

UTKAL UNIVERSITY

Syllabus

for

Integrated Master of Computer Applications (IMCA)

(5-Years Syllabus)



**Department of Computer Science & Applications
Utkal University
Bhubaneswar (Odisha)
2019-20**

**(The Model Syllabus of BCA Programme designed by Higher Education,
Govt. of Odisha is being adopted for the first 3 years of Integrated MCA)**

COMMON SYLLABUS FOR BACHELOR IN COMPUTER APPLICATIONS (BCA)

Preamble

There are various innovative technologies that have come up in the IT world which are transforming the business landscape at a rapid pace. The challenges of the IT sector today are a lot different from those that existed a few years ago. The changing technological environment has made it necessary for students who aspire to a career in IT to be well aware of the latest innovations. The BCA program is specially designed to cater to the present industry skills and knowledge needs. It comprises of theoretical and practical training sessions that can prepare students for the well-rewarding jobs of the future.

Programme Details

- The programme is designed keeping in mind the needs of the industry. It comprises of subjects that are relevant to the current technological environment.
- Develop strong fundamental knowledge in core IT subjects such as C Programming, Computer Organization and Data Structures
- Learn the basics of programming in Java, Web Technology and Database Systems
- Get a chance to work on projects that give better exposure to current industry

CHOICE BASED CREDIT SYSTEM (CBCS): The CBCS provides an opportunity for the students to choose courses from the prescribed courses comprising core, elective/minor or skill based courses. The courses can be evaluated following the grading system, which is considered to be better than the conventional marks system. Therefore, it is necessary to introduce uniform grading system in the entire higher education in Odisha. This will benefit the students to move across institutions within Odisha to begin with and across states and countries. The uniform grading system will also enable potential employers in assessing the performance of the candidates. In order to bring uniformity in evaluation system and computation of the Cumulative Grade Point Average (CGPA) based on student's performance in examinations, the UGC has formulated the guidelines to be followed.

PROGRAMME LEARNING OUTCOMES

The Bachelor in Computer Applications (BCA) programme enables students to attain, by the time of graduation:

- Ability to apply knowledge of computing appropriate to the discipline.

- Ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
- Ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs.
- Ability to function effectively on teams to accomplish a common goal.
- Understanding of professional, ethical, legal, security and social issues and responsibilities.
- Ability to communicate effectively with a range of audiences.
- Ability to analyze the local and global impact of computing on individuals, organizations, and society.
- Recognition of the need for an ability to engage in continuing professional development.
- Ability to use current techniques, skills, and tools necessary for computing practice.
- Ability to apply algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
- Ability to apply design and development principles in the construction of software systems of varying complexity.

OUTLINE OF CHOICE BASED CREDIT SYSTEM

1. Core Course: A course, which should compulsorily be studied by a candidate as a core requirement is termed as a Core course.

2. Elective Course: Generally, a course which can be chosen from a pool of courses and which may be very specific or specialized or advanced or supportive to the discipline/ subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the candidate's proficiency/skill is called an Elective Course.

2.1 Discipline Specific Elective (DSE) Course: Elective courses may be offered by the main discipline/subject of study is referred to as Discipline Specific Elective. The University/Institute may also offer discipline related Elective courses of interdisciplinary nature (to be offered by main discipline/subject of study).

2.2 Dissertation/Project: An elective course designed to acquire special/advanced knowledge, such as supplement study/support study to a project work, and a candidate studies such a course on his own with an advisory support by a teacher/faculty member is called dissertation/project.

2.3 Generic Elective (GE) Course: An elective course chosen generally from an unrelated discipline/subject, with an intention to seek exposure is called a Generic Elective.

P.S.: A core course offered in a discipline/subject may be treated as an elective by other discipline/subject and vice versa and such electives may also be referred to as Generic Elective.

Project work/Dissertation is considered as a special course involving application of knowledge in solving / analyzing /exploring a real life situation / difficult problem. A Project/Dissertation work would be of 6 credits. A Project/Dissertation work may be given in lieu of a discipline specific elective paper.

GUIDELINES FOR PROJECT FORMULATION

As the project work constitutes a major component in most of the professional programs and it is to be carried out with due care and should be executed with seriousness by the candidates.

