Far Western University

Faculty of Education

CSIT in Education

Course Title: Mathematics for ICT	Nature of course: Theory
Course No. : CS.Ed.123	Credit Hour: 3
Level: Undergraduate	Teaching Hrs: 48
Semester: 2nd	

1. Course Introduction

This course is designed for the second semester students of B. Ed. in CSIT to equip them with mathematical concepts and skills. This course will help them to study several contents of different courses of the CSIT program. All the students enrolled in this program need to study this subject. Therefore, the content of the course is designed in such a way that students who have not taken mathematics as major/additional subject in grade XI and XII can also deal with this subject. The prerequisites of the course are algebraic skills, arithmetical skills, concept of real number line, concept of coordinate plane, and problem solving skill. It deals with sets, function, calculus, matrix, and counting principles.

2. General Objectives

At the end of the course the students are expected to achieve the following objectives:

- To demonstrate skills of set operations through Venn-diagrams.
- To show understanding and skills concerning multiple representations of functions.
- To show understanding and skills of limit and continuity of functions.
- To perform differentiation and integration of some basic functions.
- To demonstrate understanding and skills of performing operation between matrices.
- To show ability of using counting principles in solving contextual problems.
- To be engaged in understanding and applying concepts and skills of calculus, matrix, and counting principles.
- To be confident on the learning of skills, concepts, formulae and applications of calculus, matrix, and counting principles.

3. Course Contents and Specific Objectives

Specific Objectives	**Content**
• To explain the concept of set. • To describe different types of sets with examples. • To find union, intersection, difference and complement of sets. • To show relation among sets (at most three) in	**Unit I: Sets** [5] 1.1 Introduction of sets 1.2 Set operations 1.3 Venn-diagram 1.4 Use of Venn-diagram in performing set operations 1.5 Cardinal Number of sets [through

Venn-diagram. • To use Venn-diagram in performing set operations and solving problems concerning cardinal numbers.	Venn-diagram]
• To explain several concepts associated with matrix. • To construct examples of different types of matrices. • To find scalar multiplication, sum, difference, product of matrices. • To find value of determinant expanding from any row or column (at most order 3). • To state properties of determinant (without proof) and apply them in evaluating value of determinant. • To complete a project work assigned by teacher regarding application of matrix and determinant in ICT.	**Unit II: Matrix and Determinants** [8] 2.1 Introduction of a matrix 2.2 Types of matrices 2.3 Operations on matrices 2.4 Concept of a determinant 2.5 Evaluating value of a determinant 2.6 Properties of determinants 2.7 Project Work
• To use counting principles in solving related contextual problems. • To differentiate between permutation and combination with examples. • To apply different formulae associated with permutations in finding number of arrangements. • To apply formula of combinations in solving problems associated with selection. • To complete a project work assigned by teacher regarding application of permutation and combination in ICT.	**Unit III: Permutation and Combination [6]** 3.1 Basic principle of counting 3.2 Permutation 3.2.1 Permutations when all objects are distinct 3.2.2 Permutations when the objects are not all distinct 3.2.3 Circular permutations 3.3 Combinations 3.4 Project work
• To find Cartesian product of two sets. • To determine relation between two sets and find its domain, range, and inverse. • To describe equivalence relation with example. • To illustrate the concept of function through different approaches (as a relation, as a rule, as a machine) • To test the function for one-to-one and onto. • To construct examples of Polynomial, rational, exponential, and logarithmic function. • To find functional value, domain and range of a function (domain and range of function having finite domain only). • To describe and use properties of logarithm. • To represent functions (polynomial, exponential, and logarithmic) graphically.	**Unit IV: Functions** [10] 4.1 Cartesian product of sets 4.2 Relations 4.3 Concept of a Function 4.4 Domain and range of a function 4.5 Properties of functions (one to one and onto) 4.6 Types of functions (Polynomial, rational, exponential, and logarithmic function) 4.7 Properties of logarithm 4.8 Graph of a function 4.9 Project work

