers have the opportunity to create apps that cater to a global audience and enhance the mobile experience for users worldwide.

2. Course Objectives

After Successful completion of the above course, students will be able to:

- **Acquire** an insight into concepts of mobile application development terminologies, environment and architecture
- **Design** mobile applications using various UI components and layouts.
- **Develop** software with reasonable complexity on mobile platforms.
- **Deploy** applications on mobile devices incorporating most of the key aspects of the platform.

3. Course Details

3.1 Theory (48 hrs)

Unit 1: Introduction to Android Operating System, Development using Kotlin and UI Designer: **(4 hrs)**

Overview, Android Architecture, Android Ecosystem, Android Versioning and APIs, Development Environment, Setup and Interface, Android Emulator and ADB (Android Debug Bridge) setup, The Structure of Android's Code, Android Components, Activity Lifecycle and debugging, Migration from JAVA to Kotlin, OOP concepts in Kotlin, Functions and Lambda in Kotlin and Intents.

Unit 2: Android Resource handling, Layouts, Views and Material Design for Multi-Screen (Responsive): **(6 hrs)**

- **Resources**: String, Color, Drawable, Styles, Theme,
- **Menus**: Option and Context
- **Layouts**: Linear Layout, Frame Layout, Relative Layout, Constraint Layout, Dynamic Implementation of Layout.
- **Views**: Dialog boxes, Navigation Drawer, Snackbar, Toast and basic views

Material Design: UI widgets with properties, events and methods, and wiring up layouts with Kotlin code

- **Multi-Screen**: Orientations, and Design for all Screen Sizes

Unit 3: Advanced Views, Fragments, Adapters and Binding Models: **(5 hrs)**

- **Views**: GridView, WebView, ScrollView, ListView, RecyclerView, CardView
- **Fragment**: Introduction, life Cycle, Implementation, Adapters and Array-List, Model-View-Controller, Data-binding and View-binding with Kotlin

Unit 4: Data Storage Techniques **(4 hrs)**

Connectivity using Firebase, Authenticating Users, Real-Time Database and CRUD, Cloud Messaging using Firebase, Storage using firebase, SQL-lite Connectivity and CRUD operations, Room persistence library, Data Store preference

Unit 5: Version Control System with GitHub **(3 hrs)**

Introduction to GitHub, Making Repository, Cloning Repository, Push, Pull and Fork

Unit 6: Web Application Integration Techniques **(4 hrs)**

API Communication with Web API, Introduction to JSON Data, JSON Parsing, Working with REST API, Data Classes in Kotlin, Introduction to Glide, Loading images from URL, Google API, Web-view

Unit 7: Overview of Jetpack and Sensor Managers **(3hrs)**

Introduction to Jetpack, CameraX, Regular Sensors and Coroutines

Unit 8: Polish and Publish Application **(4 hrs)**

Different Ways to Monetize, Versioning, Signing AAB, Packaging and Beta Test of Mobile Application, Distributing Application on Mobile Market Place

3.2 Laboratory Work **(16 hrs)**

- Installation of Android studio.
- Development Of Hello World Application
- Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button
- Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout)
- Design an android application to create page using Intent and one Button and pass the Values from one Activity to second Activity
- Design an android application Send SMS using Intent
- Create an android application using Fragments
- Design an android application Using Radio buttons
- Design an android application for menu.
- Create a user registration application that stores the user details in a database table.

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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10

	Total Internal Marks		60	30
Semester-End Examination			40	20

5. Books(4-10 books):

- John Horton,"Android Programming with Kotlin for Beginners" 2nd Edition, Packt Publication.
- Pierre-Olivier Laurence, Amanda Hinchman-Dominguez, Mike Dunn, G. Blake Meike, Title: "Programming Android with Kotlin", O`Reilly.
- Reto Meier, "Professional Android 4 Application Development", John Wiley & Sons.
- Alex Forrester; Eran Boudjnah; Alexandru Dumbravan; Jomar Tigcal, "How to Build Android Apps with Kotlin", Packt.

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Big Data Analytics**

Course Code	MIT 705	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course will provide students with a broad introduction to Big Data technologies including Hadoop based architectures, data ingestion, data transformation, data management, analytics and predictive analytics for manipulating and discovering perception. The major topics include Hadoop, HDFS, MapReduce, NoSQL and Big data & Machine learning. The course also includes case studies and applications.

