ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(AUTONOMOUS)

ACCREDITED BY NBA & NAAC

Affiliated to Andhra University

PRAGNANAM BRAHMA

ANITS

# DEPARTMENT OF INFORMATION TECHNOLOGY

## Academic Regulations

## Course Structure & Detailed Syllabus (R-23)

## II YEAR

## Applicable for the batch admitted in 2023-24

## IT - COURSE STRUCTURE R23

### SECOND YEAR SEMESTER – I

| Code | Course | Category | L | T | P | S | Total | Sessional Marks | External Marks | Total Marks | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 23MA1103 | Discrete Mathematical Structures | BS | 2 | 1 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23IT4113 | Software Engineering | PC | 3 | 0 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23EC3104 | Computer Organisation | ES | 3 | 0 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23IT4114 | Advanced Data Structures | PC | 3 | 0 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23IT4115 | Object Oriented Programming through Java | PC | 3 | 0 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23EC3211 | Computer Organisation Lab | ES | 0 | 0 | 3 | 0 | 3 | 50 | 50 | 100 | 1.5 |
| 23IT4213 | Advanced Data Structures Lab | PC | 0 | 0 | 3 | 0 | 3 | 50 | 50 | 100 | 1.5 |
| 23IT4214 | Java Lab | PC | 0 | 0 | 3 | 0 | 3 | 50 | 50 | 100 | 1.5 |
| 23CR9101 | Logical Reasoning & Corporate Skills | HS | 2 | 0 | 0 | 0 | 2 | 50 | 50 | 100 | 1 |
| 23IT9301 | Network Fundamentals | SC | 0 | 0 | 0 | 2 | 2 |  |  |  | 1 |
| 23MC0103 | Financial Literacy | MC | 3 | 0 | 0 | 0 | 3 | 50 | 0 | 50 | - |
| **TOTAL** |  |  | **17** | **1** | **11** |  | **31** | **450** | **500** | **950** | **21.5** |

### SECOND YEAR SEMESTER – II

| Code | Course | Category | L | T | P | S | Total | Sessional Marks | External Marks | Total Marks | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 23MA1104 | Probability & Statistics | BS | 2 | 1 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23IT4116 | Database Management Systems | PC | 3 | 0 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23IT4117 | Computer Networks | PC | 3 | 0 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23IT4118 | Python Programming for Data Science | ES | 1 | 0 | 4 | 0 | 5 | 40 | 60 | 100 | 3 |
| 23IT4119 | Operating systems | PC | 3 | 0 | 0 | 0 | 3 | 40 | 60 | 100 | 3 |
| 23IT4215 | DBMS Lab | PC | 0 | 0 | 3 | 0 | 3 | 50 | 50 | 100 | 1.5 |
| 23IT4216 | Computer Networks Lab | PC | 0 | 0 | 3 | 0 | 3 | 50 | 50 | 100 | 1.5 |
| 23IT4217 | Operating Systems lab | PC | 0 | 0 | 3 | 0 | 3 | 50 | 50 | 100 | 1.5 |
| 23CR9102 | Numerical Ability & Professional Communication Skills | HS | 0 | 0 | 2 | 0 | 2 | 50 | 50 | 100 | 1 |
| 23IT9302 | Angular Developer | SC | 0 | 0 | 0 | 2 | 2 |  |  |  | 1 |
| 23MC0104 | Entrepreneurship Development & IPR | MC | 3 | 0 | 0 | 0 | 3 | 50 | 0 | 50 | - |
| **TOTAL** |  |  | **15** | **1** | **15** |  | **33** | **450** | **500** | **950** | **21.5** |

# SEMESETER-1

# DISCRETE MATHEMATICAL STRUCTURES

COURSE CODE: 23MA1103 CREDITS 3

L T P S Sessional Marks: 40

3 0 0 0 End Exam: 3 Hours

End Exam Marks: 60

Prerequisites: Elementary knowledge of set theory, Matrices and functions.

Course Objectives:

This course will discuss fundamental concepts and tools in discrete mathematics with emphasis on their applications to computer science. Topics include logic, functions, relations, recurrence relations, fundamental concepts of number theory and graph theory.

Course Outcomes:

After course completion, the students will be able to:

1. Identify logical skills in solving mathematical problems.
2. Determine properties of binary relations, identify equivalence and partial order relations, and sketch relations.
3. Analyse recurrence relations, generating functions and solving problems involving recurrence relations.
4. Evaluate the concepts related divisibility, congruencies and number theoretic functions and identify the structure of group, ring, and field.
5. Explain the basic concepts of graph theory and develop a graph theoretical model for a real time situations

**Mapping of Course Outcomes with POs and PSOs**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | | | | | | | | | | 1 | | |
| **CO2** | 3 | 2 | | | | | | | | | | 1 | | |
| **CO3** | 3 | 2 | | | | | | | | | | 1 | | |
| **CO4** | 3 | 2 | | | | | | | | | | 1 | | |
| **CO5** | 3 | 2 | | | | | | | | | | 1 | | |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT I 10 Periods

MATHEMATICAL LOGIC: Fundamentals of logic – Logical inferences – Methods of proof of implication – First order logic and other proof methods – Rules of inference for quantified propositions – Pigeonhole principle – Mathematical induction.

UNIT II 10 Periods

RELATIONS AND ALGEBRAIC SYSTEMS: Cartesian products of sets – Relations – Properties of binary relations in a set – Relation matrix and graph of a relation – Partition and covering of set – Equivalence relations – Composition of binary relations – Transitive closure of a relation – Partial ordering – Partially ordered set – Hasse diagram – Lattice.

UNIT III 10 Periods

RECURRENCE RELATIONS: Generating functions of sequences – Calculating their coefficients – Recurrence relations –Solving recurrence relations – Method of characteristic roots – Non-homogeneous recurrence relations and their solutions.

UNIT IV 10 Periods

NUMBER THEORY: Divisibility and Modular Arithmetic – Integer representations and algorithms – Primes and greatest common divisors – Solving congruences.

ALGEBRAIC STRUCTURES: Semi Groups – Monoids – Groups – Subgroups and their properties – Introduction to rings and fields. (Only definitions and examples)

UNIT V 10 Periods

GRAPHS: Introduction to graphs – Types of graphs – Graphs basic terminology and special types of simple graphs – Representation of graphs and graph isomorphism – Euler paths and circuits – Hamilton paths and circuits – Planar graphs – Dual of a graph – Euler's formula – Graph coloring – Chromatic number.

TEXT BOOKS:

1. Joe L. Mott, Abraham Kandel & T. P. Baker, Discrete Mathematics for computer scientists & Mathematicians, Prentice Hall of India Ltd, New Delhi., 2008.
2. Keneth. H. Rosen, Discrete Mathematics and its Applications, 7/e, Tata McGraw-Hill, 2015.
3. J. P. Tremblay, R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill Publishing Company Limited,1997.

REFERENCE BOOKS:

1. Richard Johnsonburg, Discrete mathematics, 7/e, Pearson Education, 2008.
2. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall of India, 2006.

# SOFTWARE ENGINEERING

COURSE CODE: 23IT4113 CREDITS 3

L T P S Sessional Marks: 40

3 0 0 0 End Exam: 3 Hours

End Exam Marks: 60

Prerequisites: Computer Fundamentals, Any Programming Language

Course Objectives:

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and Agile Development.

Course Outcomes:

After course completion, the students will be able to:

1. Translate end-user requirements into system and software requirements, using UML, and structure the requirements into a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Develop a simple testing report, and to manage time, processes and resources effectively by prioritising competing demands toachieve personal and team goals
4. Demonstrate Agile Development & Testing Techniques

**Mapping of Course Outcomes with POs and PSOs**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 3 | 3 | 1 | 1 | 1 | | | | 2 | 2 | 3 | 3 |
| **CO2** | 2 | 2 | 2 | 2 | 1 | | | | | 2 | 2 | 2 | 3 | 3 |
| **CO3** | 2 | 2 | 3 | 3 | 3 | | | | | | 1 | 1 | 3 | 3 |
| **CO4** | 2 | 2 | 3 | 3 | 3 | | | | | | 1 | 1 | 3 | 3 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT – I 10 Periods

**Introduction to Software Engineering:** The evolving role of software, change in nature of software, software myths

**Software Requirements:** Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

**Process Models:** Waterfall model, Incremental model, Evolutionary models, Unified models

Learning outcomes:

1. Process of analysing user requirements

2. designing software application which will satisfy that requirements

UNIT – II 10 Periods

**Requirements engineering process**: Feasibility studies, requirements elicitation and analysis,requirements validation, requirements management.

**System models**: Context models, behavioral models, object models, structured methods.

UNIT – III 10 Periods

**Design Engineering**: Design process and design quality, design concepts, the design model. **Creating an Architectural Design:** Software architecture, data design, architectural styles andpatterns, architectural design.

**Testing Strategies**: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging

Learning outcomes:

1. To produce efficient, reliable, robust and cost-effective software solutions.
2. Ability to perform independent research and analysis.

UNIT – IV 9 Periods

**Metrics for Process and Products:** Software measurement, metrics for software quality, Metrics for Process and Product

**Quality Management:** Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO standards

Learning outcomes:

1. check whether the actual software product matches expected requirements
2. making it efficient and effective as per the quality standards defined for software products
3. Ability to understand and meet ethical standards and legal responsibilities
4. Ability to develop, maintain and evaluate large-scale software systems

UNIT-V 9 Periods

**Agile Methodology:** Theories for Agile management – agile software development – traditional model vs. agile model - classification of agile methods – agile manifesto and principles – agile project management – agile team interactions – ethics in agile teams - agility in design, testing – agile documentations – agile drivers, capabilities and values.

Learning outcomes:

1. Importance of interacting with business stakeholders in determining the requirements for a software system

TEXT BOOKS:

1. Software Engineering, A practitioner’s Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.
2. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.
4. Hazza& Dubinsky, ―Agile Software Engineering, Series: Undergraduate Topics in Computer Science‖, Springer, VIII edition, 2009
5. Craig Larman, ―Agile and Iterative Development: A manager‗s Guide, Addison-Wesley, 2004
6. Software Engineering- Sommerville, 7th edition, Pearson Education.
7. Dingsoyr, Torgeir, Dyba, Tore, Moe, Nils Brede (Eds.), ―Agile Software Development, Current Research and Future Directions‖, Springer-Verlag Berlin Heidelberg, 2010
8. David J. Anderson; Eli Schragenheim, ―Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results‖, Prentice Hall, 2003

CHANGE OF SYLLABUS

| **Unit No** | **Changes Incorporated** |
|---|---|
| Unit – 5 | **Agile Methodology** |
| **Change of Syllabus: 20%** | |

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Software Engineering | 23IT4113 | R23 (2024) | Agile Methodology<br>Product metrics | Employability |

# COMPUTER ORGANIZATION

| | |
|---|---|
| COURSE CODE: 23EC3104 | CREDITS 3 |
| L T P S | Sessional Marks: 40 |
| 3 0 0 0 | End Exam: 3 Hours |
| | End Exam Marks: 60 |

Prerequisite(s):

- Digital Electronics

Course Objective:

1. To understand the fundamentals of different instruction set architectures and their relationship to the CPU design.
2. To study the control units of basic computer and their instruction formats.
3. To study the hardware involved in the CPU of a computer
4. To study the hierarchical memory system including cache memories used in basic computer.
5. To study the different ways of communicating with I/O devices and standard I/O interfaces.

Course Outcomes:

After course completion, the students will be able to:

1. Work with the typical assembly language instructions of a computer
2. Design control unit of a basic computer
3. Organize the hardware involved in the CPU of a computer
4. Use computing resources such as memory devices in an effective manner to improve the performance of a Computer.
5. Categorize input-output devices of a computer and work with standard I/O interfaces.