• To complete a project work assigned by teacher regarding application of functions in ICT.	
• To explain the meaning of limit of a function. • To use fundamental theorems on limits in evaluating limit of **polynomial, rational, exponential and logarithmic functions**. • To explain the concept of continuity of a function at a point. • To test whether a function is continuous at a particular point or not and supporting by reasons. • To identify graphs of continuous and discontinuous functions. • To complete a project work assigned by teacher regarding application of limit and continuity in ICT.	**Unit V: Limit and Continuity of Functions** [6] 5.1 Concept of a limit of a function 5.2 Evaluating limit of functions 5.3 Concept of continuity of function 5.4 Conditions for function to be continuous 5.5 Graphs of continuous and discontinuous functions 5.6 Project work
• To interpret concept of a derivative as rate of change. • To state rules (sum, product, quotient, chain) of differentiation and apply them in finding derivatives of algebraic, exponential, and logarithmic functions. • To perform implicit differentiation. • To describe some application of derivative in computer science. • To complete a project work assigned by teacher regarding application of derivative in ICT.	**Unit VI: Derivative** [5] 6.1 Concept of a derivative 6.2 Rules of differentiation 6.3 Finding derivative of functions 6.4 Implicit differentiation 6.5 Application of derivative in ICT 6.6 Project work
• To explain concept of anti-derivative. • To state rules and formulae for standard integrals and apply them in finding anti-derivative of algebraic functions. • To describe the meaning of definite integral and state rules of definite integrals. • To evaluate definite integral (algebraic problems only) • To explore application of anti-derivative in computer science. • To complete a project work assigned by teacher regarding application of anti-derivative in ICT.	**Unit VII: Anti- derivative** [5] 7.1 Concept of a anti derivative 7.2 Rules of integration 7.3 Finding anti-derivative 7.4 Definite integral 7.5 Evaluating definite integral 7.6 Application of anti-derivative in ICT 7.7 Project Work

4. Methodology and Techniques

- Since the course is to be studied by all students who has/has-not taken mathematics as a major subject at school level, teachers of this course are suggested to focus on conceptual understanding of basic concepts and developing different skills that are necessary for ICT students rather than considering comparatively difficult problems given in the recommended books.
- In each of the chapters, teachers are suggested to focus on skill development.

- Constructivist approach to develop conceptual understanding of concepts.
- Problem Based Learning to help students in solving problems in the exercises.
- Support students in their ZPD using constructivist perspective.
- **Exploration:** Help students to explore the essence of the concepts and formulae.
- Use collaborative learning methods together with expository-based demonstration methods as per the nature of the content.
- **Discussion:** discuss the application of the formulas and ask students to solve the problems applying formulae.
- Teachers may use mathematical software **Geogebra** for (function, limit, continuity, derivative, and antiderivative)
- In each unit assign the project work regarding application of the concerned topic in ICT.

5. Evaluation Scheme

5.1 **Internal Evaluation (40%)**

Internal Evaluation will be conducted by course teacher based on following activities.

a) **Attendance and Participation in class activities**: **5+5= 10 marks**

b) **Assignment I: Reflective Notes and Class presentation:** **5+5= 10 marks**
(Reflective notes on 2 to 4 questions given by teacher at the end of the every unit and presentation on any two questions among them)

c) **Assignment II**: **one Term paper/ Essay/Project and Interview:** **5+5=10 marks**
(Logical essay/term paper/project on the topics chosen by students and approved by the teacher and interview)

d) **Mid-term exam:** **10 marks**

Description of the Internal Evaluation

Mid-term exam: ***Engagement in a Class:*** Marks will be assigned based on the attendance and engagement in the classroom activities. At least 80% percent class attendance is mandatory for the students to enable them to appear in the End-Term examination. Below 80% in attendances that signify is NOT QUALIFIED (NQ) in subject to attend the end term examination.

Reflective Journal: It is individual work. Each student must submit their reflective journal of each chapter or teacher will give some questions that need reflective activities. The reflective journal will be returned to the students after its evaluation. Each student need to make presentation on their reflective journal.

Term paper*:* It is individual work. It must be prepared by the use of computer in a standard format of academic writing and must contain at least 5 pages. Quality, format, and time of submission will be the major criteria of the evaluation. Teacher will take interview of students based on their term paper.