2. Course Objectives

- To display a sound understanding of the principles of organisation, validation, transformation and analyse large volumes of data on specialized platforms (Big Data) from various data sources files, databases, server logs, etc.
- To use the Big Data platform ecosystem for processing Big Data.
- To understand the potential use in a corporate environment.
- To comprehend the advantages and limitations of Big Data technologies, including predictive analytics and build the confidence to interpret data as insights to drive organisational success.

3. Learning Outcomes

- Ability to analyze complex data sets using big data tools and techniques.
- Applying different data mining and statistical methods for extracting big data from different sources.
- Ability to apply big data analytics techniques to real-world business and decision-making processes.
- Discuss on database management systems and data warehousing technologies.
- Knowledge on data visualization and present findings in meaningful way.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Introduction of Big Data (4 hrs)

- Big data: definition and taxonomy
- Big data value for the organization
- Setting up the demo environment
- Using Big Data for business point of view
- Role of data Scientist
- Recent Trend in Big Data Analytics

Unit 2: The Hadoop ecosystem (10 hrs)

- Introduction to Hadoop, Hadoop components: MapReduce/Hive/HBase/Yarn/Spark/Storm
- Loading data into Hadoop
- Handling files in Hadoop
- Getting data from Hadoop
- HDFS, Hadoop YARN Architecture
- Exploring Pig and Oozie
- Query Languages for Hadoop
- Hadoop and Amazon Cloud Contents

Unit 3: Querying big data with Hive (10 hrs)

- Introduction to the SQL Language
- From SQL to Hive QL
- Using Hive to query Hadoop files
- Hive for massively parallel ondisk data processing

Unit 4: NoSQL, Searching and Indexing Big Data (8 hrs)

- Structured and Unstructured Data
- Taxonomy and NoSQL Implementation,
- Architecture of Hbase, Cassandra and MongoDb
- Full text Indexing and Searching, Distributed Searching with Elastic search Indexing with Lucene

Unit 5: Big data & Machine learning (8 hrs)

- Big Data & Machine Learning
- Machine learning tools (Spark & SparkML, H2O, Azure ML)
- Big data analysis using graph databases: Analyzing big data with Neo4j using Cypher queries

Unit 6: Big data analysis using graphs (8 hrs)

- Introduction to graphs
- Graph analytics (Path analytics, connectivity analytics, community analytics, centrality analytics)

- Graph analytics with Neo4j
- Graph analytics with GraphX
- Apache Spark's API for graphs and graph parallel computation

5. Laboratory Work **(16 hrs)**

Work in big data technologies using real world problems that will cover all the aspects included in the course. It will assist students to gain practical skill in knowing about problems faced and tackle prosecuting knowledge of tools learned in course.

- HDFS: Setup a HDFS in a single node to multi node cluster, perform basic file system operation on it using commands provided, monitor cluster performance.
- Map-Reduce (MR): Write various MR programs dealing with different aspects of it as studied in course
- Hbase: Setup of Hbase in single node and distributed mode, write program to write into hbase and query it.
- Elastic Search: Setup elastic search in single mode and distributed mode, Define template, Write data in it and finally query it.

6. 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

7. Books(4-10 books) :

- Jugnesh Kumar and Anubhav Kumar. Big Data and Analytics: The key concepts and practical applications of big data analytics, 2024.
- EMC Education Services. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, 2015.
- Thomas H. Davenport. Big Data at Work: Dispelling the Myths, Uncovering the Opportunities
- David Loshin. Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph.

- Balamarugan Balusamy, Nandhini Abirami R, Seifedine Kadry and Amir Gandomi. Big Data: Concepts, Technology and Architecture. Wiley Publisher.

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Human-Computer Interaction**

Course Code	MIT 706	Year/Semester	II/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course emphasizes the growing importance of understanding users and their environments, especially as computer technology becomes ever more pervasive and consumer-centered. At its core, Human-Computer Interaction (HCI) focuses on designing technologies that are secure, easy to use, and efficient—ensuring a meaningful and satisfying experience for users.

2. Course Objectives

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

- To analyze and apply the core components of interactive technology design—including usability, accessibility, and user experience (UX) principles—to create effective digital solutions.
- To evaluate the theoretical frameworks and methodological foundations (e.g., cognitive psychology, design thinking, and iterative prototyping) that guide interface design and user interaction.
- To conduct user-centered research and usability testing to assess interactive artifacts, employing both qualitative and quantitative evaluation techniques.
- To interpret usability data and feedback to diagnose design flaws and propose evidence-based improvements for enhanced user satisfaction and performance.
- To advocate for ethical and equitable design practices, ensuring technologies cater to diverse user needs while addressing societal and cultural impacts.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