**Mapping of Course Outcomes with POs and PSOs**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 1 | | 2 | | | | | | | | | |
| **CO2** | 3 | 2 | 3 | 1 | | | | | | | | | | |
| **CO3** | 3 | 2 | 3 | | 1 | | | | | | | | | |
| **CO4** | 3 | 2 | 2 | | 3 | | 1 | | | | | | | |
| **CO5** | 3 | 2 | 1 | | 2 | | | | | | | | | |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT – I 10 Periods

**REGISTER TRANSFER AND MICROOPERATIONS**: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Microoperations, Logic Micro operations, Shift Micro operations, Arithmetic Logic Shift Unit

UNIT – II 12 Periods

**BASIC COMPUTER ORGANIZATION:** Instruction Codes, Computer Registers, Computer Instructions, hardwired control unit, Instruction Cycle, Memory Reference Instructions

**MICROPROGRAMMED CONTROL:** Control Memory, Address Sequencing, Microinstruction Formats, Micro program Example, Design of Control Unit

UNIT – III 10 Periods

**CPU ORGANIZATION:** Introduction, General Register Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Stack Organization. Reduced Instruction Set Computer (RISC) and CISC architectures

UNIT – IV 09 Periods

**MEMORY ORGANIZATION:** Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory

UNIT – V 09 Periods

INPUT - OUTPUT ORGANIZATION: Peripheral Devices, Input - Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA).

**TEXT BOOKS**

1. M. Morris Mano, “*Computer System Architecture”*, 3rd Ed., PHI, 1996

**REFERENCE BOOKS:**

1. V. Carl Hamacher, Zvonko G. Vranesic and Safwat G. Zaky, “Computer Organization”, 5th Ed., McGraw Hill International, 2011
2. Sivarama P. Dandamudi, “Fundamentals of computer Organization and design”, Springer, 2002
3. William Stallings, “Computer Organization & Architecture - Designing for performance”, 8th Ed., Pearson Education India, 2013
4. John D. Carpinelli, “Computer Systems Organization & Architecture”, 1st Ed., Pearson Education India, 2000
5. Sajjan G. Shiva, “Computer design and architecture”, 3rd Ed., Marcel Dekker, 2000
6. Hennessy- Patterson, “Computer Architecture: A quantitative approach”, 5 th edition, Morgan Kaufmann, 2011

# ADVANCED DATA STRUCTRES

| | |
|---|---|
| COURSE CODE: 23IT4114 | Credits: 3 |
| L T P S | Sessional Marks: 40 |
| 3 0 0 0 | End Exam Marks: 60 |

Prerequisite(s):

- Overview on Programming languages
- Fundamental Data Structures

Course Objective:

- To explore the advanced concepts of data structure and algorithms and its implementation.

Course Outcomes:

After course completion, the students will be able to:

1. Develop and implement advanced data structures such as dictionaries, skip lists, AVL trees, Red-Black trees, Splay trees, and B-trees, and analyze their performance and use cases. (L3)
2. Construct and manipulate priority queues using binary heaps and binomial queues, understanding their operations, amortized analysis, and applications in various computing scenarios. (L3)
3. Apply fundamental graph algorithms, including depth-first and breadth-first traversals, minimum spanning tree algorithms (Prim's and Kruskal's), shortest-path algorithms (Dijkstra's and Floyd's), and topological sorting. (L3)
4. Implement and analyze string processing algorithms and data structures, such as brute-force pattern matching, Boyer-Moore, Knuth-Morris-Pratt, tries, suffix tries, Huffman coding, and algorithms for finding the longest common subsequence (LCS). (L3)

**Mapping of Course Outcomes with POs and PSOs**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 3 | 2 | 3 | 1 | 1 | 1 | 3 | 2 | 2 | 2 | 1 | 3 |
| **CO2** | 3 | 3 | 3 | 2 | 3 | 1 | 1 | 1 | 3 | 2 | 2 | 2 | 1 | 3 |
| **CO3** | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 1 | 3 | 2 | 2 | 2 | 1 | 3 |
| **CO4** | 3 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 3 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Unit-1 10 Periods

**Dictionaries:** Definition, Dictionary Abstract Data Type, Implementation of Dictionaries.

**Skip Lists:** Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists

**Learning Outcome**: At the end of the unit, the student will be able to

- Apply hashing
- Understand the need for randomizing data structures

Unit-2 10 Periods

**Advanced Trees:** AVL Trees, Red-Black Trees, Splay Trees, B-Trees

**Learning Outcome**: At the end of the unit, the student will be able to

- Explore the need for advanced tree concepts

Unit-3 10 Periods

**Priority Queues:** Binary Heaps: Implementation of Insert and Delete min, Creating Heap.

**Binomial Queues:** Binomial Queue Operations, Binomial Amortized Analysis, Lazy Binomial Queues.

**Learning Outcome**: At the end of the unit, the student will be able to

- Implement binary heaps
- Understand binomial queues

Unit-4 10 Periods

**Graphs & Algorithms:** Representation, Type of Graphs, Depth- and breadth-first traversals, Minimum Spanning Tree: Prim's and Kruskal's algorithms, Shortest-path Algorithms: Dijkstra's and Floyd's algorithm, Topological sort

**Learning Outcome**: At the end of the unit, the student will be able to

- Understand the need for graphs
- Explore different graph algorithms

Unit-5 10 Periods

**Text Processing:** String Operations, Brute-Force Pattern Matching, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS)

**Learning Outcome**: At the end of the unit, the student will be able to

- Understand the basic pattern matching techniques
- Understand Huffman coding and tries

**TEXT BOOKS**

2. Advanced Data Structures by Suman Saha, Shailendra Shukla CRC Press, Taylor & Francis Group.
3. Advanced Data Structures by Peter Brass Cambridge University Press.

**REFERENCE BOOKS:**

1. C and Data Structures: A Snap Shot Oriented Treatise with Live examples from Science and Engineering, NB Venkateswarlu & EV Prasad, S Chand, 2010.
2. Data Structures, A Pseudocode Approach, Richard F Gilberg, Behrouz A Forouzan, Cengage.
3. Data structures and Algorithm Analysis in C, 2nd edition, Mark Allen Weiss, Pearson

**WEB LINKS**

1. http://lcm.csa.iisc.ernet.in/dsa/dsa.html

2. http://utubersity.com/?page_id=878
3. http://freevideolectures.com/Course/2519/C-Programming-and-Data-Structures
4. http://freevideolectures.com/Course/2279/Data-Structures-And-Algorithms

CHANGE OF SYLLABUS

**New Course**
**Change of Syllabus: 100%**

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Advanced Data Structures | 23IT4119 | R23 (2024) | AVL Trees<br>B-Trees<br>Graphs<br>Text Processing | Employability |

# OBJECT ORIENTED PROGRAMMINGTHROUGH JAVA

COURSE CODE: 23IT4115 CREDITS 3

L T P S Sessional Marks: 40

3 0 0 0 End Exam Marks:60

End Exam:3 Hours

Prerequisite(s): Basic knowledge on introduction to programming & object oriented programmingconcepts

Course Objectives:

- To understand object oriented programming concepts, and apply them in problemsolving.
- To introduce the concepts of exception handling and multithreading.
- To introduce the design of Graphical User Interface using applets and swing controls.

Course Outcomes:

After course completion, the students will be able to:

1. Apply Object-oriented programming principles to write efficient and reusable codes.
2. Use of exception handling and Multithreading concepts to improve the robustness, Multithreaded and reliability of Java applications
3. Demonstrate the concepts of Java Files, collections and Java database connectivity in real-time problem solving.
4. Develop GUI applications using AWT, Swings and JavaFX.

**Mapping of course outcomes with program outcomes:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 3 | | | | | | | | | 3 | 3 | 2 |
| CO2 | 2 | 2 | 3 | | 2 | | 2 | 2 | 2 | | | 2 | 3 | 2 |
| CO3 | 2 | 2 | 3 | | | | | | | | 2 | 2 | 3 | 2 |
| CO4 | 2 | 2 | 3 | | 2 | | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT-I 12 Periods

**Classes and objects:** creating classes and objects, accessing methods, object initialization, java garbage collector.

**Inheritance -** Inheritance hierarchies, super and sub classes, Member access rules, super keyword, preventing inheritance: final classes and methods, the Object class and its methods. **Polymorphism -** dynamic binding, method overriding, abstract classes and methods.

**Interfaces -** Interfaces vs. Abstract classes, defining an interface, implementing interfaces, accessing implementations through interface references, extending interfaces.

**Inner classes -** uses of inner classes, local inner classes, anonymous inner classes, static inner classes, examples.

**Packages -** Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

**Learning Outcomes:**

At the end of this Unit the student will be able to

- Define Objects and Classes and methods
- Describe Inheritance and method overriding
- Explain inheritance on interfaces, implementing interface, multiple inheritance using interface
- Create and understand package, importing package, access rules for packages, class hiding rules in a package

UNIT-II 10 Periods

**Exception handling -** Dealing with errors, benefits of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, re throwing exceptions, exception specification, built in exceptions, creating own exception sub classes.

**Multithreading -** Difference between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication.

**Learning Outcomes:**

At the end of this Unit the student will be able to

- Explain errors & exceptions
- Define & create threads, and can implement multithreading, thread priority & synchronization

UNIT-III 12 Periods

**Collection Framework in Java -** Introduction to Java Collections, Overview of Java Collection frame work, commonly used Collection classes Array List, Vector, Hash table, Stack, Introduction to concurrent classes

Utilities - Enumerator, Iterator, String Tokenizer, Scanner

**Files – I/O** streams - byte streams, character streams, text input/output, binary input/output, random access file operations, File management using File class.

.**Learning Outcomes:**

At the end of this Unit the student will be able to

- Illustrate appropriate use of collections in solving real world problems.
- Explain basics of streams, stream classes, creation, reading and writing files in context to file handling

UNIT-IV 10 Periods

**GUI Programming with Java -** The AWT class hierarchy, Introduction to Swing, Swing vs AWT, Hierarchy for Swing components, simple swing applications, Layout management - Layout manager types - border, grid and flow layouts. **Event handling -** Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Examples: handling a button click, handling mouse events, Adapter classes.

**Learning Outcomes:**

At the end of this Unit the student will be able to

- Create GUI applications using AWT and Swing Components.
- Differentiate the AWT and Swing components.
- Develop GUI applications using event handlers.

UNIT – V 8 Periods

**Connecting to Database -** Introduction to major JDBC Classes and Interfaces, JDBC drivers, steps to connecting to a database, querying a database and processing the results, updating data with JDBC, Types of Statements (Statement Interface, Prepared Statement, Callable Statement)

**Learning Outcomes:**

At the end of this Unit the student will be able to

- Develop applications to interact with Databases.
- Distinguish various drivers available in JDBC
- Distinguish various statements available in JDBC

**TEXT BOOKS:**

1. Java Fundamentals - A Comprehensive Introduction, Herbet Schidt and Dale Srien,TMH.
2. Introducing JavaFX 8 Programming (Oracle Press)- Herbet Schidt

**REFERENCES BOOKS:**

1. Java for Programmers, P.J. Deitel and H.M. Deitel, Pearson education
2. Object Orientd Programming through Java, P. Radha Krishna, Universities Press

**ONLINE REFERENCES:**

1. https://www.amazon.in/Design-Patterns-Object-Oriented-Addison-Wesley-Professional-ebook/dp/B000SEIBB8
2. https://www.amazon.in/Introducing-JavaFX-Programming-Oracle-Press/dp/0071842551

CHANGE OF SYLLABUS

| **Unit No** | **Changes Incorporated** |
|---|---|
| Unit – 5 | **Concurrent classes, Database connectivity** |
| **Change of Syllabus: 10%** | |

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Object Oriented Object Oriented Programming Through Java | 23IT4115 | R23 (2024) | Exception Handling<br>Collection Framework in Java<br>GUI Programming with Java<br>Event Handling | Employability |

# COMPUTER ORGANISATION LAB

| | |
|---|---|
| COURSE CODE: 23EC3211 | CREDITS 1.5 |
| L T P S | Sessional Marks: 50 |
| 0 0 3 0 | End Exam Marks: 50 |
| | End Exam:3 Hours |

Prerequisite: Digital electronics concepts

Course Objectives:

1. To introduce Digital logic design software for constructing various types of digital circuits used in real time applications.
2. To understand how to design various combinational and sequential circuits.
3. To develop and test VHDL Program code for combinational and sequential circuits.

Course Outcomes:

After course completion, the students will be able to:

1. Design and verify various logic gates & their functions
2. Design a combinational circuit used for code conversion.
3. Design, Analyze and Implement various combinational logic circuits.
4. Design, Analyze and Implement various sequential logic circuits.
5. Design & verify the functionality of Shift registers, Arithmetic Logic unit (ALU).