Project Work: Students will be divided into groups. Each group will be assigned the project concerning application of concept and skills of selected topic of mathematics in ICT. Each group will present their findings in a whole class. Teacher may assign project work individually as well.

Mid-Term Examinations: It is a written examination and the questions will be set covering the topics as taught in the sessions. Mid-term examination will be based on the model prescribed for End-term examination.

5.2 External Evaluation (60%)

External Examinations: It is also a written examination and the questions will be asked covering all the topics in the session of the course. It carries 60 marks.

End Semester Examination Model

Nature of question	Total questions to be asked	Total questions to be answered	Total marks
Group A: Multiple choice	10 questions	10	10×1 = 10
Group B: Short answer type question	6 with 2 'or' questions	6	6×5 = 30
Group C: Long answer type question/case studies	2 with 1 'or' question	2	2×10 =20
Total			**60**

Recommended Books

Bajracharya, P. M., Basnet, G. B., Phulera, K. R. (2013). *Fundamentals of Mathematics*Buddha Publications. [for units 2,4,5,6, and 7]

Chand, H. B. & Chand, D. B. (2077). *Computational Literacy*. Intellectual's Book Palace. [for units 1 and 3]

References

Agrawal, R. S. (2010). *Senior secondary school mathematics*. Bharati Bhawan.

Dobbs, S. & Miller, J. (2008). *Advanced level mathematics: Statistics I*. Cambridge. Cambridge University Press.

Thomas, G.B. & Finney, R.L. (2001). *Calculus* (9th edition). Pearson Education.

Bachelor in Computer Science Education

Course Title: Computer Architecture and Organization	**Credits:** 3
Course No. : CS.Ed.124	**Teaching Hrs**: 48+16
Semester: 2nd	**Nature**: Theory and Practical
Level: Undergraduate	**No. of practical Periods**: 15 (2 hours per period)

1. Course Introduction

This course provides an introduction to computer organization and architecture, focusing on the design and implementation of computer systems. The course also introduces the basic knowledge about computer instructions set architecture, addressing modes, control unit of computer system, relationship between CPU, primary memory(RAM) and cache, the operations performed on computer registers and their interconnection with RAM and CPU.

2. Learning Objectives

At the end of this course the students should be able to:

- Understand the basic principles of computer organization and architecture
- Understand computer representation of data
- Understand the design and implementation of combinational and sequential circuit
- Describe different operations in terms of Micro-operations
- Develop skills in designing and implementing computer systems
- Develop an understanding of digital logic design and computer arithmetic
- Understand micro-programmed control unit
- Understand the design and implementation of processor architecture
- Learn about memory systems and input/output systems

3. Specific Objectives and Contents

Specific Objectives	**Contents**
• Illustrate the concept of Von Neuman and Harvard Architecture. • Describe the Flynn's classification • Memorize the knowledge of history of computer architecture.	**Unit 1: Introduction to Computer Architecture and Organization (2 hr)** 1.1 Digital computer 1.2 Computer Architecture 1.3 Computer Organization and Design 1.4 Von Neuman Architecture Vs Harvard Architecture 1.5 Flynns's Classification of Computer 1.6 Historical Perspective of Computer Architecture

