DEPARTMENT OF COMPUTTER SCIENCE & ENGINEERING
JNTUH COLLEGE OF ENGINEERING HYDERABAD (AUTONOMOUS)
Kukatpally, Hyderabad - 500 085, Telangana, India

MASTER OF COMPUTER APPLICATIONS (MCA)

COURSE STRUCTURE & SYLLABUS (R20)
Effective from Academic Year 2020-21 Admitted Batch

I YEAR I SEMESTER

Category	Course Title	L	T	P	Credits
Core Course - I	Mathematical Foundations of Computer Science	3	0	0	3
Core Course - II	C and Data Structures	3	0	0	3
Core Course – III	Python Programming (with Lab)	3	0	2*	4
Core Course – IV	Computer Organization	3	0	0	3
Core Course- V	Database Management Systems	3	0	0	3
Core Course – VI	Accounting and Financial Management	3	0	0	3
Laboratory – I	C and Data Structures Lab	0	0	2	1
Laboratory – II	Database Management Systems Lab	0	0	2	1
	Total Credits	**18**	**0**	**6**	**21**

*Only internal assessment

I YEAR II SEMESTER

Category	Course Title	L	T	P	Credits
Core Course - I	Java Programming	3	0	0	3
Core Course - II	Operating Systems (with Lab)	3	0	2*	4
Core Course – III	Computer Networks	3	0	0	3
Core Course – IV	Data Mining (with R Programming Lab)	3	0	2*	4
Core Course – V	Software Engineering (with Lab)	3	0	2*	4
Core Course – VI	Mobile Application Development	3	0	0	3
Laboratory – I	Mobile Application Development Lab	0	0	2	1
Laboratory – II	Java Programming Lab	0	0	2	1
Audit Course – I (Mandatory)	Cyber Security	**2**	**0**	**0**	**0**
	Total Credits	**20**	**0**	**10**	**23**

* Only internal assessment

II YEAR I SEMESTER (III SEMESTER)

Category	Course Title	L	T	P	Credits
Core Course - I	Network Administration (with Lab)	3	0	2*	4
Core Course - II	Web Technologies	3	0	0	3
Core Course – III	Internet of Things (with Lab)	3	0	2*	4
Professional Elective - I	Cloud Applications Information Retrieval Systems Information Systems Audit and Control Design and Analysis of Algorithms	3	0	0	3
Professional Elective - II	Mobile Computing Software Testing Methodologies Biometrics e-Commerce	3	0	0	3
Laboratory – I	Web Technologies Lab	0	0	2	1
Project	Project Phase I	0	0	4	2
Audit Course – II (Mandatory)	Artificial Intelligence	**2**	**0**	**0**	**0**
	Total Credits	**17**	**0**	**10**	**20**

*Only internal assessment

II YEAR II SEMESTER (IV SEMESTER)

Category	Course Title	L	T	P	Credits
Professional Elective - III	Machine Learning Blockchain Technology Big Data Analytics Virtual and Augmented Reality	3	0	0	3
Open Elective - I	Optimization Techniques Cyber Laws Management Information Systems Entrepreneurship	3	0	0	3
Seminar	Seminar	0	0	4	2
Project	Project Phase II	0	0	16	8
	Total Credits	**6**	**0**	**20**	**16**

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

MCA I Year I Sem	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Prerequisites:

1. An understanding of Math in general is sufficient.

Objectives:

1. Introduces the elementary discrete mathematics for computer science and engineering.
2. Topics include formal logic notation, methods of proof, induction, sets, relations, graph theory, permutations and combinations, counting principles; recurrence relations and generating functions

Outcomes:

1. Demonstrate the ability to understand and construct precise mathematical proofs
2. Demonstrate the ability to use logic and set theory to formulate precise statements
3. Acquire the knowledge to analyse and solve counting problems on finite and discrete structures
4. Demonstrate the ability to describe and manipulate sequences
5. Demonstrate the ability to apply graph theory in solving computing problems

UNIT - I

The Foundations Logic and Proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalence, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.

UNIT - II

Basic Structures, Sets, Functions, Sequences, Sums, Matrices and Relations: Sets, Functions, Sequences & Summations, Cardinality of Sets and Matrices Relations, Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.

UNIT - III

Algorithms, Induction and Recursion: Algorithms, The Growth of Functions, Complexity of Algorithms.

Induction and Recursion: Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness.

UNIT - IV

Discrete Probability and Advanced Counting Techniques: An Introduction to Discrete Probability, Probability Theory, Bayes' Theorem, Expected Value and Variance.

Advanced Counting Techniques: Recurrence Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

UNIT - V

Graphs: Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.

Trees: Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.

TEXTBOOKS

1. Discrete Mathematics and Its Applications with Combinatorics and Graph Theory- Kenneth H Rosen, 7th Edition, TMH.

REFERENCES

1. Discrete Mathematical Structures with Applications to Computer Science-J.P. Tremblay and R. Manohar, TMH,
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe L. Mott, Abraham Kandel, Teodore P. Baker, 2nd ed., Pearson Education.
3. Discrete Mathematics- Richard Johnsonbaugh, 7th ed., Pearson Education.
4. Discrete Mathematics with Graph Theory- Edgar G. Goodaire, Michael M. Parmenter.
5. Discrete and Combinatorial Mathematics - an applied introduction: Ralph.P. Grimald, 5th edition, Pearson Education.