- Upon completing this course, Students will be able to:
- understand core HCI principles
- apply user-centered design methods
- analyze user behavior and cognition
- design and prototype interactive systems
- conduct and apply usability evaluations

4. Course Details

4.1 Theory (48 hrs)

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

- Ergonomics and Human Factors
- Foundational Concepts of Interaction Design; Good and poor design; Usability and user experience goals
- Computer-Supported Cooperative Work (CSCW)
- Collaborative Technologies (e.g., Google Docs, Zoom)

Unit 2: Design Lifecycle **(8 hrs)**

- User-Centered Design Concepts
- Participatory Design
- Usability Engineering ; Identifying users; Understanding user requirements, goals, skills, and environment

Unit 3: Psychology of HCI **(8 hrs)**

- Human Perception, Cognition, and Emotion; Attention, memory, emotion,
- User Behavior, Accessibility, and Diversity
- Applying Physical World Knowledge to Digital Interfaces

Unit 4: Methods for Studying Users in Context **(6 hrs)**

- Observation Techniques; Data collection and analysis
- Interviews and Questionnaires ; Types of interviews (structured, unstructured, ...) ; Survey design and analysis

Unit 5: Principles for Usability in UI Design **(6 hrs)**

- Usability Guidelines ; Navigation, display organization, data entry
- Design Principles ; Interaction styles, preventing errors, automation
- Theories of Interface Design ; GOMS, stages-of-action, widget-level theories

Unit 6: Prototyping Methods **(6 hrs)**

- Low- and High-Fidelity Prototyping
- Conceptual Design Techniques; Scenarios, models, and iterations
- From Design to Implementation

Unit 7: Evaluation Methods **(6 hrs)**

- Expert Reviews; Heuristic evaluation, cognitive walkthroughs, ...
- Usability Testing
- Evaluation During Active Use; Surveys, focus groups, data logging, online feedback

Unit 8: Mini-Project Development and Integration **(2 hrs)**

- Application of Full HCI Lifecycle; From design to prototype to evaluation

4.2 Laboratory Work **(16 hrs)**

Students are required to conceptualize and develop a project that applies Human-Computer Interaction principles and techniques covered throughout the course. The evaluation will be based

on the relevance and originality of the project topic, its practical significance, design, and technical quality, depth of HCI integration, user-centered approach, report clarity, and effectiveness of the final presentation.

5 Evaluation Scheme

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

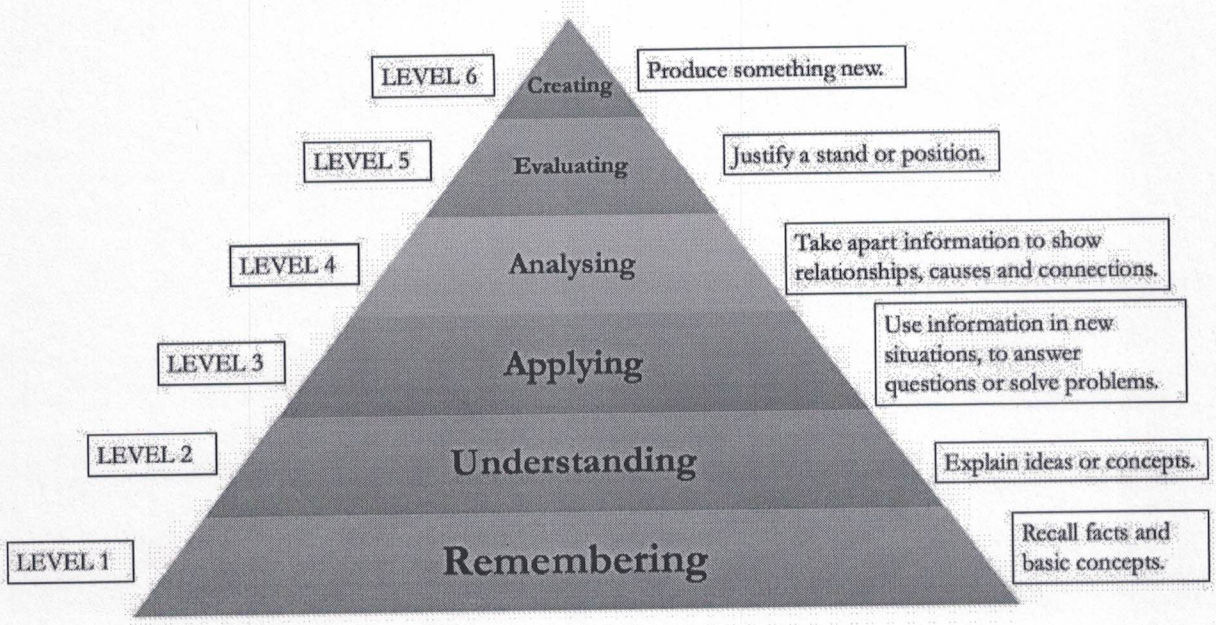
Evaluation	Components	Weightage	Full Marks	Pass Marks
Internal Evaluation	Class Attendance and Performance	5	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6 Books (4-10 books):