• Interpret how numbers and text can be represented in computer system. • Organize concept of overflow and detection of overflow. • Demonstrate how errors can be detected using parity bits.	**Unit 2: Data Representation (6 hr)** 2.1. Introduction 2.2. Representation of numeric data (Review of number system: binary, octal and hex) 2.3. Complements:(r’s and r-1’: 1’s and 2’s) 2.4. Arithmetic’s using complements and detection of overflow 2.5. Integer Representation (positive and negative numbers) 2.6. Floating point representation 2.7. Representation of Textual data using different coding techniques (ASCII, Unicode, UTF-8) 2.8. Other codes(BCD, Gray, excess-three) and applications 2.9. Error Detection Codes: Parity Bit, Odd Parity, Even parity, Parity Generator & Checker
• Explorer the concept of Boolean Logic and algebra • Implement the operation of logic gates in real practical scenario • List the concept of Boolean algebra and laws of Boolean Algebra • Simulate the combinational and sequential circuits in practical environment	**Unit 3: Digital Logic Circuits and Boolean Algebra (8 hr)** 3.1. Introduction to Boolean Algebra 3.2. Basic laws of Boolean Algebra 3.3. Logic Gates (Basic, Advance and Universal gates) 3.4. Boolean Functions and its Simplifications (K-map method: 2, 3 and 4 variable map) 3.5. Combinational Circuits and its Design Procedures: half adder, full adder, half subtractor, full subtractor, decoder, encoder, code converter, multiplexer, demultiplexer. 3.6. Sequential Circuits: Flip flops(RS,JK,T,D), Flip flop excitation table 3.7. Design and Analysis of Sequential Circuit
• Memorize register transfer language • Implement arithmetic, logic and shift operations in terms of microperations. • Design circuit diagrams of arithmetic, logic and shift operations.	**Unit 4: Register Transfer and Micro operations(6 hr)** 4.1. Introduction to Micro operations and Register Transfer 4.2. Arithmetic Micro operation: Binary adder, Binary adder-subtractor, Binary incrementer, Arithmetic circuit 4.3. Logic Micro operations: Hardware Implementation and Applications 4.4. Shift Micro operations: Logical shift, Circular shift, arithmetic shift and its Hardware Implementation
• Illustrate computer organization and architecture using hypothetical computer system. • Explain Common bus system of basic computer. • Interpret instruction set of basic	**Unit 5: Basic Computer Organization and Design (7 hr)** 5.1 Introduction to Computer Instruction: Instruction code, operation code and addressing mode (direct and indirect) 5.2 Stored Program Concept 5.3 Computer Registers and Common Bus System 5.4 Instruction Format and types (Memory, Register, I/O

computer • Interpret interrupt cycle of basic computer Memorize overall execution cycle of basic computer	reference instructions) instruction. 5.5 Instruction Cycle of Basic computer 5.6 Program Interrupt & Interrupt Cycle 5.7 Description and Flowchart of Basic Computer 5.8 Control Unit (Micro programmed and Hardwired Control Unit) 5.9 Design of Control Unit
• List different CPU organizations • Implement types of instructions on the basic of number of operands • Interpret operand using addressing modes. • Describe the basics of data manipulation • Compare and Contrast RISC and CISC computer architectures	**Unit 6: Central Processing Unit (5 hr)** 6.1 Introduction to CPU: Components of CPU 6.2 CPU organization: Single Accumulator Organization, General Register Organization, Stack Organization 6.3 Instruction Formats based on Address or Operands: Three address, Two address, One address and Zero address 6.4 Addressing Modes 6.5 Data Transfer and Manipulation: Data Transfer instruction, Data Manipulation and Program Control Instruction 6.6 RISC and CISC Computer Architecture
• Demonstrate addition, subtraction and multiplication algorithm for signed magnitude data • Demonstrate addition, subtraction and multiplication algorithm for signed 2's complement data • Recall hardware implementation of all described algorithms	**Unit 7: Computer Arithmetic(3 hr)** 7.1 Addition and Subtraction with Signed Magnitude Data (Hardware Implementation and Algorithm), Addition and Subtraction with Signed 2's Complement Data 7.2 Multiplication of Signed Magnitude Data (Hardware Implementation and Algorithm), Booth Multiplication (Hardware Implementation and Algorithm)
• List interface between I/O devices and CPU. • Compare strobe and handshaking mechanism of data transfer • Describe modes of data transfer along with their pros and cons • Differentiate DMA from input-output processors	**Unit 8: Input Output Organization (4 hr)** **8.1** Input-Output Interface: I/O Bus and Interface Modules, I/O vs Memory Bus, Isolated vs Memory-Mapped I/O 8.2Asynchronous Data Transfer: Strobe, Handshaking (Source and Destination Initiated) 8.3. Modes of Transfer: Programmed I/O, Interrupt-Initiated I/O, Direct memory Access DMA (Bus Arbitration, DMA Controller, DMA Transfer) 8.4 Input Output Processor(IOP)
• Understand why a memory hierarchy is necessary to reduce the effective memory latency. • Describe techniques of mapping data stored in RAM to the data in cache memory	**Unit 9: Memory Organization(4 hr)** 9.1 Memory Hierarchy: Main Memory (RAM and ROM Chip Memory address Map, Memory Connection to CPU), Auxiliary Memory (magnetic Disk, Magnetic Tape, SSD) 9.2 Associative Memory: Hardware Organization, Match Logic,