C AND DATA STRUCTURES

MCA I Year I Sem

L	T	P	C
3	0	0	3

Prerequisites:

1. Requires analytical skills and logical reasoning.

Objectives:

1. Understand and Learn algorithms, Pseudo code and Flow chart.
2. Acquire the knowledge of Programming Environment
3. Learn basics of computer programming
4. Gain the knowledge of C Language
5. Acquire the skills in applying various concepts of C
6. Provides hands-on experience with the concepts of programming in C

Outcomes:

1. Identify computational tasks to apply for writing programming
2. Identify the right concept in C to apply based on the requirement of the problem
3. Develop solutions for computational problems and real-life applications through programming

UNIT - I

Introduction to Computers: Computer Systems, Computing Environments, Computer Languages, Creating and running programs, Software Development Method, Algorithms, Pseudo code, flow charts, applying the software development method.

Introduction to C Language: Background, Simple C programs, Identifiers, Basic data types, Variables, Constants, Input / Output, Operators. Expressions, Precedence and Associatively, Expression Evaluation, Type conversions, Bit wise operators, Statements, Simple C Programming examples.

UNIT - II

Statements: if and switch statements, Repetition statements – while, for, do-while statements, Loop examples, other statements related to looping – break, continue, go to, Simple C Programming examples.

Designing Structured Programs: Functions, basics, user defined functions, inter function communication, Scope, Storage classes-auto, register, static, extern, scope rules, type qualifiers, recursion- recursive functions,

Arrays: Concepts, using arrays in C, inter function communication, array applications, two–dimensional arrays, multidimensional arrays, C program examples. Concepts,

UNIT - III

Pointers and Strings: Introduction (Basic Concepts), Pointers for inter function communication, pointers to pointers, compatibility, memory allocation functions, array of pointers, programming applications, pointers to void, pointers to functions, command –line arguments. Pre-processor commands, example C programs.

C Strings, String Input/ Output functions, arrays of strings, string manipulation functions, string / data conversion, C program examples.

UNIT-IV:

Derived Types: Structures – Declaration, definition and initialization of structures, accessing structures, nested structures, arrays of structures, structures and functions, pointers to structures, self referential structures, unions, typedef, bit fields, enumerated types, C programming examples.

Input and Output: Concept of a file, streams, standard input / output functions, formatted input / output functions, text files and binary files, file input / output operations, file status functions (error handling), C program examples.

UNIT - V

Sorting and Searching: selection sort, bubble sort, insertion sort, linear and binary search methods. **Data Structures:** Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks-Operations, array and linked representations of stacks, stack applications, Queues-operations, array and linked representations.

TEXTBOOKS

1. C Programming & Data Structures, B.A.Forouzan and R.F. Gilberg, 3rd Edition, Cengage Learning.
2. The C Programming Language, B.W. Kernighan and Dennis M.Ritchie, PHI/Pearson Education
3. Head First C, David Griffiths, Dawn Griffiths, O'reilly Publications 2012.

REFERENCES

1. C for Engineers and Scientists, H.Cheng, Mc.Graw-Hill International Edition
2. C: The Complete Reference by Herbert Schildt, 4th Edition, TMH
3. C Programming & Data Structures, P. Dey, M Ghosh R Thereja, Oxford University Press
4. Learn to Code with C : The MagPi Essentials by Simon Long
5. Problem Solving and Program Design in C, J.R. Hanly and E.B. Koffman, 5th Edition, Pearson Education.
6. Let Us C: Authentic Guide to C Programming Language by Yashvanth P. Kanetkar

PYTHON PROGRAMMING

MCA I Year I Sem

L	T	P	C
3	0	2*	4

Prerequisites:

1. Requires analytical skills and logical reasoning.

Objectives:

1. To learn syntax and semantics and create functions in Python.
2. To handle strings and files in Python.
3. To understand lists, dictionaries and regular expressions in Python.
4. To implement object oriented programming concepts in Python.
5. To build web services and introduce network and database programming in Python.

Outcomes:

1. Demonstrate proficiency in handling Strings and File Systems.
2. Create, run and manipulate Python programs using core data structures like Lists, Dictionaries and use Regular Expressions.
3. Interpret the concepts of Object-Oriented Programming as used in Python.
4. Implement applications related to Network Programming, Web Services and Databases in Python.

UNIT - I

Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types

Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules

Sequences - Strings, Lists, and Tuples, Mapping and Set Types

UNIT - II

Files: File Objects, File Built-in Function, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules, Related Modules

Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management,

Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, Creating Exceptions, Exceptions and the sys Module, Related Modules, Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules

UNIT - III

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python

Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules

UNIT - IV

GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs

Web Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application, Advanced CGI, Web (HTTP) Servers

UNIT - V

Database Programming: Introduction, Python Database Application Programmer's Interface (DB-API), Object Relational Managers (ORMs), Related Modules

TEXTBOOKS

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.