TYPE OF PROJECT

As majority of the students are expected to work out a real-life project in some industry/research and development laboratories/educational institutions/software companies, it is suggested that the project is to be chosen which should have some direct relevance in day-to-day activities of the candidates in his/her institution. It is not mandatory for a student to work on a real-life project. The student can formulate a project problem with the help of Guide.

PROJECT PROPOSAL (SYNOPSIS)

The project proposal should be prepared in consultation with the guide. The project proposal should clearly state the project objectives and the environment of the proposed project to be undertaken. The project work should compulsorily include the software development. The project proposal should contain complete details in the following form:

1. Title of the Project
2. Introduction and Objectives of the Project
3. Project Category (RDBMS/ OOPS/ Networking/ Multimedia/ Artificial Intelligence/ Expert Systems etc.)
4. Analysis (DFDs at least up to second level, ER Diagrams/ Class Diagrams/ Database Design etc. as per the project requirements).
5. A complete structure which includes: Number of modules and their description to provide an estimation of the student's effort on the project. Data Structures as per the project requirements for all the modules. Process Logic of each module. Testing process to be used. Reports generation
6. Tools / Platform, Hardware and Software Requirement specifications
7. Future scope and further enhancement of the project.

BACHELOR IN COMPUTER APPLICATIONS (BCA)

Effective from 2019-20 Academic Session

SEM.	COURSE OPTED	COURSE NAME	CREDITS
I	Ability Enhancement Course-1	AEC-1 (Environmental Science)	4
	Core Course-1	Digital Logic	4
	Core Course-1 Practical	Digital Logic LAB	2
	Core Course-2	Programming using C	4
	Core Course-2 Practical	Programming using C LAB	2
	Generic Elective/ Inter-disciplinary Course -1	GE/IC-1 (Principles of Management)	4
	Generic Elective/ Inter-disciplinary Course -1 Tutorial/Practical	GE/IC -1 Tutorial/ LAB (Principles of Management Tutorial)	2
II	Ability Enhancement Course-2	AEC-2 (English/ MIL Communication)	4
	Core Course-3	Programming using C++	4
	Core Course-3 Practical	Programming using C++ LAB	2
	Core Course-4	Data Structure	4
	Core Course-4 Practical	Data Structure LAB	2
	Generic Elective/ Inter-disciplinary Course -2	GE/IC-2 (Statistics)	4
	Generic Elective/ Inter-disciplinary Course-2 Tutorial /Practical	GE/IC -2 Tutorial/ LAB (Statistics LAB)	2
III	Core Course-5	Computer Organization	4
	Core Course-5 Practical	Computer Organization LAB	2
	Core Course-6	JAVA Programming	4
	Core Course-6 Practical	JAVA Programming LAB	2
	Core Course-7	Discrete Mathematical Structures	4
	Core Course-7 Practical	Discrete Mathematical Structures LAB	2
	Skill Enhancement Course-1	SEC-1 (Python Programming)	4
	Generic Elective/ Inter-disciplinary Course -3	GE/IC-3 (Business Accounting)	4
	Generic Elective/ Inter-disciplinary Course -3 Tutorial/Practical	GE/IC -3 Tutorial/ LAB (Business Accounting Tutorial)	2
IV	Core Course-8	Operating System	4
	Core Course-8 Practical	Operating System LAB	2
	Core Course-9	Computer Networks	4
	Core Course-9 Practical	Computer Networks LAB	2
	Core Course-10	Database Systems	4

	Core Course-10 Practical	Database Systems LAB	2
	Skill Enhancement Course-2	SEC-2 (Android Programming)	4
	Generic Elective/ Inter-disciplinary Course -4	GE/IC-4 (Business Economics)	4
	Generic Elective/ Inter-disciplinary Course -4 Tutorial/Practical	GE/IC-4 Tutorial/ LAB (Business Economics Tutorial)	2
V	Core Course-11	Web Technology	4
	Core Course-11 Practical	Web Technology LAB	2
	Core Course-12	Software Engineering	4
	Core Course-12 Practical	Software Engineering LAB	2
	Discipline Specific Elective-1	DSE-1 (Unix Programming)	4
	Discipline Specific Elective-1 Practical	DSE-1 LAB/ Tutorial (Unix Programming LAB)	2
	Discipline Specific Elective-2	DSE-2 (Data Mining)	4
	Discipline Specific Elective-2 Practical	DSE-2 LAB/ Tutorial (Data Mining LAB)	2
VI	Core Course-13	Computer Graphics	4
	Core Course-13 Practical	Computer Graphics LAB	2
	Core Course-14	Numerical Techniques	4
	Core Course-14 Practical	Numerical Techniques LAB	2
	Discipline Specific Elective-3	DSE-3 (Data Science)	4
	Discipline Specific Elective-3 Practical	DSE-3 LAB/ Tutorial (Data Science LAB)	2
	Discipline Specific Elective-4	DSE-4 (Project Work / E-Commerce)	6/4
	Discipline Specific Elective-4 Practical	DSE-4 LAB/ Tutorial (E-Commerce Tutorial)	2