	Read Operation, Write Operation 9.3 Cache Memory: Locality of Reference, Hit & Miss Ratio, Mapping (Direct, Associative, Set Associative), Write Policies (Write-Back, Write-Through)

4. Methodology and Techniques

Modes of instruction*:* Lecture, seminar, exercise course, guided personal study, tutorial, independent study, project work, Assignments indifferent topics, group discussion, reflective writing

Types of learning activities*:* attending lectures, performing specific assignments, writing papers, independent and private study, reading books, journals and papers, providing constructive feedback, group study and peer discussion.

5. Evaluation Scheme

5.1 Internal Evaluation 40%

Internal Evaluation will be conducted by course teacher based on following activities.

a) **Attendance and Participation in class activities**: **5+5=10marks**

b) **Assignment I: Reflective Notes and Class presentation:** **5+5=10marks**

(Reflective notes on 2 to 4 questions given by teacher at the end Of the every unit and presentation on any two questions among them)

c) **Assignment II**: **OneTerm paper/Essay/Project and Interview:** **5+5=10marks**

(Logical essay/term paper/project on the topics chosen by students and approved by the teacher and interview)

d) **Mid-term exam**: **10marks**

5.2 External Evaluation (Final Examination) 40%

Types of questions	**Total questions to be asked**	**Number of questions to be answered and marks allocated**	**Total marks**
Group A: Multiple choice items	8 questions	8×1	8
Group B: Short answer questions	6 with 2 'or' questions	6×4	24
Group C: Long answer questions	1 with 1 'or' question	1×8	8

5.3 External Practical Evaluation (20%)

Office of the Controller of Examination will conduct final practical examination at the end of final examination.

After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the following evaluation criteria. There will be an internal examiner to assist the external examiner. Three hours' time will be given forth e practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Evaluation System:

Practical	Weightage	Marks
Practical Report Copy	5	20
Viva	5	
Practical Exam	10	

Laboratory Work

Student should write a program for simulating the behavior of basic logic gates and different combinational and sequential circuits using VHDL simulators like Synopsys VCS, Mentor Graphics, Model Sim, Aldec Active-HDL, Xilinx Vivado, GHDL etc. The type of practical problems that must be conducted in lab should be as given:

- ✓ Write a program in VHDL language that implements basic logic gates
- ✓ Write a program in VHDL language that implements advance logic gates
- ✓ Write a program in VHDL language that implements behavior of combinational circuits like half adder, full adder, half subtractor, full subtractor, decoder, encoder, MUX, and DMUX etc
- ✓ Write a program in VHDL language that implements behavior of sequential circuits like RS, JK, T and D flip flops etc

For conducting practical instructor can instruct different types of real life problems relevant to the subject matter.

Text books

Mano, M. M. (2007). *Computer system architecture*. Prentice-Hall, Inc.

Reference books

Stallings, W. (2003). *Computer organization and architecture: designing for performance*. Pearson Education India.

Heuring, V. P., Jordan, H. F., &Murdocca, M. (1997). *Computer systems design and architecture*. Addison-Wesley.

Bachelor in Computer Science Education

Course Title: Data Structure and Algorithm **Credits:** 3
Course No. : CS.Ed.125 **Teaching Hrs**: 48+16
Semester: 2nd **Nature**: Theory and Practical
Level: Undergraduate **No. of practical Periods**: 15 (2 hours per period)

1. Course Introduction

The main objective of this course is to provide the basic knowledge of data structures and algorithms and how they are used to solve the problems in particular fields. Further, course also introduces the basic knowledge of algorithm analysis and design and its role in computation. The course covers the concept of algorithms, some basic and advances data structures and their implementation, concept of searching and hashing, sorting and classification of algorithms.