- Helen Sharp, Jenny Preece, and Yvonne Rogers. Interaction Design: Beyond Human-Computer Interaction. "In John Wiley", 2011.
- Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs. Designing the User Interface: Strategies for Effective Human-computer Interaction, "Pearson Education", 2017.
- Donald A. Norman. The Design of Everyday Things. "Basic Books", 2013.

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*

Lumbini Technological University
Estd: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Information Security Audit**

Course Code	MIT 707	Year/Semester	I/I
Credit Weightage	3	Lecture	3 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course equips students with the knowledge and practical skills to perform effective Information Systems (IS) Audits in modern environments. Covering core concepts, security issues, risk-based auditing, and emerging challenges, it emphasizes hands-on application using current tools and techniques relevant to cloud, hybrid infrastructure, DevOps, data privacy regulations, and e-commerce. Students will learn to assess controls, identify vulnerabilities, analyze risks, and evaluate business continuity in alignment with industry standards and market demands.

2. Course Objectives

The specifics objectives of the course are:

- To explore concepts and evolving frameworks of IS auditing.
- To understand hardware, software, cloud, and network security issues in contemporary environments.
- To plan, conduct, and report on IS audits using practical methodologies and tools.

3. Learning Outcomes

Upon successful completion of the Knowledge Management course, learners will be able to:

- Articulate the principles, theories, and frameworks of IS Audit
- Explain the role of IS Audit in enhancing organizational performance, innovation, and competitive advantage.
- To apply risk-based principles to audit planning, scoping, and execution.
- To design and audit business continuity and disaster recovery plans focused on ransomware and cloud resilience.
- To evaluate security controls and compliance specifically within e-commerce, mobile, API, and cloud ecosystems.
- To understand and perform fundamental security testing techniques (vulnerability scanning, log analysis).
- To analyze the impact of data privacy regulations (GDPR, CCPA) and third-party risk on IS audits.
- Analyze case studies to identify best practices and challenges in IS Audit implementation across industries.
- Propose solutions to real-world IS Audit problems, integrating ethical considerations and stakeholder perspectives.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Foundations of Modern IS Auditing (5 hrs)

- Information Systems Audit: Evolution & Role
- The IS Auditor: Skills, Knowledge (CISA), & Ethics
- Legal & Regulatory Landscape: SOX, GDPR, CCPA, HIPAA implications
- Systems Environment: On-Prem, Cloud (IaaS/PaaS/SaaS), Hybrid, DevOps
- Information Assets: Data Classification (PII, PHI, IP)
- Classification of Controls: Preventive, Detective, Corrective
- Impact of Cloud, Mobile, IoT on Information & Auditing
- IS Audit Coverage & Scope

Unit 2: Infrastructure & Hardware Security (4 hrs)

- Hardware Security Objectives in Cloud Era
- Endpoint Security: Laptops, Mobile, IoT, and Storage Media Encryption
- Cloud Infrastructure Security Considerations: Hypervisors, SDN
- Authentication Devices: MFA, Biometrics, Hardware Tokens
- Acquisition: Cloud Service Provider (CSP) Assessments, SLAs
- Maintenance & Obsolescence: Patch Management Verification
- Disposal: Data Sanitization Standards (NIST SP 800-88)
- Problem & Change Management: Auditing CMDBs & Processes

Unit 3: Software & Development Security (5 hrs)

- Software Types: OSS Licensing & Compliance Risks, Container Security, SaaS
- Elements of Software Security: SDLC Security (Secure Coding), DevSecOps
- Control Issues: CI/CD Pipeline Security, Container Image Scanning, Infrastructure as Code (IaC) Security
- Open Source License Management & Audit
- API Security Fundamentals
- Problem & Change Management: Auditing in Agile/DevOps

Unit 4: Risk-Based IS Audit Fundamentals (6 hrs)