CORE Papers:

CORE – 1: Digital Logic

CORE – 2: Programming Using C

CORE – 3: Programming Using C++

CORE – 4: Data Structure

CORE – 5: Computer Organization

CORE – 6: Java Programming

CORE – 7: Discrete Mathematical Structures

CORE – 8: Operating System

CORE – 9: Computer Networks
CORE – 10: Database Systems
CORE – 11: Web Technologies
CORE – 12: Software Engineering
CORE – 13: Computer Graphics
CORE – 14: Numerical Techniques

Discipline Specific Electives (DSE) Papers:

DSE–1: Unix Shell Programming
DSE–2: Data Mining
DSE–3: Data Science
DSE–4: Project Work / E-Commerce

Skill Enhancement Courses (SEC):

SEC – 1: Python Programming
SEC – 2: Android Programming

Ability Enhancement Courses (AEC): (Credit: 02 each)

AEC – 1: Environmental Science.
AEC – 2: English /MIL Communication

Generic Elective (GE)/ Interdisciplinary Course (IC):

GE/IC – 1: Principles of Management
GE/IC – 2: Statistics
GE/IC – 3: Business Accounting
GE/IC – 4: Business Economics

DETAILED SYLLABUS

CORE-1: DIGITAL LOGIC

OBJECTIVES

- To understand different methods used for the simplification of Boolean functions and binary arithmetic.
 - To design and implement combinational circuits, synchronous & asynchronous sequential circuits.
 - To study in detail about Semiconductor Memory Systems.
-

Unit-1

Character Codes, Decimal System, Binary System, Decimal to Binary Conversion, Hexadecimal Notation, Boolean Algebra, Basic Logic Functions: Electronic Logic Gates, Synthesis of Logic Functions, Minimization of Logic Expressions, Minimization using Karnaugh Maps, Synthesis with NAND and NOR Gates, Tri-State Buffers

Unit-2

Arithmetic: Addition and Subtraction of Signed Numbers, Addition/ Subtraction Logic Unit, Design of Fast Adders: Carry-Lookahead Addition, Multiplication of Positive Numbers, Signed-Operand Multiplication: Booth Algorithm, Fast Multiplication: Bit-Pair Recoding Multipliers, Carry-Save Addition of Summands, Integer Division, Floating-Point Numbers and Operations: IEEE Standard for Floating-Point Numbers, Arithmetic Operations on Floating-Point Numbers, Guard Bits and Truncation, Implementing Floating-Point Operations.

Unit-3

Flip-Flops, Gated Latches, Master-Slave Flip-Flops, Edge-Triggering, T Flip-Flops, JK Flip-Flops. Registers and Shift Registers, Counters, Decoders, Multiplexers, Programmable Logic Devices (PLDs), Programmable Array Logic (PAL), Complex Programmable Logic Devices (CPLDs), Field-Programmable Gate Array (FPGA), Sequential Circuits, UP/ DOWN Counters, Timing Diagrams, The Finite State Machine Model, Synthesis of Finite State Machines.

Unit-4

Memory System: Semiconductor RAM Memories, Internal Organization of Memory Chips, Static Memories, Asynchronous DRAMS, Synchronous DRAMS, Structure of Large Memories, Memory System Considerations, RAMBUS Memory. Read-Only Memories: ROM, PROM, EPROM, EEPROM, Flash Memory, Speed, Size, and Cost of Memory. Secondary Storage: Magnetic Hard Disks, Optical Disks, Magnetic Tape Systems.