REFERENCES

1. Python for Programmers, Paul Deitel, Harvey Deitel, Pearson.
2. Think Python, Allen Downey, Green Tea Press
3. Introduction to Python, Kenneth A. Lambert, Cengage
4. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
5. Learning Python, Mark Lutz, O'Reilly

PYTHON PROGRAMMING LAB

MCA I Year I Sem

Objectives:

1. To be able to introduce core programming basics and program design with functions using Python programming language.
2. To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
3. To understand the high-performance programs designed to strengthen the practical expertise.

Outcomes:

1. Acquire the understanding of the basic concepts, scripting and the contributions of scripting language
2. Demonstrate the ability to explore python especially the object oriented concepts, and the built in objects of Python.
3. Create practical and contemporary applications such as TCP/IP network programming, Web applications, discrete event simulations

List of Experiments:

1. (a) Write a Python program to sum all the items in a list.

   (b) Write a Python program to remove duplicates from a list
2. (a) Write a Python program to sort (ascending and descending) a dictionary by value

   (b) Write a Python script to add a key to a dictionary
3. (a) Write a Python program to create a tuple

   (b) Write a Python program to convert a tuple to a string
4. (a) Write a Python program to count the number of characters (character frequency) in a string

   (b) Write a Python program to count and display the vowels of a given text
5. (a) Write a Python program to read first n lines of a file

   (b) Write a Python program to append text to a file and display the text.
6. Python program to demonstrate basic operations on single array.
7. Write a Python program that matches a string that has a followed by three 'b'.
8. Write a Python program Create a thread class to print the date
9. Write a program to demonstrate multithreading in python
10. Write a Python GUI program to import Tkinter package and create a window and set its title

11. Write a Python GUI program to create a label and change the label font style
12. Write a Python GUI program to create a window and disable to resize the window using tkinter module
13. Write a Python program to demonstrate Client Server programs
14. Write a Python program to create a SQLite database and connect with the database and print the version of the SQLite database
15. Write a Python program to connect a database and create a SQLite table within the database

COMPUTER ORGANIZATION

MCA I Year I Sem

L	T	P	C
3	0	0	3

Objectives

1. The purpose of the course is to introduce principles of computer organization and the basic architectural concepts.
2. It begins with basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations.
3. Topics include computer arithmetic, instruction set design, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors

Outcomes

1. Understand the basics of instructions sets and their impact on processor design.
2. Demonstrate an understanding of the design of the functional units of a digital computer system.
3. Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory.
4. Design a pipeline for consistent execution of instructions with minimum hazards.
5. Recognize and manipulate representations of numbers stored in digital computers

UNIT - I

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture, Basics organization of computer.

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT - II

Micro Programmed Control: Control memory, Address sequencing, micro program example, design of control unit.

Central Processing Unit: Introduction General Register Organization, Stack organization Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT - III

Data Representation: Data types, Complements, Fixed Point Representation, Conversion of Fractions Floating Point Representation.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic Unit, Decimal Arithmetic operations.

UNIT - IV

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

UNIT - V

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Arrey Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Symmetric multiprocessor, Cache Coherence.

TEXTBOOKS

1. Computer System Architecture – M. Moris Mano, Revised 3rd Edition, Pearson.

REFERENCES

1. Computer Organization – Car Hamacher, ZvonksVranesic, SafeaZaky, 5th Edition, McGraw Hill.
2. Computer Organization and Architecture – William Stallings 6th Edition, Pearson/PHI.
3. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition PHI/Pearson.

DATABASE MANAGEMENT SYSTEMS

MCA I Year I Sem

L	T	P	C
3	0	0	3

Prerequisites

1. A course on "Data Structures"

Objectives

1. To understand the basic concepts and the applications of database systems.
2. To master the basics of SQL and construct queries using SQL.
3. Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

Outcomes

1. Gain knowledge of fundamentals of DBMS, database design and normal forms
2. Master the basics of SQL for retrieval and management of data.
3. Be acquainted with the basics of transaction processing and concurrency control.
4. Acquire familiarity with database storage structures and access techniques

UNIT - I

Database System Applications: A Historical Perspective, File Systems versus a DBMS**,** the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS
Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT - II

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical data base design, introduction to views, destroying/altering tables and views.
Relational Algebra**,** Tuple relational Calculus, Domain relational calculus.

UNIT - III

SQL: Queries, Constraints, Triggers: Basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active data bases.
Schema refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multi-valued dependencies, FOURTH normal form, FIFTH normal form.

UNIT - IV

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions.

UNIT - V

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and

Secondary Indexes, Index data Structures, Hash Based Indexing, Tree base Indexing, Comparison of File Organizations, Indexes and Performance Tuning, Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index Structure.

TEXTBOOKS

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, 3rd Edition, Tata McGraw Hill
2. Database System Concepts, Silberschatz, Korth, 7th edition, McGraw hill.

REFERENCES

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate Pearson Education
3. Introduction to Database Systems, C.J.Date Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL,Shah,PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

ACCOUNTING AND FINANCIAL MANAGEMENT

MCA I Year I Sem	L	T	P	C
	3	0	0	3

Objectives:

1. To learn Financial Management and Accounting
2. To learn different types of costing

Outcomes:

1. Prepare balance sheets of budget.
2. Get the skill to manage finances of a firm/company

UNIT - I

Introduction to Accounting: Principles, concepts, conventions, double entry system of accounting, introduction of basic books of accounts ledgers.