**Mapping of Course Outcomes with POs and PSO:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | | | | | | | | | | | | |
| **CO2** | 3 | 2 | | | | | | | | | | | | |
| **CO3** | 3 | 3 | 3 | 3 | | | | | | | | | | |
| **CO4** | 3 | 3 | 3 | 3 | | | | | | | | | | |
| **CO5** | 3 | 3 | 3 | 3 | | | | | | | | | | |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

The students are required to design and develop VHDL source code to perform simulation using relevant simulator and analyze the obtained simulation results.

**LIST OF EXPERIMENTS:**

1. Verification of Logic gates with truth tables (AND, OR, NOT, NAND, NOR, XOR, XNOR).
2. Implementing Binary Adders (HALF ADDER & FULL ADDER).
3. Implementing Binary Subtractors (HALF SUBTRACTOR & FULL SUBTRACTOR).
4. Implementing Binary-to-Gray & Gray-to-Binary Code Converters.
5. Implementing Multiplexers (4×1 or 8×1 or 16×1)
6. Implementing De-Multiplexers (1×4 or 1×8 or 1×16)
7. Implementing Decoders(2×4 or 3×8)

8. Implementing Encoders(4×2 or 8×3)
9. Verify the characteristic tables of the Flip-flops (SR, D, JK, T).
10. Design of Shift Registers.
11. Design of Bidirectional Universal Shift Register.
12. Design of Counters.
13. Design of n-bit Arithmetic Logic Unit(ALU)

**Equipment Required:**

1. Xilinix ISE software-latest version
2. Personal computer with necessary pheripherals.

REFERENCE BOOKS:

1. M. Morris Mano, Digital Design, Pearson Education, Inc., 2008, 4th Edition.

# ADVANCED DATA STRUCTURES LAB

COURSE CODE: 23IT4213 Credits: 1.5

L T P S Sessional Marks: 50

0 0 3 0 End Exam Marks: 50

Prerequisites: C++ / Java

Course Objectives:

- Apply the concepts of advanced data structures to real world applications and implement then using object oriented programming language.

Course Outcomes:

After course completion, the students will be able to:

1. Develop and implement advanced data structures such as dictionaries, skip lists, AVL trees, Red-Black trees, binary heaps, and binomial queues to manage and optimize various applications like inventory systems, contact management, event scheduling, and job scheduling. (L3)
2. Utilize fundamental graph algorithms, including BFS, DFS, Prim’s, Kruskal’s, and Dijkstra’s algorithms, to solve practical problems such as public transportation navigation, road and bridge network construction, and efficient route planning for logistics. (L3)
3. Implement and analyze text processing algorithms and data structures, such as Boyer-Moore, Knuth-Morris-Pratt, tries, and Huffman coding, to handle tasks like malicious code detection, log file analysis, and search engine autocomplete features. (L3)

**Mapping of Course Outcomes with POs and PSO:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 3 | 2 | 3 | 1 | 1 | 1 | 3 | 2 | 2 | 2 | 1 | 3 |
| **CO2** | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 1 | 3 | 2 | 2 | 2 | 1 | 3 |
| **CO3** | 3 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 3 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**List of Programs**

1. Assume that you are working for a company that manages a large library of books. Each book has a unique identifier (ID), title, author, genre, and publication year. The company wants to implement a system that allows for efficient retrieval of information about the books based on different criteria such as ID, author, genre, or a combination of these (dictionaries). – CO1
2. Imagine that you are a software engineer at an e-commerce company that manages a large online store. The store’s inventory system needs to handle a variety of operations efficiently, including adding new products, removing discontinued products, and searching for specific products based on their unique IDs. The current implementation using a balanced binary search tree is proving to be a bottleneck as the inventory grows.
   Design and implement a skip list to manage the inventory of the online store. The skip list should support the following operations efficiently: – CO1
   Search(ProductID): Return the details of the product with the given ProductID.
   Insert(ProductID, ProductDetails): Add a new product to the inventory.
   Delete(ProductID): Remove a product from the inventory.

Update(ProductID, ProductDetails): Update the details of an existing product.

3. If you are a software engineer tasked with implementing a contact management system. The system must efficiently handle insertions, deletions, and lookups of contacts. Each contact is identified by a unique name. Write a program ensuring that the contact list is always balanced and operations remain efficient (AVL tree). – CO1
4. Imagine that you are tasked with designing an event scheduling system where events are added dynamically. Each event is identified by a unique time stamp. Design and implement a program to ensure that the schedule remains balanced and that operations like insertion, deletion, and lookup are efficient (Red-Black Tree). – CO1
5. Write a program to implement job scheduling, where tasks with higher priority need to be executed before others. – CO1

   **Hint:** A min-heap can be used to always retrieve the task with the highest priority efficiently.
6. Assume that you are a software engineer developing a navigation system for a city's public transportation network. The city map is represented as a graph where each node is a bus stop and each edge is a direct bus route between stops. You need to implement two features: finding the shortest path (in terms of the number of stops) from one bus stop to another using BFS, and checking if there's any path between two bus stops using DFS. – CO2
7. If you are a civil engineer planning to build a network of roads to connect several towns. Each town is represented as a node in a graph, and each road between towns is represented as an edge with a weight (cost) indicating the distance between them. Use Prim's algorithm to find the minimum cost to connect all towns with roads. – CO2
8. If you are a project manager overseeing the construction of a network of bridges to connect several islands. Each island is represented as a node in a graph, and each potential bridge between islands is represented as an edge with a weight (cost) indicating the construction cost. Use Kruskal's algorithm to find the minimum cost to connect all islands with bridges. – CO2
9. Imagine that you are a logistics manager planning the most efficient route for delivering goods between cities. Each city is represented as a node in a graph, and each direct route between cities is represented as an edge with a weight (distance or travel time). Apply Dijkstra's algorithm to find the shortest path from a starting city to a destination city. – CO2
10. Assume that you are a software developer working for a cybersecurity company. Your task is to build a tool that scans large text files for specific patterns of malicious code signatures. Apply Boyer-Moore pattern matching algorithm to implement this task. – CO3
11. Imagine you are working for a security software company that monitors large logs of network activity to detect potential intrusions or malicious behavior. One of the tasks involves searching for known malicious patterns within the network logs. You have a large log file where each entry is a sequence of characters representing network packets. You need to search for a specific pattern that indicates a potential attack. This pattern can occur multiple times within the log, and the log itself is very large. (Knuth-Morris-Pratt algorithm). – CO3
12. Imagine you work for a company that develops a search engine. One of your tasks is to implement an autocomplete feature for search queries. The autocomplete feature should suggest possible completions of a user's partial input in real-time. This requires efficient storage and retrieval of a large dictionary of terms. You need to design a system that can quickly suggest words based on a prefix provided by the user. The dictionary of terms is large, containing thousands of words, and the system needs to

handle real-time user input with minimal delay. Write a program to check and correct misspelled words in a given text. – CO3

CHANGE OF SYLLABUS

**New Lab**
**Change of Syllabus: 100%**

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Advanced Data Sructures Lab | 23IT4217 | R23 (2024) | Dictionaries<br>Skip lists<br>AVL Tress<br>Red Black Trees<br>Binary Heap<br>Graphs<br>Pattern Matching | Skill Development |

# OBJECT ORIENTED PROGRAMMING USING JAVA LAB

COURSE CODE: 23IT4214 CREDITS 1.5

L T P S Sessional Marks: 50

0 0 3 0 End Exam Marks: 50

End Exam:3 Hours

Prerequisites: Basic knowledge on introduction to programming & object oriented programming conceptsare essential.

Course Objectives:

- To understand object oriented programming concepts, and apply them in problem solving.
- To familiarize the concepts of Exception handling, File I/O and Database connectivity.
- To introduce the design of Graphical User Interface using Awt,swing and JavaFX controls.

Course Outcomes:

After course completion, the students will be able to:

1. Implement OOP'S concepts using Java programming in problem solving
2. Solve problems using Java Files, collections, Exception handling and multithreading
3. Develop simple and complex UI applications using GUI components and databases.

**Mapping of course outcomes with program outcomes:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 2 | 2 | 3 | | | | 3 | 1 | 2 | 3 | 2 | 2 |
| **CO2** | 3 | 2 | 2 | 2 | 3 | | | | 3 | 1 | 2 | 3 | 2 | 2 |
| **CO3** | 3 | 1 | | | 3 | | | | | | | 1 | 3 | 2 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**List of Programs:**

1. Write a java program to calculate gross salary & net salary taking the following data. [CO-1]

   Input : empno,empname,basic

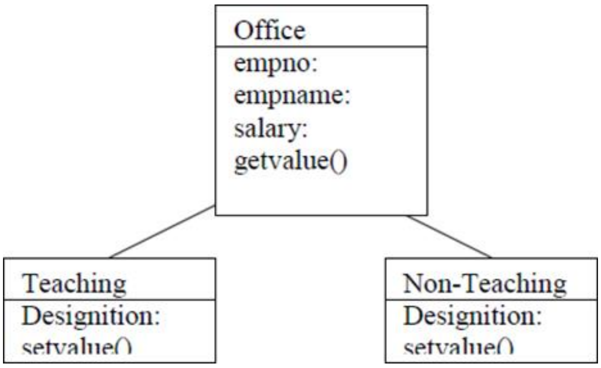
   Process: DA=50%of basic

   HRA=25%of basic

   PF=10%of basic

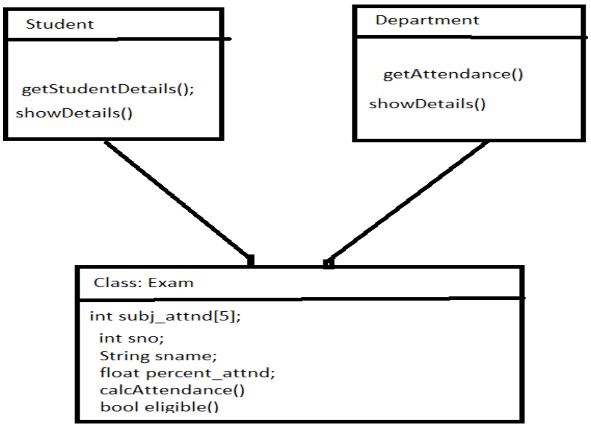
   PT=Rs100/-

2. Write a java program that implements educational hierarchy using inheritance. [CO1]



3. Write a program to identify the accessibility of a variable by means of different access specifies within and outside package. [CO1]

4. Write a java program to find the details of the students eligible to enroll for the examination (Students, Department combined give the eligibility criteria for the enrolment class) using interfaces. [CO1]



5. Write a Java program that displays area of different Figures (Rectangle, Square, Triangle) using the method overloading. [CO1]

6. Write a Java program that displays that displays the time in different formats in the formof HH,MM,SS using constructor Overloading. [CO1]

7. Write a Java program that counts the number of objects created by using static variable.[CO1]

8. Write a Java program to count the frequency of words, characters in the given line of text. [CO2]

9. Write a Java program for sorting a given list of names in ascending order. [CO2]

10. Write a Java program that reads a line of integers separated by commas and then displayseach integer and fund the sum of the integers (using String Tokenizer). [CO2]

11. Write a Java program that reads a file name from the user then displays information about whether that file exists, file is writable, the type of file and length of the file in bytes. [CO2]

12. Write a Java program that reads a file and displays the file on the screen with a line number before each line. [CO2]

13. Write a Java program that reads a file and displays the no of lines and words in that

file.[CO2]

14. Write a Java program that reads to copy source File and display on the console. [CO2]
15. Write a java program that implements Array Index out of bound Exception using built-in-Exception. [CO2]
16. Write a java program that implements bank transactions using user denied exception.[CO2]
17. Write a java program to identify the significance of finally block in handling exceptions. [CO2]
18. Write a java program to generate multiple threads of creating clock pulses .(using runnable interface). [CO2]
19. Write a java program to identify the use of synchronized blocks to synchronized methods. [CO2]
20. Write a java program to demonstrate key events by using Delegation event model. [CO3]
21. Write a java program to implement mouse events like mouse pressed, mouse releasedand mouse moved by means of adapter classes. [CO3]
22. Write a java program to demonstrate window events on frame. [CO3]
23. Write a java program to design a registration form for creating a new eMail account.[CO3]
24. Write a java program to design the page authenticating user name and password by usingSWING. [CO3]
25. Write a java program to design a calculator by using Grid Layout. [CO3]
26. Write a JDBC program which creates customer table, insert data into it and retrieves data from customer. [CO3]

CHANGE OF SYLLABUS

| **Exp. No** | **Changes Incorporated** |
|---|---|
| 20 | **Write a JDBC program which creates customer table, insert data into it and retrieves data from customer** |
| **Change of Syllabus: 5%** | |

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Object Oriented Programming Using Java Lab | 23IT4214 | R23 (2024) | Complete Lab | Employability / Skill Development |

# LOGICAL REASONING & CORPORATE SKILLS

COURSE CODE: 23CR9101 CREDITS 1

L T P S Sessional Marks: 100

2 2 0 0

Prerequisites: Knowledge of LSRW Skills, Basic Maths

Course Outcomes:

After course completion, the students will be able to:

1. Build career-oriented demeanour and learn corporate culture to work in teams with LSRW skills (L3), Detect errors of grammar and usage in a given sentence/text and rectify them by making appropriate changes
2. Acquire leadership qualities through verbal, nonverbal, written and emotional intelligence. (L3), Detect errors of grammar and usage in a given sentence/text and rectify them by making appropriate changes
3. Analyse different types of vocabulary and do model papers to obtain LSRW skills. (L4), Detect errors of grammar and usage in a given sentence/text and rectify them by making appropriate changes
4. Use their logical thinking and analytical abilities to solve reasoning questions from number analogy and series and letter based aptitude questions company specific and other competitive tests.
5. Solve questions related to clock and calendar , etc..from company specific and other competitive tests.