Text Books:

1. Carl Hamacher, Z. Vranesic, S. Zaky: Computer Organization, 5/e (TMH)

Reference Books:

1. M. Morris Mano: Digital Logic and Computer Design, Pearson

CORE-2 PRACTICAL: DIGITAL LOGIC LAB

1. Introduction to Xilinx software (VHDL)

Write the VHDL code for

2. Realizing all logic gates.
3. Combination Circuit.
4. ADDER.
5. SUBTRACTOR.
6. MUX.
7. DE-MUX.
8. Encoder.
9. Decoder.
10. PAL.
11. PLA.

Write the VHDL program for the following Sequential Logic Circuits

12. Flip Flops.
13. Shift Registers.
14. Counters.
15. Memory Elements.

CORE-2: PROGRAMMING USING C

OBJECTIVES:

- To learn basics of C programming language.
 - To be able to develop logics to create programs/ applications in C.
-

Unit-1

Introduction: Introduction to Programming Language, Introduction to C Programming, Keywords & Identifiers, Constants, Variables, Input and Output Operations, Compilation and pre-processing, **Data types:** Different data types, Data types qualifier, modifiers, Memory representation, size and range, **Operators:** Operators (Arithmetic, Relational, Logical, Bitwise,

Assignment & compound assignment, Increment & Decrement, Conditional), Operator types (unary, binary, ternary). Expressions, Order of expression (Precedence and associativity)

Control structures: Decision Making and Branching (Simple IF Statement, IF...ELSE Statement, Nesting IF... ELSE Statement, ELSE IF Ladder), Selection control structure (Switch Statement).

Unit-2

Loops: The WHILE Statement, The DO...WHILE Statement, The FOR Statement, Jumps in Loops, **Array:** Concept of Array, Array Declaration, types of array (one and multiple dimension), Character Arrays and Strings, Subscript and pointer representation of array, Array of Pointers, Limitation of array, **Pointers:** Concept of Pointer (null pointer, wild pointer, dangling pointer, generic pointer), Pointer Expressions, Accessing the Address of a Variable, Declaring Pointer Variables, Initializations of Pointer Variable, Accessing a Variable through its Pointer, Pointer arithmetic.

Unit-3

Storage

class: Types (auto, register, static, extern), scope rules, declaration and definition. **Function:** Function & types (User defined function, library function) Function Definition, Declaration, Function Calls, Header file and library, Function Arguments, string handling function (strlen, strcmp, strcpy, strncpy, strcat, strstr), Function recursion, Functions Returning Pointers, Pointers to Functions, Command line arguments, Application of pointer (dynamic memory allocation).

Unit-4

Structure and Union: Defining, Declaring, Accessing, Initialization Structure, nested structure, self-referential structure, bit-field, Arrays of Structures, Structures and Functions, Unions, difference between structure and union, active data member, structure within union, Self-referential Structure.

File: File Management in C, Defining and Opening a File, File opening modes (read, write, append), Closing a File, File operations, file and stream, Error Handling During I/O Operations, sequential and random access file, low level and high level file.

Text Books:

1. E. Balagurusamy, "Programming in ANSI C", 4/e, (TMH)

Reference Books:

1. B. Kernighan & Dennis Ritchie, "The C Programming Language", 2/e PHI
2. Paul Deitel, Harvey Deitel, "C: How to Program", 8/e, Prentice Hall.
3. P.C. Sethi, P.K. Behera, "Programming using C", Kalyani Publisher, Ludhiana

Core-2 PRACTICAL: PROGRAMMING USING C LAB

1. Write a Program to find greatest among three numbers.
2. Write a Program to all arithmetic operation using switch case.
3. Write a Program to print the sum and product of digits of an integer.
4. Write a Program to reverse a number.
5. Write a Program to compute the sum of the first n terms of the following series

$$S = 1 + 1/2 + 1/3 + 1/4 + \dots$$

6. Write a Program to compute the sum of the first n terms of the following series

$$S = 1 - 2 + 3 - 4 + 5 - \dots$$
7. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
8. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
9. Write a Program to compute the factors of a given number.
10. Write a program to swap two numbers using macro.
11. Write a Program to print a triangle of stars as follows (take number of lines from user):

```

*
***
*****
*****