Preparation of Trial Balance: Final accounts - company final accounts.

UNIT - II

Financial Management: Meaning and scope, role, objectives of time value of money - over vitalization - under capitalization - profit maximization - wealth maximization - EPS maximization.

Ratio Analysis: Advantages - limitations - Fund flow analysis - meaning, importance, preparation and interpretation of Funds flow and cash flow statements-statement.

UNIT - III

Costing: Nature and importance and basic principles, Absorption costing vs. marginal costing - Financial accounting vs. cost accounting vs. management accounting.

Marginal Costing and Break-even Analysis: Nature, scope and importance - practical applications of marginal costing, limitations and importance of cost - volume, profit analysis.

UNIT - IV

Standard Costing and Budgeting: Nature, scope and computation and analysis - materials variance, labor variance and sales variance - budgeting - cash budget, sales budget - flexible Budgets, master budgets.

UNIT - V

Introduction to Computerized Accounting System: coding logic and codes, master files, transaction files, introduction documents used for data collection, processing of different files and Outputs obtained.

TEXTBOOKS

1. Van Horne, James, C: Financial Management and Policy, 12th Edition, Pearson Education.
2. Financial Accounting, S.N.Maheshwari, Sultan Chand Company.
3. Financial Management, S.N.Maheshwari, Sultan Chand Company

C AND DATA STRUCTURES LAB

MCA I Year I Sem

L	T	P	C
0	0	2	1

Co-requisites:

1. "C and Data Structures" course

Objectives:

1. It covers various concepts of C programming language
2. It introduces searching and sorting algorithms
3. It provides an understanding of data structures such as stacks and queues.

Course Outcomes:

1. Develop C programs for computing and real life applications using basic elements like control statements, arrays, functions, pointers and strings, and data structures like stacks, queues and linked lists.
2. Implement searching and sorting algorithms

Week 1:

1. Write a C program to find the sum of individual digits of a positive integer.
2. Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
3. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
4. Write a C program to find the roots of a quadratic equation.

Week 2:

5. Write a C program to find the factorial of a given integer.
6. Write a C program to find the GCD (greatest common divisor) of two given integers.
7. Write a C program to solve Towers of Hanoi problem.
8. Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,*, /, % and use Switch Statement)

Week 3:

9. Write a C program to find both the largest and smallest number in a list of integers.
10. Write a C program that uses functions to perform the following:

    i) Addition of Two Matrices ii) Multiplication of Two Matrices

Week 4:

11. Write a C program that uses functions to perform the following operations:

i) To insert a sub-string in to a given main string from a given position.

ii) To delete n Characters from a given position in a given string.

12. Write a C program to determine if the given string is a palindrome or not
13. Write a C program that displays the position or index in the string S where the string T begins, or – 1if S doesn't contain T.
14. Write a C program to count the lines, words and characters in a given text.

Week 5:

15. Write a C program to generate Pascal's triangle.
16. Write a C program to construct a pyramid of numbers.
17. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: $1+x+x^2+x^3+ +x^n$

Week 6:

18. 2'scomplement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C program to find the 2's complement of a binary number.
19. Write a C program to convert a Roman numeral to its decimal equivalent.

Week 7:

20. Write a C program that uses functions to perform the following operations:

iii) Reading a complex number

iv) Writing a complex number

v) Addition of two complex numbers

vi) Multiplication of two complex numbers (Note: represent complex number using a structure.)

Week 8:

21. i) Write a C program which copies one file to another.

ii) Write a C program to reverse the first n characters in a file. (Note: The file name and n are specified on the command line.)

22. i) Write a C program to display the contents of a file.

ii) Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file)

Week 9:

23. Write a C program that uses functions to perform the following operations on singly linked list:

i) Creation ii)Insertion iii)Deletion iv)Traversal

Week 10:

24. Write C programs that implement stack (its operations) using

    i) Arrays ii)Pointers

25. Write C programs that implement Queue (its operations) using

    i) Arrays ii)Pointers

Week 11:

26. Write a C program that implements the following sorting methods to sort a given list of integers in ascending order

    i) Bubble sort ii) Selection sort

Week 12:

27. Write C programs that use both recursive and non recursive functions to perform the following searching operations for a Key value in a given list of integers:

    i) Linear search ii) Binary search

TEXTBOOKS

1. C Programming & Data Structures, B.A.Forouzan and R.F. Gilberg, 3rd Edition, Cengage Learning.
2. The C Programming Language, B.W. Kernighan and Dennis M.Ritchie, PHI/Pearson Education
3. Head First C, David Griffiths, Dawn Griffiths, O'reilly Publications 2012.