**Mapping of course outcomes with program outcomes:**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | | | | | | | | | **2** | **2** | | **2** | | |
| **CO2** | | | | | | | | | **2** | **2** | | **2** | | |
| **CO3** | | | | | | | | | **2** | **2** | | **2** | | |
| **CO4** | **2** | | | | | | | | | | | | | |
| **CO5** | **2** | | | | | | | | | | | | | |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

## PART-A: Verbal Ability Skills

UNIT-I 10 Periods

Corporate Etiquette – Work Place Etiquette and Conflict Resolution - Grammar Revision Verbal Ability : Prepositions, Articles, tenses and conjunction

UNIT-II 10 Periods

EQ – Negotiation Skills – Telephone Etiquette – MNCs Paper Model Introduction, Situational Dialogue Practice – Team Activities Related to Spoken English

Verbal Ability: Fill in the blanks (Based on the given appropriate words)

UNIT-III 10 Periods

E Mail Writing – Vocabulary from Story Telling Activity –MNCs Model Paper 1 Practice

Verbal Ability: Sentence arrangements

UNIT-IV 10 Periods

Virtual Reading – Functional English – IELTS Vocabulary – News Paper Reading Using AI Based Applications

Verbal Ability: Inferred meaning (Homophones, Homonyms)

**PART-B: Logical Reasoning**

UNIT-I

Numerical computation: Number Series, Letter Series, Number analogy, letter analogy, word analogy

UNIT-II

Coding Decoding- Letter to letter, letter to digit, letter to number and symbol, Word to word coding, odd man out

UNIT-III

Directions-Finding distance, Direction and Shadow based problem, Blood Relations-Mixed Blood Relations, Puzzle-Based Blood Relation, Single-Person Blood Relation, Symbol based Blood Relations.

UNIT-IV

Clocks –finding Angle, Time, Mirror image, Faulty clock, Calendars – Finding day of the week, Number of odd days, Repetition of same calendar

UNIT-V

Seating Arrangement-Circular arrangement, linear arrangement ,Order Sequence and Ranking

## References:

1. Deb Boelkes. Heartfelt Leadership: How to Capture the Top Spot and Keep on Soaring. Business World Rising, LLC, 2020.
2. Marty Strong Navy. Strategic Leadership in the Age of Optimization. Business Books, 2023.
3. Ken Lawson. Successful Negotiating: Identify Points for Negotiation, Understand Win-Win Situations, Communicate Wants and Needs Effectively, Set and Respond to Agendas. New Holland Publishers Ltd. 2009.
4. Barbara Pachter. The Essentials of Business Etiquette: How to Greet, Eat, and Tweet Your Way to Success. McGraw-Hill, UK, 2013.
5. Bradberry, Travis & Jean Greaves. Emotional Intelligence 2.0. Perseus Books Group. 2009.

# NETWORK FUNDAMENTALS

COURSE CODE: 23IT9301 CREDITS 1

| L | T | P | S |
|---|---|---|---|
| 0 | 0 | 0 | 2 |

https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012683751296065536354_shared/overview

# FINANCIAL LITERACY

COURSE CODE: 23MC0103 CREDITS 0

L T P S Sessional Marks: 100

3 0 0 0

Prerequisites: NIL

Course Objectives:

- The course has been designed to give familiarity with different aspects of financial literacy such as savings, investment, taxation, and insurance and understand the relevance and process of financial planning.

Course Outcomes:

After course completion, the students will be able to:

1. Recognize the role of saving money in reaching financial goals and identify components of a spending plan
2. Describe the importance of banks and their purpose as financial institutions.
3. Apply the concept of investment planning.
4. Ability to analyse banking and insurance products.
5. Estimate Personal tax.

**Mapping of course outcomes with program outcomes:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | | | | | | 1 | | 1 | | | 2 | 2 | | |
| **CO2** | | | | | | 1 | | 1 | | | 2 | 2 | | |
| **CO3** | | | | | | 1 | | 1 | | | 2 | 2 | | |
| **CO4** | | | | | | 1 | | 1 | | | 2 | 2 | | |
| **CO5** | | | | | | 1 | | 1 | | | 2 | 2 | | |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT – I 8 Periods

Introduction to Financial Planning: Introduction to saving: Benefits of Savings-Saving vs Investment, Investment vs Gambling-Time value of money-Management of spending and financial discipline.

UNIT – II 8 Periods

Banking and Digital Payment: Banking products and services -Savings account, Current Account, Fixed deposits, Recurring deposits-Digitisation of financial transaction- Modes of digital payments: Debit cards, Credit cards, Net banking and UPI,-Digital Wallets-Role of RBI in banking sector.

UNIT – III 8 Periods

Financial Markets and Investment Planning: Financial Markets: Primary and Secondary markets- Securities and its types, i.e., Equity, Debentures or Bonds, IPOs and FPOs-Mutual Funds: Types of Mutual Funds-Stock Market, DEMAT.

UNIT – IV 8 Periods

Insurance Services: Life Insurance: Policies- Term insurance, Endowment policies, Pension policies-Health Insurance Plans- ULIP-General Insurance-Understanding of Ponzi Schemes.

UNIT – V 8 Periods

Personal Tax: Introduction to basic tax structure in India for personal taxation-Basic concepts of Income Tax- Exemption and Deduction for individual-Income Tax Act, 1961-E-Filling.

TEXT BOOKS:

1. Introduction to Financial Planning (4th Edition 2017)- Indian Institute of Banking & Finance.
2. Sinha, Madhu. Financial Planning: A Ready Reckoner July 2017, McGraw Hill.

REFERENCE BOOKS:

1. Halan, Monika, Lets Talk Money: You've Worked Hard for It, Now Make It Work for You, July 2018 Harper Business.
2. Pandit, Amar The Only Financial Planning Book that You Will Ever Need, Network 18 Publications Ltd.

WEB RESOURCES:

1. https://onlinecourses.nptel.ac.in/noc21_mg40/preview
2. https://corporatefinanceinstitute.com/resources/management/financial-literacy/

# SEMESETER-2

# PROBABILITY & STATISTICS

| | |
|---|---|
| COURSE CODE: 23MA1103 | CREDITS 3 |
| L T P S | Sessional Marks: 40 |
| 3 0 0 0 | End Exam Marks:60 |
| | End Exam:3 Hours |

Prerequisite(s): Elementary knowledge of set theory, combinations and basic statistics

Course Objectives:

To provide the required mathematical support in real life problems and develop probabilistic models which can be used in several areas of science and engineering.

Course Outcomes:

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

1. Analyze the basic principles of statistical measures and probability.
2. Demonstrate a random variable that describes randomness or uncertainty in certain realistic situation.
3. Differentiate the concepts of discrete, continuous probability distributions and able to solve problems of probability.
4. Evaluate simple correlation between the two variables and fit curves by the method of least square approximation.
5. Analyze the statistical data and apply various small and large samples tests for testing the hypothesis.

**Mapping of Course Outcomes with POs and PSO:**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | | | | | | | | | | 1 | | |
| **CO2** | 3 | 2 | | | | | | | | | | 1 | | |
| **CO3** | 3 | 2 | | | | | | | | | | 1 | | |
| **CO4** | 3 | 2 | | | | | | | | | | 1 | | |
| **CO5** | 3 | 2 | | | | | | | | | | 1 | | |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT I 10 Periods

STATISITCAL METHODS AND DATA ANALYSIS: Measures of central tendency : Mean – Median – Mode. Measures of dispersion : Mean deviation – Standard deviation – Variance.

PROBABILITY: Introduction to Probability : Definition of random experiment – Events and sample space – Definition of probability – Addition and multiplication theorems Conditional probability – Baye's theorem – Simple problems on Baye's theorem.

UNIT II 10 Periods

RANDOM VARIABLES: Discrete and continuous random variables – Distribution function of random variable – Properties, Probability mass function, Probability density function – Mathematical expectation – Properties of mathematical expectation – Moments – Moment generating function – Mean and variance – Simulation of random variable – Solving problems by using Monte Carlo method.

UNIT III 10 Periods

PROBABILITY DISTRIBUTIONS: Discrete Distributions: Binomial distribution – Poisson distribution – Mean, Variance, Moment Generating function and problems.

Continuous Probability Distributions: Uniform distribution – Exponential distribution, Memoryless property – Normal distribution – Properties of normal distribution – Importance of normal distribution – Area properties of normal curve – MGF – Mean ,variance and simple problems.

UNIT IV 10 Periods

CORRELATION, REGRESSION ANALYSIS AND CURVE FITTING

Correlation: Definition – Karl pearson's coefficient of correlation – Measures of correlation – Rank correlation coefficients.

Regression: Simple linear regression – Regression lines and properties.

Curve Fitting: Principle of least squares – Method of least squares – Fitting of straight lines – Fitting of second degree curves and exponential curves.

UNIT V 10 Periods

TESTING OF HYPOTHESIS: Introduction – Null hypothesis – Alternative hypothesis – Type – I, II errors – Level of significance – Critical region – Confidence interval – One sided test – Two sided test.

Small Sample Tests: Students t - distribution and its properties – Test of significance difference between sample mean and population mean – Difference between means of two small samples – F- Distribution – Test of equality of two population variances – Chi-square test of goodness of fit .

Large sample Tests: Test of significance of large samples – Tests of significance difference between sample proportion and population proportion & difference between two sample proportions – Tests of significance difference between sample mean and population mean & difference between two sample means.

TEXT BOOK:

1. T. Veerarajan, Probability, Statistics and Random Processes, Tata McGraw Hill Publications.

REFERENCE BOOKS:

1. Kishor S. Trivedi, Probability & Statistics with Reliability, Queuing and Computer Applications, Prentice Hall of India .
2. B. S. Grewal, Higher Engineering Mathematics, 43rd edition, Khanna publishers, 2017.
3. Sheldon M. Ross, Probability and Statistics for Engineers and Scientists, Academic Press.
4. S C Gupta and V.K.Kapoor, Fundamentals of Mathematical Statistics.

# DATABASE MANAGEMENT SYSTEMS

COURSE CODE: 23IT4116 CREDITS 3

L T P S Sessional Marks: 40

3 0 0 0 End Exam Marks:60

End Exam:3 Hours

Prerequisite(s): Relational Algebra, Set Theory, knowledge in any program language.

Course Objectives:

- Understand basic database concepts, including the structure and operation of the relational data model.
- Construct simple and moderately advanced database queries usingo Structured Query Language (SQL).
- Understand and successfully apply logical database design principles, including ER diagrams and database normalization.
- Understand the concept of a database transaction and related databaseo facilities, including concurrency control, locking and protocols.