```
12. Write a Program to perform following actions on an array entered by the user:
 - a) Print the even-valued elements
 - b) Print the odd-valued elements
 - c) Calculate and print the sum and average of the elements of array
 - d) Print the maximum and minimum element of array
 - e) Remove the duplicates from the array
 - f) Print the array in reverse order

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
13. Write a Program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
14. Write a program that swaps two numbers using pointers.
15. Write a program in which a function is passed address of two variables and then alter its contents.
16. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
17. Write a program to find sum and average of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions.
18. Write a menu driven program to perform following operations on strings:
 - a) Show address of each character in string
 - b) Concatenate two strings without using strcat function.
 - c) Concatenate two strings using strcat function.
 - d) Compare two strings
 - e) Calculate length of the string (use pointers)
 - f) Convert all lowercase characters to uppercase
 - g) Convert all uppercase characters to lowercase
 - h) Calculate number of vowels
 - i) Reverse the string
19. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
20. Write a program to copy the content of one file to other.

CORE-3: PROGRAMMING USING C++

OBJECTIVES

- To know about the Object Oriented Programming concepts.
 - To learn basics of C++ programming language.
 - To be able to develop logics to create programs/ applications in C++.
-

Unit-1

Principles of Object-Oriented Programming: Object-Oriented Programming (OOP) Paradigm, Basic Concepts of OOP, Benefits of OOP, Characteristics of OOPS, Object Oriented Languages, Applications of OOP.

Introduction to C++, Difference between C & C++, Tokens, Data types, Operators, Structure of C++ Program, C++ statements, Expressions and Control Structures.

Functions in C++: Argument passing in function, Inline Functions, Default Arguments, Const. Arguments, Friend function.

Unit-2

Classes and Objects: Defining Member Functions, Making an outside Function Inline, Nested Member Functions, Private Member Functions, Arrays within a Class, Memory Allocation for Objects, Static Data Members, Static Member Functions, Arrays of Objects, Objects as Function Arguments, Friend Functions.

Constructors & Destructors: Constructors, Parameterized Constructors, Constructors with Default Arguments, Dynamic Initialization of Objects, Copy Constructor, Dynamic Constructors, Destructors.

Unit-3

Inheritance: Basics of Inheritance, Type of Inheritance, Virtual Base Classes, Abstract Classes, Member Classes, Nesting of Classes. Polymorphism: Pointers, Pointers to Objects, this Pointer, Pointers to Derived Classes, Virtual Functions, Pure Virtual Functions, Function Overloading, Operator Overloading.

Unit-4

Managing Console I/O Operations: C++ Streams, C++ Stream Classes, Unformatted I/O Operations, Formatted Console I/O Operations, Managing Output with Manipulators.

Files: Classes for File Stream Operations, Opening and Closing a File, Detecting end-of-file, File Modes, File Pointers and their Manipulations, Sequential Input and Output Operations, Updating a File: Random Access, Error Handling during File Operations, Command-line Arguments.

Text Books

1. E. Balgurusawmy, Object Oriented Programming with C++, 4/e (TMH).
2. Paul Deitel, Harvey Deitel, "C++: How to Program", 9/e. Prentice Hall.

Reference Books:

1. Bjarne Stroustrup, Programming - Principles and Practice using C++, 2/e, Addison-Wesley 2014
2. Herbtz Schildt, C++: The Complete reference, MGH, 4/ed.
3. P. C. Sethi, P. K. Behera, "Programming in C++"- Kalyani Publisher, Ludhiana