2. Objectives

At the end of this course the student should able to:

- ✓ Learn about the data structure and algorithm
- ✓ Know about the analysis of algorithm
- ✓ Understand the relationship between data structure and algorithms
- ✓ Implement the data structures like the stack, queue, list, graph and tree etc
- ✓ Implement the concept of searching and hashing
- ✓ Implement the concept of sorting
- ✓ Classify the algorithms
- ✓ Design the algorithms for solving problems

3. Specific Objectives and Contents

Specific Objectives	**Contents**
• List the concept of data structure and algorithm • Describe the relationships between data structure and algorithm • Illustrate the analysis of an algorithms in terms of complexity (space and time) • Implement the stack operations • Implement the evaluations of expressions	**Unit 1: Data Structure and Algorithm (7 hr)** 1.1. Data Structure and its Classification 1.2. Algorithm and its Properties 1.3. Relationship between Data Structure and Algorithm 1.4. Analysis of an Algorithm: Asymptotic Notations 1.5. Rate of Growth 1.6. Concept of Abstract data type(ADT) **1.7. Stack** 1.8. Operations on Stack: push() and pop() 1.9. Algorithms for push() and pop() operations 1.10. Stack as an ADT

	1.11. Stack application: Evaluation of Infix, Postfix and prefix expressions **Lab Work:** Write a Program to Implement Stack operations. Write a Program to Implement Evaluations of expressions
• Define QUEUE and queue as ADT • Differentiate the types of queues • Write and execute the codes for queue operation on different types of queue • Identify and debug the errors • List the real applications of different types of queues	**Unit 2: QUEUES (4 hr)** 2.10. Introduction to Queue: Linear and Circular 2.11. Basic Operations on Queue: enqueue and dequeue 2.12. Algorithm of enqueue and dequeue 2.13. Queue as ADT 2.14. Double ended Queue 2.15. Priority queue 2.16. Applications of queue **Lab Work:** Write a Program that Implements Queue data structure
• Define the linked list • Differentiate between array and linked list • Differentiate the types of linked list and their nature • Implement the operations of linked list • Implement sparse matrix • List out the applications of linked list	**Unit 3: Linked List (7 hr)** 3.8. Introduction to Linked list 3.9. Array vs Linked List 3.10. Types of linked List: Single and double Linked List 3.11. Operations on Linked List: Creations of node, Insertion of node at the beginning and end of the list, deletions of node from beginning and end, insertion and deletion from specific position of list (Single and Double) 3.12. Algorithms for each operations of Linked List 3.13. Concept of Circular Linked List 3.14. Sparse Matrix and its use **Lab Work:** Write a Program that Implements single and double Linked List operations
• Memorize the concept of recursion • List out the need of recursion • Write and execute the codes for recursive program • Identify and debug the errors	**UNIT 4: Recursion (3 hr)** 4.5. Principle of recursion, Advantages and disadvantages of recursion. 4.6. Implementation recursion on: Factorial, GCD, TOH and Fibonacci sequence 4.7. Comparison between recursion and iteration, recursion example 4.8. Applications of recursion **Lab work:** Write a Program that Implements Recursive functions