Course Outcomes:

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

1. Explain the evolution, applications, and structure of database systems, highlighting the differences between database systems and traditional file systems. (L2)
2. Apply data modeling techniques, including Entity-Relationship diagrams and relational models, to design and conceptualize databases for various applications, ensuring data integrity and logical organization. (L3)
3. Formulate and execute queries using relational algebra, relational calculus, and SQL, and apply query optimization techniques to enhance database performance. (L3)
4. Analyze and implement concurrency control mechanisms, transaction management protocols, and recovery techniques to maintain data integrity and handle concurrent database transactions effectively. (L3)

**Mapping of Course Outcomes with POs and PSO:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 1 | 1 | 3 | | | | 2 | 2 | | 2 | 2 | 1 |
| **CO2** | 3 | 2 | 1 | 1 | 3 | | | | 2 | 2 | | 2 | 2 | 1 |
| **CO3** | 3 | 2 | 1 | 1 | 3 | | | | 2 | 2 | | 2 | 2 | 1 |
| **CO4** | 3 | 2 | 1 | 1 | 3 | | | | 2 | 2 | | 2 | 2 | 1 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT-I 12 Periods

History of Database Systems, Database System Applications, database System vs file System – View of Data – Data Abstraction –Instances and Schemas – data Models – the ER Model – Relational Model – Other Models – Database Languages – DDL, DML – Transaction Management – database System Structure – Storage Manager – the Query Processor.

**Database design and E-R diagrams** – Beyond E-R Design Entities, Attributes and Entity sets – Relationships and Relationship sets –Additional features of ER Model – Concept Design with the ER Model – Conceptual Design for Large enterprises.

Learning Outcomes:

At the end of the module, the student will be able to

1. Describe when to use files and when to use a DBMS. (L2)
2. Explain how data can be stored and processed. (L2)
3. Apply data modelling tools like Entity-Relationship Diagrams. (L3)

UNIT-II 10 Periods

**Introduction to the Relational Model** – Integrity Constraint Over relations – Enforcing Integrity constraints – Querying relational data – Logical database Design – Introduction to Views – Destroying/altering Tables and Views. Relational Algebra – Selection and projection set operations – renaming – Joins – Division – Relational calculus –Tuple relational Calculus– Domain relational calculus.

Learning Outcomes:

At the end of the module, the student will be able to

1. Describe the data using a relational model. (L2)
2. Solve queries using relational algebra and calculus (L3)
3. Summarize what views are for and how to use them. (L2)

UNIT-III 8 Periods

**Schema refinement** – Problems Caused by redundancy – Decompositions – Problem related to decomposition – reasoning about FDS – FIRST, SECOND, THIRD Normal forms – BCNF– Schema refinement in Database Design – Multi valued Dependencies – FOURTH Normal Form.

Learning Outcomes:

At the end of the module, the student will be able to

1. Examine the anomalies in a database (L3)
2. Determine the keys for a given set of functional dependencies. (L3)
3. Apply the normal forms to normalize the tables. (L3)

UNIT-IV 12 Periods

**Transaction Processing:** ACID properties – Concurrent Executions-Conflict serializability - view serializability - Concurrency Control: Lock – Based Protocols-Deadlock Handling-Time stamp Based Protocols- Multiple Granularity. Advance Recovery systems- ARIES, Log, the Write–ahead Log Protocol, check pointing, and Recovering from a System Crash.

Learning Outcomes:

At the end of the module, the student will be able to

1. Demonstrate the logging techniques used to ensure Atomicity and Durability. (L3)
2. Summarize the anomalies that occur without ACID properties. (L2)
3. Explain how different indexing techniques work. (L2)

UNIT-V 8 Periods

Primary and Secondary Indexes – Index data structures – Hash-Based Indexing – Tree base Indexing – B+ Trees: A Dynamic Index Structure.

**Query Processing and optimization**: Translating SQL Queries into Relational Algebra, Algorithms for External Sorting, Algorithms for SELECT and JOIN Operations, Algorithms for PROJECT and Set Operations, Implementing Aggregate Operations and OUTER JOINs, Combining Operations Using Pipelining, Using Heuristics in Query Optimization, Using Selectivity and Cost Estimates in Query Optimization

Learning Outcomes:

At the end of the module, the student will be able to

1. Understand different types of indexes and their importance (L2)
2. Apply query processing and optimization for relational databases (L3)

TEXTBOOKS:

1. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", 3rd Edition, McGraw-Hill, 2014.
2. Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database Systems" 6th Edition, Addison-Wesley, Pearson.

REFERENCES:

1. Silberschatz, Korth and Sudharshan, "Data Base System Concepts", 6th Edition, McGraw Hill, 2013.
2. C.J.Date, an introduction to Database Systems, 8th Edition, Pearson Education, 2003.
3. Peter Rob & Carlos Coronel, Database Systems design, Implementation, and Management, 9th Edition, Pearson Education, 2009.
4. Sibsankar Haldar, SQLite Database System Design, and Implementation, O'Reilly publications, 2nd Edition, 2015

WEB REFERENCES:

1. https://onlinecourses.nptel.ac.in/noc21_cs58/preview
2. https://www.mongodb.com/docs/manual/tutorial/getting-started/

CHANGE OF SYLLABUS

| **Unit No** | **Changes Incorporated** |
|---|---|
| Unit – 4 | Advance Recovery systems- ARIES, Log, the Write–ahead Log Protocol, check pointing, and Recovering from a System Crash**.** |
| Unit – 5 | Primary and Secondary Indexes – Index data structures – Hash-Based Indexing – Tree base Indexing – B+ Trees: A Dynamic Index Structure.<br>Query Processing and optimization: Translating SQL Queries into Relational Algebra, Algorithms for External Sorting, Algorithms for SELECT and JOIN Operations, Algorithms for PROJECT and Set Operations, Implementing Aggregate Operations and OUTER JOINs, Combining Operations Using Pipelining, Using Heuristics in Query Optimization, Using Selectivity and Cost Estimates in |

| | |
|---|---|
| | Query Optimization |
| **Change of Syllabus: 30%** | |

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Database Management Systems | 23IT4116 | R23 (2020) | Relational Model<br>Query Optimization | Employability |

# COMPUTER NETWORKS

COURSE CODE: 23IT4117 Credits: 3

L T P S Sessional Marks: 40

3 0 0 0 End Exam Marks: 60

Prerequisite(s): Network Fundamentals

Course Objective:

- Build an understanding of the fundamental concepts of data transmission.
- Familiarize the student with the basics of encoding of analog and digital data
- Preparing the student for understanding advanced courses in computer networking

Course Outcomes:

After course completion, the students will be able to:

1. Define various data communication components in networking and Describe networking with reference to different types of models and topologies.
2. Summarize the concepts related to error detection, correction and error control techniques.
3. Identify the features of the different IP address classes and different routing types.
4. Estimate the congestion control mechanism to improve quality of service in transporting data through different protocols such as TCP, UDP, Understand and analyse application layer protocols and internet applications

**Mapping of Course Outcomes with POs and PSO:**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 1 | 2 | 2 | 2 | 3 | 2 | 2 | 1 | 2 | 2 | 3 | 2 | 3 |
| **CO2** | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 3 | 3 | 3 | 2 |
| **CO3** | 3 | 1 | 2 | 2 | 3 | 3 | 2 | 1 | 2 | 2 | 3 | 2 | 2 | 3 |
| **CO4** | 3 | 2 | 3 | 2 | 3 | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 3 | 3 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT – I 12 Periods

**Introduction**: Components – Direction of Data flow – Networks –Categories – Types of Connections – Topologies –Protocols and Standards – OSI, TCP/IP models, Example Networks such as ATM, Frame Relay. Physical layer: Transmission modes, Multiplexing, Transmission Media, Switching, Circuit Switched Networks.

Learning outcomes: At the end of the unit the students are able to

1. Independently understand basic computer network technology and design of LANs both wired and wireless in an Ethernet network.
2. Identify the different types of network devices and their functions within a network

UNIT – II 10 Periods

**Data link layer**: Introduction, Framing, and Error – Detection and Correction – Parity –

– CRC Hamming code, Flow and Error Control, Noiseless Channels, Noisy Channels, HDLC, Point to Point Protocols. Medium Access sub layer: ALOHA, CSMA/CD, LAN – Ethernet IEEE 802.3, IEEE 802.11, introduction to IEEE 802.15, Random access, Controlled access, Channelization.

Learning outcomes: At the end of the unit the students are able to

1. Classify and apply appropriate error detection and correction algorithms for integrity of Data
2. Understand the working of data link layer protocols in providing a shared resource in data transfer

UNIT – III 10 Periods

**Network layer**: Datagram Networks, Virtual Circuit Networks. Logical Addressing, IPv4, Introduction to IPv6, Internetworking, Tunnelling, Address mapping, Address Translation, ICMP, IGMP, Uni-Cast Routing Protocols, Multicast Routing Protocols.

Learning outcomes: At the end of the unit the students are able to

1. Understand and building the skills of subnetting and routing mechanisms
2. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.
3. Classify the routing protocols and analyze how to assign the IP addresses for the given network

UNIT – IV 10 Periods

**Transport Layer:** Process to Process Delivery, UDP and TCP protocols, Data Traffic, Congestion, Congestion Control, QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.

Learning outcomes: At the end of the unit the students are able to

1. Understand the building principals of end to end communication
2. Able to distinguish protocols used in several real time applications and choose appropriate protocol in application design.
3. Able to trouble suite congestion issues in a network by providing best QoS.

UNIT – V 8 Periods

**Application Layer:** Domain name space, DNS in internet, Network Load Balancing, electronic mail, SMTP, FTP, WWW, HTTP, HTTPS, SNMP, SSL 3, TLS 1.2.

Learning outcomes: At the end of the unit the students are able to

1. Understand the building principals of several application layer protocols such as DNS, Email and HTTP.
2. Able to innumerate the design principals of protocols in designing an application

**TEXT BOOKS:**

1. Data Communications and Networking with TCPIP Protocol Suite, By Behrouz A. Forouzan, 6th Edition, TMH, Published: August 2, 2022.
2. Computer Networks, Andrew S Tanenbaum, 6th Edition. Pearson Education.

**REFERENCE BOOKS:**

1. Data communications and Computer Networks, P.C .Gupta, PHI.
2. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education.
3. Understanding communications and Networks, 3rd Edition, W.A. Shay, Cengage Learning.
4. Computer Networking: A Top-Down Approach Featuring the Internet. James

F.Kurose & Keith W. Ross, 3 rd Edition, Pearson Education.
5. Data and Computer Communication, William Stallings, Sixth Edition, Pearson Education,

CHANGE OF SYLLABUS

| Unit No | Changes Incorporated |
|---|---|
| 3 | IPv4, introduction to IPv6 |
| 5 | Network Load Balancing, HTTPS, SSL 3, TLS 1.2 |
| **Change of Syllabus: 10%** | |

**1.1.3 of NAAC**

| Name of the Course | Course Code | Year of Introduction | Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development | Mapping with Employability/Skill development/ Entrepreneurship |
|---|---|---|---|---|
| Data Communication and Computer Networks | 23IT4117 | R23 (2024) | ISO / OSI mode<br>Digital Transmission<br>Logical Addressing, Internetworking<br>Address mapping, ICMP, IGMP<br>Domain name space, DNS in internet, electronic mail, SMTP, FTP, WWW, HTTP, SNMP | Employability |

# PYTHON PROGRAMMING FOR DATA SCIENCE

COURSE CODE: 23IT4118 Credits: 3

L T P S Sessional Marks: 40

1 0 4 0 End Exam Marks: 60

Prerequisites: Basic knowledge of any programming language concepts like conditional statements, iterative statements, functions, OOP’s concepts. and basic mathematics.

Course Objectives:

1. Describe the core syntax and semantics of Python programming language. Illustrate the process of structuring the data using lists, dictionaries, tuples, strings and sets.
2. Discover the need for working with the functions, modules and packages.
3. Infer the Object-oriented Programming concepts in Python.
4. Familiarize the advanced concepts like regular expressions, date and time. Able to handle abnormal termination of the python scripts.