CORE-3 PRACTICAL: PROGRAMMING USING C++ LAB

1. Write a Program to find greatest among three numbers using nested if...else statement.
2. Write a Program to check a number is prime or not.
3. Write a Program to find the GCD and LCM of two numbers.
4. Write a program to print the result for following series: $1! + 2! + 3! + \dots$
5. Write a program to print multiplication table from 1 to 10.
6. Write a Program for Swapping of two numbers using pass by value.
7. Write a Program for Swapping of two numbers using pass by address.
8. Write a Program for Swapping of two numbers using pass by reference.
9. Write a Program to find sum of four numbers using default argument passing.
10. Write a Program to find square and cube of a number using inline function.
11. Write a Program to find the factorial of a number.
12. Write a Program to find reverse of a number.
13. Write a program to find sum of four numbers using default argument passing in member function.
14. Write a Program to find area of circle, triangle and rectangle using function overloading.
15. Write a program to distinguish the properties of static and non-static ata members.
16. Write a program to show the method of accessing static private member function.
17. Write a program to show the ways of calling constructors and destructors.
18. Write a program to perform ++ operator overloading using member function.
19. Write a program to perform ++ operator overloading using friend function.
20. Write a program to perform + operator overloading for two complex number addition.
21. Write a program to perform + operator overloading for string concatenation.
22. Write a program to perform single inheritance.
23. Write a program to perform multiple inheritance.
24. Write a program to create an integer array using new operator and find the sum and average of array elements.
25. Write a program to implement virtual destructor.
26. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
27. Write a program to Copy the contents of one file to other.

CORE-4: DATA STRUCTURE

OBJECTIVES

- To learn how the choice of data structures impacts the performance of programs.
 - To study specific data structures such as arrays, linear lists, stacks, queues, hash tables, binary trees, binary search trees, heaps and AVL trees.
 - To learn efficient searching and sorting techniques.
-

Unit-1

Introduction: Basic Terminology, Data structure, Time and space complexity, Review of Array, Structures, Pointers.

Linked Lists: Dynamic memory allocation, representation, Linked list insertion and deletion, Searching, Traversing in a list, Doubly linked list, Sparse matrices.

Unit-2

Stack: Definition, Representation, Stack operations, Applications (Infix–Prefix–Postfix Conversion & Evaluation, Recursion).

Queues: Definition, Representation, Types of queue, Queue operations, Applications.

Unit-3

Trees: Tree Terminologies, General Tree, Binary Tree, Representations, Traversing, BST, Operations on BST, Heap tree, AVL Search Trees, M-way search tree, Applications of all trees.

Unit-4

Sorting: Exchange sorts, Selection Sort, Bubble sort, Insertion Sorts, Merge Sort, Quick Sort, Radix Sort, Heap sort.

Searching: Linear search, Binary search.

Text book

1. Classic Data Structure , D. Samanta , PHI , 2/ed.

REFERENCES

1. Ellis Horowitz, Sartaj Sahni, “Fundamentals of Data Structures”, Galgotia Publications, 2000.
2. Sastry C.V., Nayak R, Ch. Rajaramesh, Data Structure & Algorithms, I. K. International Publishing House Pvt. Ltd, New Delhi.

CORE – 4 PRACTICAL: DATA STRUCTURE LAB

Write a C/ C++ Program for the followings

1. To insert and delete elements from appropriate position in an array.
2. To search an element and print the total time of occurrence in the array.
3. To delete all occurrence of an element in an array.
4. Array implementation of Stack.
5. Array implementation of Linear Queue.
6. Array implementation of Circular Queue.
7. To implement linear linked list and perform different operation such as node insert and delete, search of an item, reverse the list.
8. To implement circular linked list and perform different operation such as node insert and delete.
9. To implement double linked list and perform different operation such as node insert and delete.
10. Linked list implementation of Stack.
11. Linked list implementation of Queue.
12. Polynomial representation using linked list.
13. To implement a Binary Search Tree.
14. To represent a Sparse Matrix.
15. To perform binary search operation.
16. To perform Bubble sort.
17. To perform Selection sort.
18. To perform Insertion sort.
19. To perform Quick sort.
20. To perform Merge sort.

CORE-5: COMPUTER ORGANIZATION

OBJECTIVES

- To study the basic organization of digital computers (CPU, memory, I/O, software).
 - To have a better understanding and utilization of digital computers.
 - To be familiar with Assembly Language Programming (ALP)
-

Unit-1

Basic Structure of Computers: Computer Types, Functional Units, Input Unit, Memory Unit, Arithmetic and Logic Unit, Output Unit, Control Unit, Basic Operational Concepts, Bus Structures, Software. Machine Instructions and Programs: Numbers, Arithmetic Operations, and Characters: Number Representation, Addition of Positive Numbers, Addition and Subtraction of

Signed Numbers, Overflow of Integer Arithmetic, Floating-Point Numbers & Operations, Characters, Memory Locations and Addresses, Byte Addressability, Word Alignment, Accessing Numbers, Characters, and Character Strings, Memory Operations, Instructions and Instruction Sequencing, Register Transfer Notation, Basic Instruction Types, Instruction Execution and Straight-Line Sequencing, Branching, Condition Codes, Generating Memory Addresses, Addressing Modes, Implementation of Variables and Constants, Indirection and Pointers, Indexing and Arrays, Relative Addressing.