REFERENCES

1. C for Engineers and Scientists, H.Cheng, Mc.Graw-Hill International Edition
2. C: The Complete Reference by Herbert Schildt, 4th Edition, TMH
3. C Programming & Data Structures, P. Dey, M Ghosh R Thereja, Oxford University Press
4. Learn to Code with C : The MagPi Essentials by Simon Long
5. Problem Solving and Program Design in C, J.R. Hanly and E.B. Koffman, 5th Edition, Pearson Education.
6. Let Us C: Authentic Guide to C Programming Language by Yashvanth P. Kanetkar

DATABASE MANAGEMENT SYSTEMS LAB

MCA I Year I Sem

L	T	P	C
0	0	2	1

Co-requisites

1. Co-requisite of course "Database Management Systems"

Objectives

1. Introduce ER data model, database design and normalization
2. Learn SQL basics for data definition and data manipulation

Outcomes

1. Design database schema for a given application and apply normalization
2. Acquire skills in using SQL commands for data definition and data manipulation. Develop solutions for database applications using procedures, cursors and triggers

List of Experiments:-

1. Conceptual design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

Textbooks

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, 3rd Edition Tata McGraw Hill
2. Database System Concepts, Silberschatz, Korth,, 7th edition, McGrawhill.

References

1. Database Systems Design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, ElmasriNavrate Pearson Education
3. Introduction to Database Systems, C.J.Date Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL,Shah,PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

JAVA PROGRAMMING

MCA I Year II Sem

L	T	P	C
3	0	0	3

Prerequisites:

1. A course on "Computer Programming & Data Structures"

Objectives:

1. Introduces object oriented programming concepts using the Java language.
2. Introduces the principles of inheritance and polymorphism; and demonstrates how they relate to the design of abstract classes
3. Introduces the implementation of packages and interfaces
4. Introduces exception handling, event handling and multithreading
5. Introduces the design of Graphical User Interface using applets and swings

Outcomes:

1. Develop applications for a range of problems using object-oriented programming techniques
2. Design simple Graphical User Interface applications

UNIT - I

Object Oriented Thinking and Java Basics: Need for OOP paradigm, summary of OOP concepts, History of Java, Java buzzwords, data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, using final with variables, garbage collection, overloading methods and constructors, recursion, nested and inner classes, exploring string class.

UNIT - II

Inheritance: Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance, member access rules, super uses, using final with inheritance and methods, polymorphism- method overriding, abstract classes, the Object class.

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

UNIT - III

Interfaces: Differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

Exception Handling: Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. String handling,

UNIT - IV

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. Handling menus, graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

Swing: Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons –

The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

UNIT - V

Applets: Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

Multithreading: Differences between multi-threading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter-thread communication, thread groups.

TEXTBOOKS

1. Java the complete reference, Herbert schildt, 7th editon, TMH.
2. Understanding OOP with Java, updated edition, T. Budd, pearson edition.

REFERENCES

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John wiley& sons.
2. An Introduction to OOP, T. Budd, 3rd edition, pearson education.
3. Introduction to Java programming, Y. Daniel Liang, pearson education.
4. An introduction to Java programming and object oriented application development, R.A. Johnson- Thomson.
5. Core Java 2, Vol 1, Fundamentals, Cay.S.Horstmann and Gary
6. Cornell, 8th Edition, Pearson Education.
7. Core Java 2, Vol 2, Advanced Features, Cay.S.Horstmann and Gary Cornell, 8th Edition, Pearson Education
8. Object Oriented Programming with Java, R.Buyya,S.T.Selvi,X.Chu,TMH.
9. Java and Object Orientation, an introduction, John Hunt, 2nd edition, Springer.
10. Maurach's Beginning Java2 JDK 5 , SPD.
11. Programming and Problem Solving with Java, JM Slack, B S Publications.

OPERATING SYSTEMS

MCA I Year II Sem

L	T	P	C
3	0	2*	4

Prerequisites:

1. Computer Programming and Data Structures
2. A good knowledge of C programming language
3. Fundamental concepts of Computer Organization and Architecture.

Objectives:

1. To learn various concepts and issues related to Operating systems
2. To learn the underlying issues in the design of an operating system.

Outcomes:

1. Appreciate all features of an operating system.
2. Recognize and address various issues of an operating system.
3. Gains knowledge of interaction among Programming Languages, operating systems and their architecture.

UNIT - I

Operating System Introduction: Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls.

UNIT - II

Process and CPU Scheduling: Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads, and Interposes Communication, Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling.

System call interface for process management: fork, exit, wait, waitpid, exec

UNIT - III

Deadlocks: System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

Process Management and Synchronization: The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors.

Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT - IV

Memory Management and Virtual Memory: Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

UNIT - V

File System Interface and Operations: Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

TEXTBOOKS

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley
2. Advanced programming in the Unix Environment, W.R.Stevens, Pearson education.

REFERENCES

1. Operating Systems – Internals and Design Principles Stallings, 5th Edition–2005, Pearson Education/PHI
2. Operating System A Design Approach- Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum 2nd edition, Pearson/PHI
4. Unix programming environment, Kernighan and Pike, PHI. / Pearson Education
5. Unix Internals The New Frontiers, U.Vahalia, Pearson Education.