Course Outcomes:

After course completion, the students will be able to:

1. Interpret the fundamental Python syntax and semantics and able to solve, test and debug python programs
2. Fluency in the use of Python control flow statements and determine the methods to create and manipulate Python programs by utilizing the data structures like lists, dictionaries, tuples, strings and sets.
3. Express proficiency in the handling of functions, modules, regular expressions, packages and handle abnormal termination of the programs.
4. Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance and polymorphism as used in Python.

**Mapping of Course Outcomes with Program Outcomes:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 2 | 3 | | | | | | | | | | 1 | 1 |
| **CO2** | 2 | 2 | 2 | | | | | | | | | 1 | 1 | 1 |
| **CO3** | 3 | 2 | 3 | | 3 | | | | | | | 1 | 2 | 2 |
| **CO4** | 3 | 2 | 3 | 3 | 3 | | | | | | | 1 | | |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Unit-I 8 Periods

**Introduction to Python:** History and Features of Python, Variables, Reserved words, Data types, Operators, Input and output, Indentation, Comments and Documentation, Lists, Set, Dictionaries, Tuple. Strings: Creating strings and basic operations on strings, string slicing and indexing.

**Learning Outcomes:** At the end of this unit, Students are able to

1. Analyse fundamental advantages of python over the other programming languages and Solve, test and debug basic problems using python scrip
2. Manipulate python programs by using the python data structures like lists, dictionaries, tuples, strings and sets.

Unit-II 10 Periods

Decision Control Statements, loop Control Statements, Functions: Defining a function, Calling a function, returning multiple values from a function, formal and actual arguments, positional arguments, default arguments, recursive functions. lambda functions, modules, Statistical functions like mean, fmean, harmonic_mean , median, median__low , median__high, median__grouped, mode, multimode, quantiles , geometric_mean

**Learning Outcomes**: At the end of this Unit the student will be able to

1. Implement Flow control statements required real world problems.
2. Resolve real world problems using python functions and statistical functions.

Unit-III 12 Periods

Exception Handling in Python, What is an Exception?, Syntax for Exception Handing, Handling Single Exception, Handling Multiple Exceptions.

The regex package, Regular expression methods: findall, finditer, match, search, split, sub, subn. Date and Time Data Types and Tools: The datetime package,Types in datetime module, Datetime format specification, Locale-specific date formatting, Time Series Basics, Date Ranges, Frequencies, and Shifting, Base Time Series Frequencies, Time Zone Handling.

**Learning Outcomes:** At the end of this Unit the student will be able to

1. Problem solving with the usage of exceptions
2. Resolve the problems like pattern matching and manipulation of time and date

Unit-IV 10 Periods

**NumPy package:** Arrays and Vectorized Computation - The NumPy ndarray - Creating ndarrays – Data Types for ndarrays – Arithmetic with NumPy Arrays – Basic Indexing and Slicing - Boolean Indexing - Transposing Arrays and Swapping Axes.

**Pandas package**: Introduction to pandas Data Structures, Dataframe, Index objects, Reindexing, Dropping entries from axis, Indexing, selection, and filtering, Arithmetic and data alignment, Function application and mapping, Sorting and ranking, Axis indexes with duplicate values, Summarizing and Computing Descriptive Statistics, Correlation and Covariance, Unique Values, Value Counts, and Membership.

**Learning Outcomes**: At the end of this Unit the student will be able to

1.Familiarize the usage of Modules and packages to enhance the problem solving
2.Resolve real world problems using NumPy and Pandas

Unit-V 10 Periods

Handling Missing Data – Filtering Out Missing Data, Filling in Missing Data, Hierarchical Indexing. Data Transformation: Removing Duplicates, Replacing Values, Renaming Axis Indexes. Discretization and Binning: Detecting and Filtering Outliers.

Visualizing Data: matplotlib package: Bar Charts, Line Charts, Scatter plots. Histograms and Density Plots, Colors, Markers, and Line Styles,Visualizing iris Data, Seaborn package, Ticks, Labels, and Legends.

**Learning Outcomes**: At the end of this Unit the student will be able to

1.Process the real world data pre-processing

2.Analyse the real world data using visualizing tecniques

**TEXT BOOKS:**

1. Wes McKinney, “Python for Data Analysis”, O’REILLY, 2012.
2. Reema, Thareja, “Python Programming: Using Problems Solving Approach”, Oxford University Press, 2017
3. Gowrishankar S, Veena, “Introduction to Python Programming”, CRC Press/Taylor & Francis, 2019.

**REFERENCE BOOKS:**

1. Adnan Aziz, Luciano Ramalho, “Elements of Programming Interviews in Python: The Insiders' Guide, Fluent Python: Clear, Concise, and Effective Programming”, Createspace Independent Pub, 2016.
2. Vamsi Kurama, Pearson, “Python Programming: A Modern Approach”, Pearson India**,** 2017
3. Jake VanderPlas, “Python Data Science Handbook: Essential Tools for Working with Data”, Kindle Edition, O’REILLY,2016.
4. Wesley J. Chun, “Core Python Programming”, Prentice Hall,2006.

CHANGE OF SYLLABUS

| **Unit No** | **Changes Incorporated** |
|---|---|
| Unit – 3 | **Vectorized Computation** |
| Unit – 4 | **Frequencies, and Shifting** |
| Unit – 5 | **Handling Missing Data-Filtering Out Missing Data, Filling in Missing Data, Hierarchical Indexing. Data Transformation: Removing Duplicates, Replacing Values, Renaming Axis Indexes. Discretization and Binning: Detecting and Filtering Outliers.**<br>**Visualizing Data: matplotlib package: Bar Charts, Line Charts, Scatter plots. Histograms and Density Plots, Colors, Markers, and Line Styles, Visualizing iris Data, Seaboran package, Ticks, Labels, and Legends.** |
| **Change of Syllabus: 20%** | |

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/ Entrepreneurship** |
|---|---|---|---|---|
| Python Programming for Data Science | 23IT4118 | R23 (2020) | NumPy package<br>Pandas package | **Employability/Skill development** |

## PYTHON PROGRAMMING FOR DATA SCIENCE LAB EXPERIMENTS

1. Write a python script to perform different Arithmetic Operations on numeric data types in Python (CO1)

   Sample input: 2770

   Sample output: 772

2. Write a python Program to Demonstrate a Function with and without Argument (CO1)
3. Write a python program to create, append, and remove lists in Python. (CO1)
4. Write Python Program to Conduct a Linear Search for a Given Key Number in the List and Report Success or Failure. (CO1)

   **Sample Input 1:**

   ```
   Sample input [10,14,19,25,27,31,34,43,45,52] key = 27

   Element found at index:  4
   ```

   **Sample Input 2:**

   ```
   Sample input  [10,14,19,25,27,31,34,43,45,52]

   key = 16

   Element not found
   ```

5. Write a python program to demonstrate working with Tuple in Python. (CO1)
6. Write a python program to demonstrate working with dictionaries in Python. (CO1)
7. Program to Dynamically Build User Input as dictionaries. (CO1)
8. Write a python program to compute summary statistics such as mean, median, mode, standard deviation and variance of the given different types of data. (CO2)

   **Sample Output:**

   ```
   Original array:[0 1 2 3 4 5]

   Mean: 2.5

   std: 1

   variance: 2.9166666666666665
   ```

9. Write a python program to calculate the sum of every column in a NumPy array (CO3)

   **Sample Output:**

   ```
   [[ 1  2  3]

   [ 4  5  6]

   [7  8  9]]

   Sum of every column is [12 15 18 ]
   ```

10. Write a python program to calculate the sum of every row in a NumPy array (CO3)

    **Sample Output:**

    ```
    [[ 1  2  3]

    [ 4  5  6]

    [ 7  8  9]]

    Sum of every row is [ 6 15 24]
    ```

11. Write a NumPy program to compute the 80th percentile for all elements in a given array along the second axis. (CO3)

Sample Output:

```
Original array:
[[ 0  1  2  3  4  5]
[ 6  7  8  9 10 11]]
80th percentile for all elements of the said array along the second axis: [ 4. 10.]
```

12. Write a NumPy program to compute the median of flattened given array (CO3)

Sample Output:

```
Original array: [[ 0  1  2  3  4  5]
[ 6  7  8  9 10 11]]
Median of said array: 5.5
```

13. Write a NumPy program to compute the weighted of a given array (CO3)

Sample Output: Original array: [0 1 2 3 4]

```
Weighted average of the said array: 2.6666666666666665
```

14. Write a NumPy program to compute the covariance matrix of two given arrays (CO3)

Sample Output:

```
Original array1: [3 6 8]
Original array2: [2 4 7]
Covariance matrix of the said arrays: [[6.33333333 6.16666667] [6.16666667 6.33333333]]
```

15. Write a NumPy program to compute cross-correlation of two given arrays (CO3)

Sample Output:

```
Original array1: [4 7 9]
Original array2: [2 6 8]
Cross-correlation of the said arrays: [[6.33333333 7.66666667] [7.66666667 9.33333333]]
```

16. Write a Python NumPy program to compute the weighted average along the specified axis of a given flattened array (CO3)

Sample Output:

```
Original flattened array: [[0 1 2]
[3 4 5] [6 7 8]]
Weighted average along the specified axis of the above flattened array: [1.2 4.2 7.2]
```

17. Write a NumPy program to compute the histogram of nums against the bins (CO4)

Sample Output:

```
nums: [1.5 0.7 1.  1.2 1.3 2.75] bins:  [0 1 2 3]
Result: (array([1, 4, 1]), array([0, 1, 2, 3]))
```

18. Write a Pandas program to add, subtract, multiple and divide two Pandas Series. Sample Series: [2, 4, 6, 8, 10], [1, 3, 5, 7, 9] (CO4) Sample Output:

```
Add two Series:
0       3
1       7
2       11
3       15
4       19
dtype: int64
Subtract two Series:
0       1
1       1
2       1
3       1
4       1
dtype: int64
Multiply two Series:
0       2
1       12
2       30
3       56
4       90
dtype: int64
Divide Series1 by Series2:
0       2.000000
1       1.333333
2       1.200000
3       1.142857
4       1.111111
dtype: float64
```

19. Write a Pandas program to join the two given dataframes along row and assign all data. (CO4) Sample Output:

```
Original DataFrames:
student_idname  marks
0       S1  Danniella Fenton  200
```

```
1         S2       Ryder Storey   210
2         S3       Bryce Jensen   190
3         S4       Ed Bernal       222
4         S5       Kwame Morin 199
------------------------------------
student_idname  marks
0         S4  Scarlette Fisher     201
1         S5  Carla Williamson  200
2         S6       Dante Morse   198
3         S7   Kaiser William   219
4         S8   Madeeha Preston 201
Join the said two dataframes along rows: student_idname  marks
0         S1  Danniella Fenton  200
1         S2       Ryder Storey   210
2         S3       Bryce Jensen   190
3         S4       Ed Bernal       222
4         S5       Kwame Morin 199
0         S4  Scarlette Fisher     201
1         S5  Carla Williamson  200
2         S6       Dante Morse   198
3         S7   Kaiser William   219
4         S8  Madeeha Preston   201
```

20. Write a Pandas program to append rows to an existing DataFrame and display the combined data. (CO4) Sample Output:

```
Original DataFrames: student_id name  marks
0         S1  Danniella Fenton  200
1         S2       Ryder Storey   210
2         S3       Bryce Jensen   190
3         S4       Ed Bernal       222
4         S5       Kwame Morin 199
New Row(s) student_id
S6
name   Scarlette Fisher
marks  205
dtype: object
```

```
Combined Data:
student_idname  marks
0       S1  Danniella Fenton  200
1       S2      Ryder Storey   210
2       S3      Bryce Jensen   190
3       S4      Ed Bernal      222
4       S5      Kwame Morin    199
5       S6  Scarlette Fisher   205
```

21. Write a Pandas program

i. To identify the column(s) of a given Data Frame which have at least one missing value

ii. To count the number of missing values in each column of a given data frame

iii. Calculate the total number of missing values in a given data frame

iv. To find and replace the missing values in a given data frame

Which do not have any valuable information

v. To replace the missing values with the most frequent values present in each column of a given dataframe (CO4)