• Explorer the uses of tree in problem solving • Implements various types of trees • List out the concept of Huffman Algorithm and its applications	**UNIT 5: Trees (7 hr)** 5.1. Introduction to Tree: Definition, Terminologies, Properties and Types 5.2. Binary Tree: Definitions, Properties, Types, traversal (Pre-order, In-order and Post-order) and Representation. 5.3. Binary Search Tree: Definitions and Operations (Searching, Insertion, deletion) 5.4. Balanced trees: AVL balanced Tree, Balancing algorithm 5.5. B Tree and its Operations: Searching, Insertion and Deletion 5.6. The Huffman Algorithm and its Applications **Lab Work:** Write a Program that implements Binary tree Write a Program that implements Binary Search Tree Write a Program that implements B tree
• Define terminologies used in graph and able to explorer the uses of graph in life problems. • Implement the types of graph • Implement the Graphs • List the applications of shortest path and MST algorithms • To explore the uses of transitive closure	**UNIT 6: Graphs (5 hr)** 6.1. Graph: Definition, Terminologies, Types, and Applications 6.2. Representation of Graph: Adjacency Matrix, Adjacency List, Edge List 6.3. Graph Traversal (BFS and DFS) 6.4. Shortest Path Algorithms: Dijkstra's Algorithm 6.5. Minimum Spanning Tree Algorithms: Kruskal's and Prims Algorithms 6.6. Transitive Closure **Lab Work**: Write a program that implements Graph and its representation
• Compare and select the best searching techniques • Implement the searching algorithms • Implement the concept of hashing and collision resolution techniques	**UNIT 7: Searching and Hashing (4 hr)** 7.1. Searching 7.2. Linear Vs Binary Search 7.3. Hashing: Hash function and hash table 7.4. Hash Collision 7.5. Collision Resolution Techniques: Open Addressing (Linear Probing, Quadratic Probing, Double Hashing) and Chaining **Lab Work:** Write a program that implements searching and hashing Write a program that implements concept of hash collision resolution

• Memorize the applications of sorting • Write and execute the codes for different types of sorting • identify, detect, and debug the errors • Interpret the time, space and performance complexity of different types of sorting	**UNIT 8: Sorting (4 hr)** 8.1. Introduction and Application of Sorting 8.2. Types of sorting**:** Bubble sort, Insertion sort, Selection sort, Quick sort, Radix sort, Merge sort, Heap sort and Shell sort. **Lab Work:** Write a program that implements sorting algorithm
• Classify the algorithms on different basis • List the applications of different algorithms in different fields	**Unit 9: Classification of an Algorithms (3hr)** 9.1. Introduction 9.2. Types of Algorithms: Deterministic and non-deterministic algorithm, Divide and conquer Algorithm Serial and parallel algorithm, Heuristic and Approximate algorithms

4. Methodology and Techniques

Modes of instruction*:* Lecture, seminar, exercise course, guided personal study, tutorial, independent study, project work, Assignments in different topics, group discussion, reflectivewriting

Types of learning activities*:* attending lectures, performing specific assignments, writing papers,independent and private study, reading books, journals and papers, providing constructive feedback, group study and peer discussion.

5. Evaluation Scheme

5.2 Internal Evaluation 40%

Internal Evaluation will be conducted by course teacher based on following activities.

e) **Attendance and Participation in class activities**: **5+5= 10 marks**

f) **Assignment I: Reflective Notes and Class presentation:** **5+5= 10 marks**

(Reflective notes on 2 to 4 questions given by teacher at the end of the every unit and presentation on any two questions among them)

g) **Assignment II**: **One Term paper/ Essay/Project and Interview:** **5+5=10 marks**

(Logical essay/term paper/project on the topics chosen by studentsand approved by the teacher and interview)

h) **Mid-term exam**: **10 marks**

5.2 External Evaluation (Final Examination) 40%

Types of questions	**Total questions to be asked**	**Number of questions to be answered and marks allocated**	**Total marks**
Group A: Multiple choice items	8 questions	8 × 1	8
Group B: Short answer questions	6 with 2 'or' questions	6 × 4	24
Group C: Long answer questions	1 with 1 'or' question	1× 8	8

5.3 External Practical Evaluation (20%)

Office of the Controller of Examination will conduct final practical examination at the end of finalexamination.

After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the following evaluation criteria. There will be an internal examiner to assist the external examiner. Three hours' time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Evaluation System:

Practical	Weightage	Marks
Practical Report Copy	5	20
Viva	5	
Practical Exam	10	

Text book

Kanetkar, Y. (2019). *Data Structures Through C: Learn the fundamentals of Data Structures through C*. BPB publications.(All chapters)

Reference book

Bhatt, H., & Bhatt, B.P. (2078). Data Structure and Algorithm. Dreamland Publications.(All Chapters)