OPERATING SYSTEMS LAB

MCA I Year II Sem

Prerequisites:

1. A course on "Computer Organization and Architecture".

Co-requisite:

1. A course on "Operating Systems".

Objectives:

1. To provide an understanding of the design aspects of operating system concepts through simulation
2. Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

Outcomes:

1. Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
2. Able to implement C programs using Unix system calls

List of Experiments:

1. Write C programs to simulate the following CPU Scheduling algorithms
   a) FCFS b) SJF c) Round Robin d) priority
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following IPC mechanisms
   a) Pipes b) FIFOs c) Message Queues d) Shared Memory
6. Write C programs to simulate the following memory management techniques
   a) Paging b) Segmentation

TEXT BOOKS

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley
2. Advanced programming in the Unix environment, W.R.Stevens, *Pearson* education.
3. Operating system concepts : Milan Milenkovik (TMH Publication)

REFERENCE BOOKS

1. Operating Systems – Internals and Design Principles, William Stallings, Fifth Edition–

2005, Pearson Education/PHI
2. Operating System - A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition, Pearson/PHI
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education

COMPUTER NETWORKS

MCA I Year II Sem

L	T	P	C
3	0	0	3

Objectives

1. The objective of the course is to equip the students with a general overview of the concepts and fundamentals of computer networks.
2. Familiarize the students with the standard models that use the layered approach for facilitating communication between machines in a network
3. The protocols that support communication in computer network standards.

Outcomes

1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

UNIT - I

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.

Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.

UNIT - II

Data link layer: Design issues, framing, Error detection and correction.

Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.

Sliding Window protocols: one-bit sliding window protocol, a protocol using Go-Back-N, a protocol using Selective Repeat, Example data link protocols.

Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT - III

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.

UNIT - IV

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT - V

Application Layer: Domain name system, SNMP, Electronic Mail, World Wide Web, HTTP, Streaming audio and video.

TEXTBOOK

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education

REFERENCES

1. DataCommunicationsandNetworking–BehrouzA.Forouzan.5th Edition, TMH.
2. Computer Networking: Top-down Approach – Kurose and Ross, 6th Edition, Pearson Education

DATA MINING

MCA I Year II Sem	L	T	P	C
	3	0	2*	4

Prerequisites:

1. A course on "Database Management Systems"

Course Objectives:

1. Introduce the concepts, principles and techniques related to Data Mining

Course Outcomes:

1. Understand the strengths and limitations of various data mining techniques like classification, association rule mining, cluster analysis and outlier detection.

UNIT-I

Introduction- What is Data Mining? What Kinds of Data Can Be Mined? What Kinds of Patterns Can Be Mined? Which Technologies are Used? Which Kinds of Applications are Targeted? Major Issues in Data Mining

Data Preprocessing- Overview, Data Cleaning-Missing Values, Noisy Data, Data Reduction-Attribute Subset Selection, Data Transformation and Data Discretization-Data Transformation by Normalization, Discretization by Binning

UNIT-II

Association Rule Mining- Market Basket Analysis, Frequent Itemsets, Closed Itemsets and Association Rules, Apriori Algoritm, Generating Association Rules from Frequent Itemsets

Classification-Basic Concepts, Decision Tree Induction, Bayes Classification Methods, k-Nearest Neighbor Classifiers, Rule-Based Classification, Model Evaluation and Selection, Techniques to Improve Classification Accuracy

UNIT-III

Cluster Analysis- What is Cluster Analysis, Requirements for Cluster Analysis, Overview of Basic Clustering Methods**,** k-Means, k-Medoids, Agglomerative vs. Divisive Hierarchical Clustering

UNIT-IV

Outlier Detection-Outliers and Outlier Analysis, Outlier Detection Methods, Statistical Approaches, Proximity-Based Approaches, Clustering Based Approaches and Classification-Based Approaches

UNIT-V

Text Mining: Text Data Analysis and Information Retrieval, Dimensionality Reduction for Text, Text Mining Approaches

Mining the WWW: Mining the Web Page Layout Structure, Mining the Web's Link Structures, Mining Multimedia Data on the Web, Automatic Classification of Web Documents, Web Usage Mining

TEXTBOOKS:

1. Data Mining- Concepts and Techniques- Third Edition, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann Publication

2. Data Mining- Concepts and Techniques- Second Edition, Jiawei Han, Micheline Kamber, Morgan Kaufmann Publication

REFERENCES:

1. Data Mining: Practical Machine Learning Tools and Techniques, Ian H. Witten, Eibe Frank, Mark A. Hall, The Morgan Kaufmann Publication, 2011
2. R for Data Science, Hadley Wickham & Garrett Grolemund, O'Reilly, 2017

DATA MINING WITH R PROGRAMMING LAB

MCA I Year II Sem

Co-Requisite:

1. Course on "Datamining"

Objectives:

1. In this course student will learn how to program in R, and learn how to use R for Data mining

Outcomes:

1. Acquire the ability to do statistical computing in R
2. Acquire the ability to do data mining using R

List of Experiments:

1. Apply Descriptive statistics on AUTO MPG Data Set (download from UCI Repository).
2. Consider your class result data and analyze the performance of students in each subject at 1^{st} quantile (25%), 2^{nd} quantile (50%) and 3^{rd} quantile (75%).
3. Apply Linear Regression by extracting mpg and displacement from AUTO MPG Data Set. Consider mpg as dependent variable and displacement as independent variable.
4. Apply Multiple Regression by extracting mpg, cylinders, displacement, horsepower, weight, acceleration attributes from the AUTO MPG Data set, consider acceleration as dependent variable and remaining all are independent variables.
5. Consider the Indian Liver Patient Dataset (download from UCI Repository) divide the data set into training set and test set based on 75%:25%, train KNN model based on training set and generate confusion matrix based on test set.
6. Consider IRIS data set from UCI Repository divide it into training set and test set by applying sample techniques and train Decision tree on training set and generate confusion matrix based on test set.
7. Apply Random forest tree on Indian Liver Patient Dataset (download from UCI Repository) and generate confusion matrix and also compare accuracies of training and test of KNN and Random Forest Tree.
8. Apply K-Means Clustering algorithm on bag of words (download from UCI Repository).
9. Apply Hierarchical Clustering algorithm on bag of words (download from UCI Repository)