- Audit Requirements & Planning
- Modern Risk Analysis Frameworks: NIST RMF, ISO 27005
- Threats, Vulnerabilities, Likelihood, Impact, Risk Appetite
- IS Control Objectives: COBIT 2019, NIST CSF mapping
- IS Audit Objectives: Effectiveness, Efficiency, Compliance (Privacy!)
- Information Systems Abuse & Fraud Detection
- Asset Safeguarding: Data Lifecycle Controls
- Evidence Collection: Electronic Evidence Handling, Chain of Custody
- Logs & Audit Trails: SIEM Concepts, Centralized Log Management

Unit 5: Conducting the IS Audit (7 hrs)

- Audit Program & Plan: Agile Audit Techniques
- Procedures & Approaches: Data Analytics (ACL, IDEA basics), Continuous Auditing/Monitoring
- System Understanding: Documenting Cloud Architectures, Microservices

- Compliance & Substantive Testing: Automated Testing Scripts
- Auditing Tools: Vulnerability Scanners (Nessus, OpenVAS), Configuration Compliance Scanners
- Sampling Techniques: Data-Driven Auditing
- Audit Documentation: Electronic Workpapers
- Audit Report: Clear Communication of Technical Findings

Unit 6: Advanced Risk-Based Audit **(4 hrs)**

- Integrating Risk into Audit Lifecycle
- Risk Assessment: Quantitative Methods Intro, Risk Heat Maps
- Risk Matrix & Prioritization
- Risk & Audit Sample Determination: Focusing on High-Risk Areas
- Audit Risk Assessment Model: Inherent, Control, Detection Risk
- Third-Party Risk Management (TPRM) Audit Considerations

Unit 7: Business Continuity & Disaster Recovery **(8 hrs)**

- BC/DR Process: Ransomware Focus, Cloud DR Strategies (AWS DR, Azure Site Recovery)
- Business Impact Analysis (BIA): RTO/RPO for Critical Cloud Services
- Cyber Incident Response Plan (CIRP) Integration
- DR Plans: Cloud-Based Failover, Testing in Cloud Environments
- Auditing Resilience against Ransomware: Multi-Cloud, Immutable Backups
- Plan Testing: Tabletop Exercises, Simulation Drills
- Checklists & Maintenance: Automating DR Validation Checks

Unit 8: Auditing Digital & E-Commerce **(5 hrs)**

- Expanded Scope: Mobile Commerce (m-Commerce), API Ecosystems, Digital Payment Gateways
- Audit Objectives: API Security, Mobile App Security, Fraud Prevention
- General Overview: OWASP Top 10 for Web, API, Mobile
- Auditing Functions: Shopping Cart, Payment Processing (PCI DSS Intro), User Authentication, Inventory Management (Cloud)
- Policies & Procedures: Acceptable Use, Data Retention, Incident Response for Digital
- Internal Control Impact: Segregation in Microservices, Tokenization, Fraud Analytics

Unit 9: Security Testing & Cyber Threats **(6 hrs)**

- Cybersecurity Landscape: Ransomware, Supply Chain Attacks, Phishing
- Cybercrimes: Business Email Compromise (BEC), Cryptojacking
- Attack Surfaces: Cloud Misconfigurations, APIs, SaaS Settings, Employees
- Attack Vectors: Exploit Kits, Credential Stuffing, Zero-Days
- Vulnerability Assessment vs. Penetration Testing
- Vulnerability Management Process Lifecycle
- Introduction to Penetration Testing (Ethical Hacking) Concepts
- Cyber Forensics: Digital Evidence Acquisition Basics, Volatility
- Auditor's Role in Security Testing Review & Oversight

4.2 Laboratory Work **(16 hrs)**

- Practical 1: Case Study - Identifying applicable regulations for a given scenario; exploring CISA domains.
- Practical 2: Lab - Reviewing cloud provider (AWS/Azure/GCP) security documentation & SLAs; Simulating hardware disposal audit steps.
- Practical 3: Lab - Using SAST/DAST tools (e.g., OWASP ZAP, Bandit) on sample code; Analysing OSS licenses for compliance; Reviewing IaC templates for misconfigurations.
- Practical 4: Workshop - Performing a qualitative risk assessment on a cloud migration project; Defining audit objectives based on NIST CSF controls.
- Practical 5: Simulation - Creating an audit program for a SaaS application; Using a vulnerability scanner (e.g., OpenVAS) against a test environment; Documenting findings in a sample audit report section.
- Practical 6: Exercise - Building a risk matrix for an e-commerce platform; Prioritizing audit tests based on risk assessment output.
- Practical 7: Workshop - Reviewing a sample cloud DR plan; Designing a tabletop exercise scenario for a ransomware attack; Auditing backup encryption and immutability settings.
- Practical 8: Lab - Analysing e-commerce site/auth flows for OWASP Top 10 vulnerabilities (using proxies like Burp Suite Community); Reviewing PCI DSS SAQ requirements.
- Practical 9 : Lab - Interpreting vulnerability scan reports (critical/high findings); Basic log analysis with Splunk/ELK demo; Hands-on with OSINT tools (e.g., the Harvester); Overview of a pen test report structure.