Sample Output:

```
Original Orders Data Frame:
ord_no purch_amt   sale_amt   ord_date     customer_id   salesman_id
0      70001.0     150.50     2012-10-05   3002          5002.0
1      NaN         270.65     2012-09-10   3001          5003.0
2      70002.0     65.26      NaN          3001          5001.0
3      70004.0     110.50     2012-08-17   3003          NaN
4      NaN         948.50     2012-09-10   3002          5002.0
5      70005.0     2400.60    2012-07-27   3001          5001.0
6      NaN         5760.00    2012-09-10   3001          5001.0
7      70010.0     1983.43    2012-10-10   3004          NaN
8      70003.0     2480.40    2012-10-10   3003          5003.0
9      70012.0     250.45     2012-06-27   3002          5002.0
10     NaN         75.29      2012-08-17   3001          5003.0
11     70013.0     3045.60    2012-04-25   3001          NaN
```

(i) Identify the columns which have at least one missing value:

```
ord_no True
purch_amt    False
```

```
ord_date        True
customer_id     False
salesman_id     True
dtype: bool
```

(ii) Number of missing values of the said dataframe:

```
 ord_no         4
 purch_amt      0
 ord_date       1
 customer_id    0
salesman_id     3
dtype: int64
```

(iii) Total number of missing values of the said DataFrame: 17

(iv) Replace the missing values with NaN:

```
ord_no purch_amt   sale_amt   ord_date     customer_id  salesman_id
0      70001.0     150.50     NaN          3002.0       5002.0
1      NaN         270.65     2012-09-10   3001.0       5003.0
2      70002.0     65.26      NaN          3001.0       NaN
3      70004.0     110.50     2012-08-17   3003.0       5001.0
4      NaN         948.50     2012-09-10   3002.0       NaN
5      70005.0     2400.60    2012-07-27   3001.0       5002.0
6      NaN         5760.00    2012-09-10   3001.0       5001.0
7      70010.0     NaN        2012-10-10   3004.0       NaN
8      70003.0     12.43      2012-10-10   NaN          5003.0
9      70012.0     2480.40    2012-06-27   3002.0       5002.0
10     NaN         250.45     2012-08-17   3001.0       5003.0
11     70013.0     3045.60    2012-04-25   3001.0       NaN
```

(v)

```
ord_no purch_amt   sale_amt   ord_date     customer_id  salesman_id
0      70001.0     150.50     2012-10-05   3002         5002.0
1      70001.0     270.65     2012-09-10   3001         5003.0
2      70002.0     65.26      2012-10-10   3001         5001.0
3      70004.0     110.50     2012-08-17   3003         5001.0
4      70001.0     948.50     2012-09-10   3002         5002.0
5      70005.0     2400.60    2012-07-27   3001         5001.0
6      70001.0     5760.00    2012-09-10   3001         5001.0
```

```
7     70010.0     1983.43     2012-10-10     3004     5001.0
8     70003.0     2480.40     2012-10-10     3003     5003.0
9     70012.0     250.45      2012-06-27     3002     5002.0
10    70001.0     75.29       2012-08-17     3001     5003.0
11    70013.0     3045.60     2012-04-25     3001     5001.0
```

# OPERATING SYSTEMS

COURSE CODE: 23IT4119 CREDITS 3

L T P S Sessional Marks: 40

3 0 0 0 End Exam Marks: 60

End Exam:3 Hours

Prerequisite: Knowledge in Computer Organization

Course Objectives:

- Understand Functions, Services and structure of Operating Systems. Understand processes, threads, schedulers and explanation of CPU scheduling.
- Understand issues related to Process Synchronization and focus on principles of Deadlock and related problems
- Comprehend the mechanisms used in Memory Management and Virtual Memory. Understand the concepts of File System, secondary storage management and Disk Scheduling

Course Outcomes:

After course completion, the students will be able to:

1. Illustrate the overall view of operating system structure and process management.
2. Apply appropriate synchronization techniques in handling deadlocks.
3. Interpret the issues and challenges of memory management
4. Demonstrate the concepts of file system implementation and protecting mechanisms.

**Mapping of course outcomes with program outcomes:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 2 | 3 | 1 | | | | | 1 | 2 | 3 | 3 | 3 |
| **CO2** | 3 | 1 | 2 | 2 | | | | 2 | 2 | | 1 | 1 | 1 | 3 |
| **CO3** | 3 | 2 | 2 | 1 | 2 | | | 3 | 2 | | 1 | 2 | 2 | 3 |
| **CO4** | 2 | 2 | 1 | 1 | 2 | | 1 | 2 | 1 | | | 2 | 2 | 3 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT – I 10 Periods

**Introduction:** What is an operating system? History of operating system, computer hardware, different operating systems, operating system concepts, system calls, operating system structure.

**Processes and Threads:** Processes, threads, inter-process communication, scheduling, IPC problems.

**Learning Outcomes:** At the end of this Unit the student will be able to

1. Explain Types of operating systems
2. Describe process states and process models
3. Compare processor scheduling algorithm

UNIT – II 10 Periods

**PROCESS SYNCHRONIZATION AND DEADLOCK**

**Process Synchronization:** The critical section problem, Synchronization hardware, Semaphores, Classic problems of synchronization, critical regions, Monitors.

**Deadlock:** System model, Deadlock characterization, Methods for handling deadlocks, Deadlockprevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

**Learning Outcomes:** At the end of this Unit the student will be able to

1. Describe race condition & mutual exclusion
2. Identify Deadlocks
3. Apply Deadlock recovery procedure

UNIT – III 10 Periods

**Memory Management:** No memory abstraction, memory abstraction: address spaces, virtual memory, page replacement algorithms, design issues for paging systems, implementation issues, segmentation

**File Systems:** Files, directories, file system implementation, file-system management and optimization

**Learning Outcomes:** At the end of this Unit the student will be able to

1. Describe memory management
2. Differentiate Contiguous and Non-contiguous memory
3. Differentiate physical and virtual primary memory
4. Apply file management concepts in Operating System

UNIT – IV 8 Periods

**Input-Output:** Principles of I/O hardware, Principles of I/O software, I/O software layers, disks, clocks, user interfaces: keyboard, mouse, monitor, thin clients, power management

**Case Study on Windows:** History of windows through Windows 10/11, programming windows, system structure, processes and threads in windows, memory management, caching in windows, I/O in windows, Windows NT file system, Windows power management, Security in windows

**Case Study on LINUX and ANDROID:** History of Unix and Linux, Linux Overview, Processes in Linux, Memory management in Linux, I/O in Linux, Linux file system, security in Linux. Android

**Learning Outcomes:** At the end of this Unit the student will be able to

1. Describe I/O management
2. Describe Windows 10/11 structure, process and memory management.
3. Identify advantages and disadvantages in Security in windows

UNIT – V 10 Periods

**Virtualization and Cloud:** History, requirements for virtualization, type 1 and 2 hypervisors, techniques for efficient virtualization, hypervisor microkernels, memory virtualization, I/O virtualization, Virtual appliances, virtual machines on multicore CPUs, Clouds.

**Learning Outcome:** At the end of this Unit the student will be able to

1. Describe techniques for efficient virtualization
2. Describe virtual machines on multicore CPUs

**TEXT BOOK:**

1. Silberschatz, Galvin, and Gagne, "Operating System Concepts", Sixth Edition, Wiley India Pvt Ltd, 2003.

**REFERENCE BOOKS:**

1. Andrew S. Tanenbaum, "Modern Operating Systems", Second Edition, Pearson Education,2004.
2. Gary Nutt, "Operating Systems", Third Edition, Pearson Education, 2004.
3. Harvey M. Deitel, "Operating Systems", Third Edition, Pearson Education, 2004.

CHANGE OF SYLLABUS

| **Unit No** | **Changes Incorporated** |
|---|---|
| Unit – 1 | **IPC Problems** |
| Unit – 3 | **Optimization, MS-DoS file system, UNIX V7 file system, CD ROM file system** |
| Unit – 4 | **Thin clients, Power Management, Case Studies on Windows** |
| Unit – 5 | **Virtualization and Cloud** |
| **Change of Syllabus: <20%** | |

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Operating Systems | 23IT4114 | R23 (2024) | Virtualization and Cloud | Employability |

## DATABASE MANAGEMENT SYSTEMS LAB

| | |
|---|---|
| Course Code: 23IT4215 | Credits: 1.5 |
| L T P S | Sessional Marks: 50 |
| 0 0 3 0 | End Exam Marks: 50 |

Prerequisite(s): Fundamentals of computers, familiarity of any one program language.

Course Objectives:

- To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing and implementing a DBMS.
- Perform Pl/SQL programming using concept of Cursor Management, Error Handling, Package and Triggers.
- Understand query processing and techniques involved in query optimization.

Course Outcomes:

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

1. Design and implement a database schema for a given problem-domain. Query a database using SQL commands.
2. Declare and enforce integrity constraints on a database using RDBMS and optimize the database using normalization concept.
3. Programming PL/SQL including stored procedures, stored functions, cursors, packages.

**Mapping of Course Outcomes with POs and PSO:**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 1 | 1 | 3 | 2 | | | 2 | 2 | | 2 | 2 | 1 |
| **CO2** | 3 | 2 | 1 | 1 | 3 | 2 | | | 2 | 2 | | 2 | 2 | 1 |
| **CO3** | 3 | 2 | 1 | 1 | 3 | 2 | | | 2 | 2 | | 2 | 2 | 1 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

LIST OF EXPERIMENTS:

1. Data Definition Commands, Data Manipulation Commands for inserting and deletingdata from Tables. CO-1
2. Data Manipulation Commands for updating and retrieving of data from Tables and Transaction Control statements CO-1
3. Basic functions like Numeric, String, Date, and conversion functions. CO-1
4. Database Querying – Simple queries. CO-1
5. Queries using aggregate functions, GROUP BY, and HAVING clauses. CO-2
6. Database Querying – Nested queries, Sub-queries. CO-2
7. Queries using Joins CO-2
8. Queries using Views CO-2

PROGRAMS USING PL/SQL: CO-3

9. Procedures and Functions.

10. Implicit and Explicit Cursors
11. Triggers
12. Exception Handling

**CASE STUDIES:** *Students shall form in groups at the beginning of the semester and perform at least one of the following questions by the end of the semester and submit a project.*

13. Design a Database for any real-life application using ER model and normalize it. CO-2
14. Connect the Database through any programming language CO-1
15. Build real-life database applications. CO-1
16. Query processing and optimization. CO-1

TEXT BOOKS

1. Raghurama Krishnan, Johannes Gehrke, "Data base Management Systems", 3rd Edition, TATAMcGrawHill,2014.
2. Elmasri Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Education, 2017.

REFERENCES:

1. C.J.Date, Introduction to Database Systems, 8th Edition, Pearson Education, 2003.
2. Peter Rob & Carlos Coronel, Database Systems design, Implementation, and Management, 9th Edition, Pearson Education, 2009.

CHANGE OF SYLLABUS

| **Exp. No** | **Changes Incorporated** |
|---|---|
| 13 | **Design a Database for any real-life application using ER model and normalize it** |
| 14 | **Connect the Database through any programming language** |
| 15 | **Build real-life database applications** |
| 16 | **Query optimization** |
| **Change of Syllabus: 25%** | |

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Database Management Systems Lab | 23IT4215 | R23 (2020) | DDL<br>DML<br>PL/SQL<br>CASE Studies | Employability / Skill Development |

# COMPUTER NETWORKS LAB

| | |
|---|---|
| COURSE CODE: 23IT4216 | Credits: 1.5 |
| L T P S | Sessional Marks: 50 |
| 0 0 3 0 | End Exam Marks: 50 |

Prerequisites: Cisco Packet Tracer/ C/C++/Java Programming

Course Objectives:

- The objective of this lab is to introduce students to the design issues that arise in building and using networks and to give students hands on experience with building and using network services.
- Implementation of various routing algorithms, Quality of services in networking.