SOFTWARE ENGINEERING

MCA I Year II Sem

L	T	P	C
3	0	2*	4

Objectives:

1. The aim of the course is to provide an understanding of the working knowledge of the techniques for requirements elicitation analysis, estimation, design, testing and quality management of large software development projects.
2. Have knowledge about various process models, software process/product metrics, risk management, quality management and UML diagrams

Outcomes:

1. Demonstrate the ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in 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. Will have experience and/or awareness of testing problems and will be able to develop a simple test cases and test report.

UNIT - I

Introduction to Software Engineering: The nature of software, changing nature of software, The software process, The Process Framework, Umbrella Activities, Process Adaptation, Software Engineering Practice, The Essence of Practice, General Principles software development myths.

Software process structure: A Generic Process Model, defining a Framework Activity, Identifying a Task Set, Process Patterns, Process Assessment and Improvement.

Process models: Prescriptive Process Models: The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Specialized Process Models: Component-Based Development, The Formal Methods Model, Aspect-Oriented Software Development, The Unifi ed Process: A Brief History, Phases of the Unifi ed Process, Personal and Team Process Models: Personal Software Process, Team Software Process.

UNIT - II

Agile Software Development: Agile methods, Agile development techniques, Agile Project management, Scaling Agile methods.

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

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

System modeling: Context models, interaction models, structural models, behavioral models, model-driven architecture.

UNIT - III

Design Engineering: Design process and design quality, design concepts, the design model.

Creating an architectural design: software architecture, architectural styles, architectural design, conceptual model of UML, basic structural modeling, use case diagrams, class diagrams, sequence diagrams, collaboration diagrams, component diagrams, deployment diagrams.

UNIT - IV

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

Product metrics: a framework for product metrics, metrics for the requirements model, metrics for the design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT - V

Metrics for Process and Products: Metrics in the Process and Project Domains, Software measurement, metrics for software quality.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Quality Management: Elements of Software Quality Assurance, SQA Processes and Product Characteristics, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan.

TEXTBOOKS

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, Bruce R. Maxim, 8th edition, McGraw-Hill International Edition.
2. Software Engineering- Sommerville, 10th edition, Pearson.
3. The unified modeling language user guide by Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

REFERENCES

1. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiely.
2. Software Engineering principles and practice - WamanSJawadekar, TheMcGraw-Hill Companies.
3. Fundamentals of object oriented design in UML, Meilir page-Jones: Pearson.

SOFTWARE ENGINEERING LAB

MCA I Year II Sem

Co-requisite:

1. A Course on "Software Engineering"

Objectives

1. To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

Outcomes

1. Ability to translate end-user requirements into system and software requirements
2. Ability to generate a high level design of the system from the software requirements
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

Do the following 8 exercises for any two projects in the list of sample projects:

1) Development of problem statement.
2) Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3) Preparation of Software Configuration Management and Risk Management related documents.
4) Study and usage of any Design phase CASE tool.
5) Performing the Design by using any Design phase CASE tools.
6) Develop test cases for unit testing and integration testing.
7) Develop test cases for various white box and black box testing techniques.

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

TEXTBOOKS

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, Bruce R. Maxim, 8th edition, McGrawHill International Edition.
2. Software Engineering- Sommerville, 10th edition, Pearson.
3. The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

MOBILE APPLICATION DEVELOPMENT

MCA I Year II Sem

L	T	P	C
3	0	0	3

Prerequisites:

1. Acquaintance with JAVA programming
2. A Course on DBMS

Objectives:

1. To demonstrate the understanding of the fundamentals of Android operating systems
2. To improves the skills of using Android software development tools
3. To demonstrate the ability to develop software with reasonable complexity on mobile platform
4. To demonstrate the ability to deploy software to mobile devices
5. To demonstrate the ability to debug programs running on mobile devices

Outcomes:

1. Understand the working of Android OS Practically.
2. Acquires the skills to develop Android user interfaces
3. Develop, deploy and maintain the Android Applications.

UNIT - I

Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Android Studio, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools

Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes

Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes

UNIT - II

Android User Interface: Measurements – Device and pixel density independent measuring units Layouts – Linear, Relative, Grid and Table Layouts

User Interface (UI) Components – Editable and non editable TextViews, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers

Event Handling – Handling clicks or changes of various UI components

Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

UNIT - III

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS

Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity

Notifications – Creating and Displaying notifications, Displaying Toasts

UNIT - IV

Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference

UNIT - V

Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and etindelg data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)

TEXT BOOKS

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox) , 2012
2. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013

REFERENCE BOOKS

1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013

MOBILE APPLICATION DEVELOPMENT LAB

MCA I Year II Sem

L	T	P	C
0	0	2	1

Objectives

1. To learn how to develop Applications in android environment.
2. To learn how to develop user interface applications.
3. To learn how to develop URL related applications.