5. 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6. Books (4-10 books):

- Books: "Knowledge Management in Theory and Practice" by Kimiz Dalkir, "The Knowledge-Creating Company" by Ikujiro Nonaka.
- Tools: Confluence, SharePoint, Microsoft Teams, Trello.
- Articles: Harvard Business Review on KM, case studies from KMWorld.

Lumbini Technological University
Est: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Geographic Information System**

Course Code	MIT 708	**Year/Semester**	II/I
Credit Weightage	3	**Lecture**	2 hrs/wk
		Tutorial	N/A
		Practical	3 hrs/wk
		Total	96 hrs

1. Course Description

This course introduces the principles and applications of GIS for spatial data management, analysis, and visualization. Students will learn to work with geospatial data models, coordinate systems, and analytical techniques to address real-world challenges in environmental science, urban planning, and resource management. The course emphasizes conceptual understanding and problem-solving skills applicable across industry-standard tools.

2. Course Objectives

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

- Understand core GIS concepts, including spatial data structures and coordinate systems.
- Design and query spatial databases using relational and object-oriented approaches.
- Perform advanced spatial analyses (e.g., network routing, terrain modeling).
- Create cartographically sound maps for diverse audiences.

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Compare vector, raster, and TIN data models for suitability in different scenarios.
- Transform spatial data between coordinate systems and projections.
- Execute overlay, proximity, and interpolation analyses.
- Develop a watershed model from elevation data.
- Critique map designs using principles of visual hierarchy and accessibility.

4. Course Details

4.1 Theory (48 hrs)

Unit 1: GIS Fundamentals (4 hrs)

- Definitions, history, and societal applications.
- Components of GIS : Hardware, data, methods, and people.
- Spatial vs. Non-spatial data : discrete vs. continuous phenomenon.
- Spatial Thinking : Real-world applications in disaster response, public health, and smart cities.

Unit 2 : Georeferencing and Data Models (6 hrs)

- Coordinate systems (geographic, projected) and datums.
- Vector data: Topology, spaghetti models, and attribute tables.
- Raster data: Resolution, compression, and thematic mapping.

Unit 3: Data Acquisition and Database **(6 hrs)**

- Acquiring spatial data (Vector and raster) and non-spatial data.
- Digitization, digitization process and errors.
- Primary sources: Surveys, GPS, and remote sensing.
- Secondary sources: Open data, crowdsourcing.
- Data interoperability challenges and solutions.
- Relational databases: Normalization and SQL for spatial queries.
- Quality Control: Addressing positional errors, metadata standards (ISO 19115).

Unit 4: Spatial Analysis Techniques **(7 hrs)**

- Vector Analysis: Buffering, overlay (union/intersect), and spatial joins.
- Advanced Geoprocessing uses.
- Raster Analysis: Cost surfaces, zonal statistics, and suitability modeling.
- Network Analysis: Shortest path algorithms, service area delineation.

Unit 5: Terrain & Environmental Modeling **(5 hrs)**

- DEM Applications: Slope stability, floodplain mapping, and solar radiation models.
- Hydrology: Automated watershed delineation

Unit 6: Cartography & Map Making **(4 hrs)**

- Introduction to Cartography
- Cartography Development: GIS and Modern Map Making
- Cartographic Process
- Types of Maps: Reference and Thematic
- Map Design: Colour theory, label placement, and accessibility
- Elements of Map and Map Output Formats

4.2 Laboratory Work **(16 hrs)**

Lab	Practical Titles	Hours
1	Georeferencing & Coordinate Systems	2 hrs
2	Digitization	2 hrs
3	Spatial Database Design	2 hrs
4	Vector Data Analysis	2 hrs
5	Raster-Based Modeling	2 hrs
6	Terrain Analysis	2 hrs
7	Participatory GIS	2 hrs
8	Cartographic Design	2 hrs

For each lab work, students are supposed to implement the algorithm with performance analysis.