Course Outcomes:

After course completion, the students will be able to:

1. Choose suitable tool to model a network and understand the protocols at various OSI reference levels.
2. Design a suitable network and simulate using a Network simulator tool.
3. Simulate the networking concepts and protocols using C/C++ also model the networks for different configuration and analyse the results

**Mapping of Course Outcomes with POs and PSO:**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 3 | 3 | 3 | 1 | | | 3 | | | | 1 | 1 |
| **CO2** | 3 | 2 | 3 | 2 | 3 | 1 | | | 3 | | | | 1 | 1 |
| **CO3** | 3 | 3 | 2 | 3 | 2 | 1 | | | 3 | | | | 1 | 1 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**Hands on experiments**

1. Preparing straight and cross cables.

2. Implementation of file and printer sharing

3. Designing and implementing Class A, B, and C Networks

4. Subnet planning and its implementation.

**Programming Experiments**

1. Network commands and network configuration commands

2. Configure hub, switch and router using packet tracing software

3. Configure a network topology using packet tracing software

4. configure wireless access points using packet tracing software

5. Configure static routing using packet tracer software

6. Configure dynamic routing using packet tracer software

7. DHCP, DNS, HTTP configuration using packet tracer software

8. Client-Server program using Java

9. Configure a Network with Virtual LANS using packet tracer software

**Text Books & References:**

1. CCNA Study guide
2. Cisco Networking Essentials by Troy McMillan · 2011, Publisher- Wiley
3. Cisco Networking Simplified By Paul L. Della Maggiora, Jim Doherty, Publisher: Cisco Press
4. Microsoft Encyclopedia of Networking By Mitch Tulloch, Microsoft Corporation, Publisher-Microsoft Press
5. Others (Web, Video, Simulation etc.)

CHANGE OF SYLLABUS

**New Lab**
**Change of Syllabus: 100%**

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Data Communication and Computer Networks Lab | **23IT4216** | R23 (2024) | All experiments | Employability |

# OPERATING SYSTEMS LAB

COURSE CODE: 23IT4217 CREDITS 3

L T P S Sessional Marks: 40

0 0 3 0 End Exam Marks: 60

End Exam:3 Hours

Prerequisite: Operating System Concepts.

Course Objectives:

- Analyze the working of an operating system, its programming interface and file system. Develop algorithms for process scheduling, memory management, page replacement algorithms and disk scheduling

Course Outcomes:

After course completion, the students will be able to:

1. Simulate operating system algorithms for CPU Scheduling and handling deadlocks
2. Implement operating system services like memory management
3. Experiment with Unix-like operating system to interact with kernel through shell scripts.

**Mapping of course outcomes with program outcomes:**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 1 | 2 | 2 | 2 | | | | 2 | | | 1 | 3 | 2 |
| **CO2** | 2 | 1 | 2 | 2 | 1 | | | | 2 | | | 3 | 3 | 3 |
| **CO3** | | | 2 | 3 | 2 | | | | 1 | | | 2 | 3 | 2 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**List of Experiments:**

1. Shell Programming: a) Unix Commands b) Vi Commands c) Unix Shell programming commands a) Concatenation of two strings b) Comparison of two strings c) Maximum of three numbers d) Fibonacci series e) Arithmetic operation using case (CO3)
2. System Calls a) Process Creation b) Executing a command c) Sleep command d)Sleep command using getpid e) Signal handling using kill k) Wait command i)top (CO3)
3. I/O System Calls a) Reading from a file b) Writing into a file c) File Creation (CO3)
4. a) Implementation of is command b)Implementation of grep command (CO3)
5. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. Print avg.waiting time and turnaround time. (CO1)
6. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. Print avg. waiting time and turnaround time. (CO1)
7. a) Implement Bankers algorithm for Dead Lock Avoidance b) Implement an Algorithm forDead Lock Detection(CO1)
8. Developing Application using Inter Process communication (using shared memory, pipesor message queues) (C01)
9. Producer-Consumer Problem using Semaphore (CO1)

10. Write a c program to implement Paging technique for memory management (CO2)
11. Write a c program for implementing segmentation in memory using array (CO2)
12. Write a program to simulate page replacement algorithm for a) FIFO b) LRU c) LFU (CO2)
13. Implement any file allocation technique (Linked, Indexed or Contiguous) (CO3)
14. Linux system administration a. Becoming super user b. Temporarily changing user identity with su command c. Using graphical administrative tools d. Administrative commands e. Administrative configuration files(CO4)
15. Setting up Network File System. (CO3)
16. Firewall and Security Configuration(CO3)

**TEXT BOOKS:**

1. Sumitabha Das, Unix and Shell Programming, Tata Mcgraw Hill Publishing CoLtd
2. W.Richard Stevens, Stephen A.Rago , Advanced programming in the UNIX environment",3rd Edition Pearson education.
3. Silberschatz, Galvin, and Gagne, “Operating System Concepts”, Sixth Edition, Wiley IndiaPvt Ltd, 2003.

CHANGE OF SYLLABUS

| **Exp. No** | **Changes Incorporated** |
|---|---|
| 2 | **System Calls** |
| 13 | **Implement any file allocation technique** |
| 14 | **Linux system administration** |
| 15 | **Setting up Network File System** |
| 16 | **Firewall and Security Configuration** |
| **Change of Syllabus: 25%** | |

**1.1.3 of NAAC**

| **Name of the Course** | **Course Code** | **Year of Introduction** | **Activities/Content with a direct bearing on Employability/ Entrepreneurship/ Skill development** | **Mapping with Employability/Skill development/Entrepreneurship** |
|---|---|---|---|---|
| Operating Systems Lab | 23IT4213 | R23 (2024) | Complete Lab | Employability / Skill Development |

# NUMERICAL ABILITY & PROFESSIONAL COMMUNICATION SKILLS

COURSE CODE: 23CR9102 CREDITS 1

L T P S Sessional Marks: 100

2 2 0 0

Prerequisite: Knowledge of LSRW Skills, Basic Maths

Course Outcomes:

After course completion, the students will be able to:

1. Comprehend the essentiality of LSRW skills in paper presentations, seminars, workshops, conferences etc. with teams. (L2), To solve different types of questions based on vocabulary, structure, grammar and verbal reasoning
2. Attain the knowledge of soft skills in various conditions(L3), Solve questions based on sentence completion and fill in the blanks
3. Explore diverse fields through English (L4), To solve different types of questions based on vocabulary, structure, grammar and verbal reasoning
4. Use their logical thinking and analytical abilities to solve Quantitative aptitude questions from company specific and other competitive tests.
5. Solve questions related to Time and distance and time and work etc. from company specific and other competitive tests.

**Mapping of course outcomes with program outcomes:**

| | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO 10 | PO 11 | PO 12 | PSO 1 | PSO 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | | | | | | | | | 2 | 2 | | 2 | | |
| **CO2** | | | | | | | | | 2 | 2 | | 2 | | |
| **CO3** | | | | | | | | | 2 | 2 | | 2 | | |
| **CO4** | 2 | | | | | | | | | | | | | |
| **CO5** | 2 | | | | | | | | | | | | | |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

## PART-A: Professional Communication skills

UINT I

Abstract Preparation – Noticing Key Words –Literature Survey – Using Academic Verbs

Verbal Ability : Sentence correction

UNIT II

Organizational Skills – Time Management – IELTS Test Papers Exercises Verbal Ability : sentence completion

UNIT III

Meeting Skills – Arranging a Meeting – Prior to Meeting, During Meeting and After Meeting Process – Note Making – Note Taking

Verbal Ability : Error Identification

UNIT IV

Analogy – Origin of the Words – Eponyms – MNCs Question Papers

Verbal Ability : vocabulary

## PART-B : Numerical Ability

UNIT-I

Numerical computation: Applications based on Numbers –Classification of Number System, Prime and Composite, Even and Odd Numbers, Divisibility Rule, Remainder Theorem, Finding Highest power, LCM &HCF

UNIT-II

Numerical estimation – I: Averages, Ratio Proportion, Application of Ratios (Ages), Partnerships, Shares and dividends,

UNIT-III

Numerical estimation – II: Percentages and its Applications, Profit Loss and Discount, Simple interest and Compound Interest

UNIT-IV

Time and work, Application of Time-work (Pipes & Cisterns), Time and Distance, circular Tracking, concept of Boats & steams.

UNIT-V

Mixtures and allegations, application of percentage and Ratios and Averages in Mixtures

# ANGULAR DEVELOPER

COURSE CODE: 23IT9302 CREDITS 1

| L | T | P | S |
|---|---|---|---|
| 0 | 0 | 0 | 2 |

https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012564311410999296377_shared/overview

# ENTREPRENEURSHIP DEVELOPMENT & IPR

COURSE CODE: 23MC0104 Credits: 0

L T P S Sessional Marks: 100

3 0 0 0

Course Objectives:

- The course has been designed to develop the skills of entrepreneurship & to encourage the students to become an entrepreneur and to impart the basics of Intellectual Property Rights.

Course Outcomes:

After course completion, the students will be able to:

1. Apply various theories for the entrepreneurship development ecosystem in Indian context.
2. Demonstrate the ways in which entrepreneurs perceive opportunity, manage risk, organize resources and add value.
3. Identify various schemes supporting entrepreneurship.
4. Recognize the importance of IP and outline concepts of Intellectual Property Rights.
5. Identify the significance of practice and procedure of Patents.

**Mapping of Course Outcomes with POs and PSO:**

| | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | | | | | | 1 | | 2 | 1 | 1 | 2 | 1 | | 1 |
| **CO2** | | | | | | 1 | | 2 | 1 | 1 | 2 | 1 | | 1 |
| **CO3** | | | | | | 1 | | 2 | 1 | 1 | 2 | 1 | | 1 |
| **CO4** | | | | | | 1 | | 2 | 1 | 1 | 2 | 1 | | 1 |
| **CO5** | | | | | | 1 | | 2 | 1 | 1 | 2 | 1 | | 1 |

Correlation levels 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

UNIT – I 10 Periods

Introduction to Entrepreneurship: Entrepreneurship- Concept, Nature, Functions and Importance; Entrepreneurs Characteristics, Types and Motivation; Entrepreneurial process; Enterprise- Definition and Classification (MSME Micro, Small & Medium Enterprises).

UNIT – II 10 Periods

Entrepreneurial Journey: Creativity and Innovation, Recognizing opportunities and Generating ideas, Feasibility analysis, Industry and Competitor analysis, developing effective business model.

Class Activity: Idea generation by students.

UNIT – III 10 Periods

Institutional Support to Entrepreneurs: Need for Institutional support different Government & Non-Government institutions to support Entrepreneurs like, NSIC, SIDO, SSIB, SSIDC, SISIs, DTICs, industrial Estates, Specialized Institutions.

UNIT – IV 10 Periods

INTRODUCTION TO IPR: Meaning of property, Origin, Nature, Meaning of Intellectual Property Rights – Introduction to TRIPS and WTO. – Kinds of Intellectual property rights—Copy Right, Patent, Trade Mark, Trade Secret and trade dress, Design, Layout Design, Geographical Indication, Plant Varieties and Traditional Knowledge.

UNIT – V 10 Periods

Patent system in India: Patents Act 1970 & Patent system in India; Patentability; Process, & product patent; filing of the patent, patent specification, patent claims, Patent opposition, & revocation, infringement, compulsory licensing, Patent Cooperation Treaty, Patent search, and patent database.

TEXT BOOKS:

1. Robert D. Hisrich, Mathew J. Manimala, Michael P Peters and Dean A. Shepherd, “Entrepreneurship”, 9th Edition, Tata Mc-graw Hill Publishing Co.ltd.-new Delhi, 2014.
2. Bruce R. Barringer and R. Duane Ireland, “Entrepreneurship”, 4th Edition, Pearson Publications, New Delhi, 2011
3. N.K. Acharya, Text book on intellectual Property Rights, Asha Law House New Delhi, New Edition, 2001.

REFERENCE BOOKS:

1. Narayanan, V. K., Managing technology and innovation for competitive advantage, first edition, Pearson education, New Delhi, (2006)
2. Idris, K. (2003), Intellectual property: a power tool for economic growth, second edition, WIPO publication no. 888, Switzerland
3. Bosworth D. & Webster E, The Management of Intellectual Property, Edward Elgar.

WEB RESOURCES:

1. https://ebooks.inflibnet.ac.in/hsp15/chapter/intellectual-property-management/
2. https://onlinecourses.nptel.ac.in/noc20_hs66/preview