Objectives

1. Understands the working of Android OS Practically.
2. Acquire the skills to develop user interfaces.
3. Develop, deploy and maintain the Android Applications.

The student is expected to be able to do the following problems, though not limited.

1. Create an Android application that shows Hello + name of the user and run it on an emulator. (b) Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.
2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.
3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on right fragment instead of second screen with back button. Use Fragment transactions and Rotation event listener.
4. Develop an application that uses a menu with 3 options for dialling a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.
5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.
6. Create an application that uses a text file to store user names and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with Login Failed message.
7. Create a user registration application that stores the user details in a database table.
8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.
9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.
10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.
11. Create an application that saves user information like name, age, gender etc. in shared

preference and retrieves them when the program restarts.

12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.
13. Create an application that shows the given URL (from a text field) in a browser.
14. Develop an application that shows the current location's latitude and longitude continuously as the device is moving (tracking).
15. Create an application that shows the current location on Google maps.

Note:

Android Application Development with MIT App Inventor: For the first one week, the student is advised to go through the App Inventor from MIT which gives insight into the various properties of each component.

The student should pay attention to the properties of each component, which are used later in Android programming. Following are useful links:

1. http://ai2.appinventor.mit.edu
2. https://drive.google.com/file/d/0B8rTtW_91YclTWF4czdBMEpZcWs/view

TEXTBOOKS

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox) , 2012
2. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013

REFERENCES

1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013

JAVA PROGRAMMING LAB

MCA I Year II Sem

L	T	P	C
0	0	2	1

Prerequisites

1. A course on "C and Data Structures"

Co-requisite:

1. A Course on "Object-Oriented Programming Through Java"

Objectives:

1. Introduces object oriented programming concepts using the Java language.
2. Introduces the principles of inheritance and polymorphism; and demonstrates how they relate to the design of abstract classes
3. Introduces the implementation of packages and interfaces
4. Introduces exception handling, event handling and multithreading
5. Introduces the design of Graphical User Interface using applets and swings

Outcomes:

1. Develop applications for a range of problems using object-oriented programming techniques
2. Design simple Graphical User Interface applications

1. Use Eclipse or Netbean platform and get acquainted with the various menus. Create a test project, add a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea () that prints the area of the given shape.
3. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box. [Use JOption Pane –Input dialog, Message dialog]
4. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
5. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radiobuttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown.

6. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result. Handle any possible exceptions like divide by zero
7. a. Develop an applet in Java that displays a simple message.
   b. Develop an applet in Java that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named "Compute" is clicked.
8. Write a Java program that handles all mouse events and shows the event name at the centre of the window when a mouse event is fired (Using Adapter classes).
9. Write a Java program that handles all keyboard events and shows the event name at the centre of the window when a mouse event is fired (Use Adapter classes).

TEXTBOOKS

1. Java Fundamentals – A comprehensive Introduction, Herbert Schildt and Dale Skrien, TMH.

REFERENCES

1. Java for Programmers, P.J.Deitel and H.M.Deitel, Pearson education (OR) Java: How to Program P.J.Deitel and H.M.Deitel, PHI.
2. Object Oriented Programming through Java, P.Radha Krishna, Universities Press.

CYBER SECURITY
(Audit Course - I)

MCA I Year II Sem

L	T	P	C
2	0	0	0

Prerequisites: NIL

Objectives:

1. To familiarize various types of cyber-attacks and cyber-crimes.
2. To give an overview of the cyber laws
3. To study the defensive techniques against these attacks

Outcomes:

1. The students will be able to understand cyber-attacks, types of cybercrimes, cyber laws and also how to protect them self and ultimately the entire Internet community from such attacks

UNIT - I

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Spectrum of attacks, Taxonomy of various attacks, IP spoofing, Methods of defense, Security Models, risk management, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

UNIT - II

Cyberspace and the Law & Cyber Forensics: Introduction, Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy.

Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics, Special Techniques for Forensics Auditing.

UNIT - III

Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Mobile Devices: Security Implications for Organizations, Organizational Measures for Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era, Laptops.

UNIT - IV

Cyber Security: Organizational Implications: Introduction cost of cyber crimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing:

security risks and perils for organizations, social computing and the associated challenges for organizations.

Cybercrime and Cyber terrorism: Introduction, intellectual property in the cyberspace, the ethical dimensions of cyber crimes the psychology, mindset and skills of hackers and other cyber criminals.

UNIT - V

Cybercrime: Examples and Mini-Cases

Examples: Official Website of Maharashtra Government Hacked, Indian Banks Lose Millions of Rupees, Parliament Attack, Pune City Police Bust Nigerian Racket, e-mail spoofing instances.

Mini-Cases:

The Indian Case of online Gambling, An Indian Case of Intellectual Property Crime, Financial Frauds in Cyber Domain.

TEXT BOOKS

1. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley.
2. B. B. Gupta, D. P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335, 2018.

REFERENCES

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Cyber Security Engineering, A pratical approach for systems and software Assurance, Nancy R.Mead, Carol C.Woody, Pearson Education.