5. Evaluation Scheme

The evaluation of students' 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5

	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

6. Books (4-10 books):

- Chang, K. (2019). Introduction to Geographic Information Systems (9th edition). New York: McGraw-Hill Education, Penn Plaza.
- Bolstad, P. (2016). GIS Fundamentals: A First Text on Geographic Information Systems (5th edition). United States: Eider Press.
- Jensen, J. R., Jensen, R. R. (2013). Introductory Geographic Information Systems. Boston: Pearson.
- Campbell, J. E., Shin, M. (2012). Geographic Information System Basics (v. 1.0), Creative Commons.
- Principles of Geographic Information Systems: An Introductory Textbook. (2004). Netherlands: International Institute for Geo-Information Science and Earth Observation.
- Huisman, O., Rolf A. deBy (Ed.) (2009). Principles of Geographic Information Systems: An Introductory Textbook. The Netherlands: ITC Educational Textbook Series.
- Longley, P. A., Goodchild M. F., Maguire, D. J., Rhind, D. W. (2015). Geographic Information Science and Systems (4th edition). NJ: John Wiley& Sons, Inc.
- Anthamatten, P. (2021). How to Make Maps - An Introduction to Theory and Practice of Cartography. Taylor & Francis Group.
- Sickle, J. V. (2004). Basic GIS Coordinates. Boca Raton: CRC Press LLC.

Lumbini Technological University
Est: 2079 BS (2022 AD)
Banke, Nepalgunj

Level: Master of Technology **Subject: Remote Sensing**

Course Code	MIT 711	Year/Semester	II/I
Credit Weightage	3	Lecture	2 hrs/wk
		Tutorial	N/A
		Practical	1 hrs/wk
		Total	64 hrs

1. Course Description

This course explores the principles, technologies, and applications of remote sensing (RS) for Earth observation. Students will learn to acquire, process, and analyze satellite/airborne imagery using digital tools, with a focus on environmental monitoring, urban planning, and disaster management. The course integrates theoretical foundations (e.g., electromagnetic spectrum, sensor physics) with hands-on labs (e.g., image classification, change detection) using software like QGIS, Google Earth Engine, and ENVI (Environment for Visualizing Images).

2. Course Objectives

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

- Analyze the electromagnetic spectrum and its interaction with Earth surface for RS applications.
- Evaluate sensor types (optical, SAR, LiDAR) and platforms (satellites, UAVs) for diverse use cases.
- Process and classify satellite imagery using preprocessing, enhancement, and machine learning techniques.
- Apply RS data to real-world problems (e.g., NDVI for agriculture, SAR for flood mapping).

3. Learning Outcomes (Follow Bloom's Taxonomy to the extent possible. See Annex A)

Upon completing this course, Students will be able to:

- Remember key RS concepts (e.g., spectral signatures, resolution types).
- Apply radiometric/geometric corrections to raw imagery.
- Analyze image statistics (histograms, PCA) for feature extraction.
- Evaluate classification accuracy using confusion matrices/Kappa coefficients.
- Create GIS-integrated RS projects (e.g., land-use change maps).

4. Course Details

4.1 Theory (48 hrs)

Unit 1: Fundamentals of Remote Sensing (10 hrs)

- Introduction : Definition, history, and applications (agriculture, forestry, disasters)
- Electromagnetic Spectrum : VIS, NIR, SWIR, TIR, microwave bands.
- Energy-Matter Interaction : Reflection, absorption, spectral signatures.
- Resolution Types: Spatial, spectral, temporal, radiometric trade-offs.

Unit 2: Sensors and Platforms (8 hrs)

- Optical Sensors : Multispectral (Landsat, Sentinel-2) vs. Hyperspectral.
- Active Sensors : SAR (Sentinel-1), LiDAR (topography, forestry).
- Platforms : Satellites (geo/polar-orbiting), UAVs, and aerial photogrammtry.

Unit 3: Image Preprocessing (8 hrs)

- Radiometric Correction : Atmospheric scattering (Dark Object Subtraction).
- Geometric Correction : Georeferencing, orthorectification.
- Image Fusion : Pan-sharpening, NDVI calculation.

Unit 4: Image Classification (8 hrs)

- Supervised: SVM, Random Forest.
- Unsupervised: K-Means, ISODATA.
- Accuracy Assessment: Confusion matrices, Kappa coefficient.

Unit 5: Advanced Applications of Remote Sensing (8 hrs)

- Change Detection: Time-series analysis (urban sprawl, deforestation).
- Thermal RS: Land surface temperature monitoring.
- Disaster Management: Flood mapping (SAR), wildfire detection (NIR).

Unit 6: Emerging Trends in Remote Sensing (6 hrs)

- AI/ML in RS: Deep learning for object detection.
- CubeSats and Hyperspectral Cubes.

4.2 Laboratory Work (16 hrs)

Lab	Practical Title	Hours
1	Download/visualize Optical Remote Sensing datasets	2 hrs
2	Using Spectral Profiles to Distinguish Land Cover Types in Remote Sensing	2 hrs
3	Band Ratio and Spectral Indices Analysis for Land Use and Land Cover (LULC)	2 hrs
4	Supervised classification (urban vs. forest)	2 hrs
5	SAR-based flood mapping	2 hrs
6	Change detection (2010–2025 land cover)	2 hrs
7	Thermal image analysis (urban heat islands)	2 hrs
8	AI-Driven Object Detection in Remote Sensing Using Optical Imagery	2 hrs

For each lab work, students are supposed to implement the algorithm with performance analysis.

5. 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	25	12.5
	Assignment	5		
	Seminar/Project/Presentation	15		
	Practical/Lab Examination	15	15	7.5
	Internal Examination	20	20	10
	Total Internal Marks		60	30
Semester-End Examination			40	20

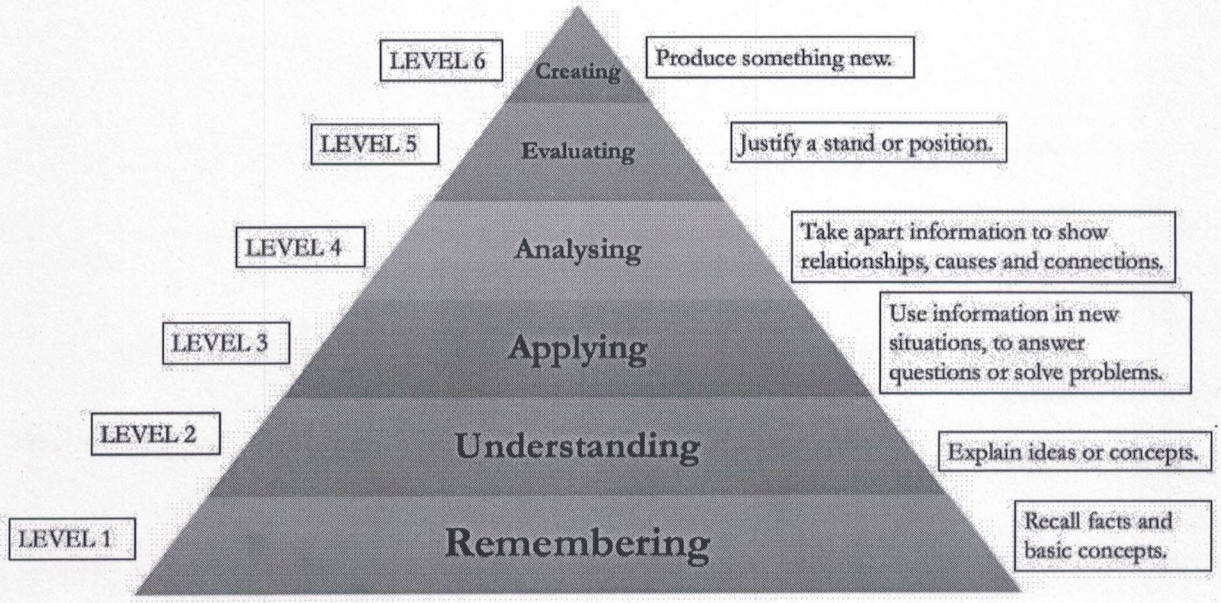
6. Books (4-10 books):

1. Lillesand, T., Kiefer, R. W., & Chipman, J. (2015). Remote Sensing and Image Interpretation. Wiley.
2. Jensen, J. R. (2015). Introduction to Remote Sensing. CRC Press.
3. Campbell, J. B., & Wynne, R. H. (2011). Introduction to Remote Sensing. Guilford Press.
4. Schott, J. R. (2007). Remote Sensing: The Image Chain Approach. Oxford University Press.
5. Mather, P. M., & Koch, B. (2011). Computer Processing of Remotely-Sensed Images: An Introduction. Wiley.
6. Pettorelli, N. (2013). The Normalization of Remote Sensing Data. Wiley-Blackwell.
7. Turner, W., J. L. T., & Spector, S. S. (2015). Free and Open Access Data for Remote Sensing Applications. Springer.
8. Kramer, G. (2002). Observing the Earth and the Universe: The Relevance of Remote Sensing for Global Change Research. Springer.
9. **ESA** (2023). *Sentinel Application Guides* (Open Access)

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*