Unit-2

Basic Processing Unit: Register Transfers, Performance on Arithmetic or Logic Operation, fetching a Word from Memory, Storing a Word in Memory. Execution of a Complete Instruction, Branch Instruction, Multiple Bus Organization Hardwired Control, A Complete Processor. Microprogrammed Control: Microinstructions, Microprogram Sequencing, Wide-Branch Addressing, Microinstructions with Next-Address Field, Prefetching Microinstructions, Emulation. Cache Memories: Mapping Functions, Replacement Algorithms, Example of Mapping Technique. Performance Considerations: Interleaving, Hit Rate and Miss Penalty, Caches on Processor Chip, Other Enhancements, Virtual Memories: Address Translation.

Unit-3

Input/ Output Organization: Accessing I/O Devices, Interrupts, Interrupt Hardware, Enabling & Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions. Direct Memory Access, Bus Arbitration, Buses, Synchronous Bus, Asynchronous Bus, Interface Circuits: Parallel Port, Serial Port, Standard I/O Interfaces, Peripheral Component Interconnect (PCI) Bus, SCSI Bus, Universal Serial Bus (USB)

Unit-4

Pipelining: Role of Cache Memory, Pipeline Performance, Data Hazards: Operand Forwarding, Handling Data Hazards in Software, Side Effects. Instruction Hazards: Unconditional Branches, Conditional Branches and Branch Prediction. Influence on Instruction Sets: Addressing Modes, Condition Codes, Data path and Control Considerations. Superscalar Operation: Out-of-Order Execution, Execution Completion, Dispatch Operation, RISC & CISC Processors.

Text Books

1. Carl Hamacher, Z. Vranesic, S. Zaky: Computer Organization, 5/Ed (TMH)

Reference Books

1. William Stallings: Computer Organization and Architecture (Design for Performance), 9/Ed
2. S. Brown, & Z. Vranesic, "Fundamentals of Digital Logic Design with VHDL", 2/Ed, McGraw-Hill

CORE-5 PRACTICAL: COMPUTER ORGANIZATION LAB

1. Study of the complete Architecture of 8085 Microprocessor along with its instruction set.
2. Introduction to GNU Simulator 8085, with its features.

3. Write an Assembly Language Program to add N consecutive numbers.
4. Write an Assembly Language Program to find the smallest and largest number from a given series.
5. Write an Assembly Language Program for subtraction of two 8-bit numbers.
6. Write an Assembly Language Program for displaying a Rolling message “Hello 123”.
7. Write an Assembly Language Program to perform ASCII to Decimal conversion.
8. Write an Assembly Language Program to add two unsigned binary numbers.
9. Write an Assembly Language Program to subtraction of two unsigned binary numbers.

CORE – 6: JAVA PROGRAMMING

OBJECTIVES

1. To learn the fundamentals of Object Oriented Programming in Java environment.
 2. To learn the use of Java language and the Java Virtual Machine.
 3. To write simple Java programming applications.
-

Unit-1

Introduction to Java: Java History, Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords (super, this, final, abstract, static, extends, implements, interface) , Data Types, Wrapper class, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods). Input through keyboard using Command line Argument, the Scanner class, BufferedReader class.

Unit-2

Object-Oriented Programming Overview: Principles of Object-Oriented Programming, Defining & Using Classes, Class Variables & Methods, Objects, Object reference, Objects as parameters, final classes, Garbage Collection.

Constructor- types of constructor, this keyword, super keyword. Method overloading and Constructor overloading. Aggregation vs Inheritance, Inheritance: extends vs implements, types of Inheritance, Interface, Up-Casting, Down-Casting, Auto-Boxing, Enumerations, Polymorphism, Method Overriding and restrictions. Package: Pre-defined packages and Custom packages.

Unit-3

Arrays: Creating & Using Arrays (1D, 2D, 3D and Jagged Array), Array of Object, Referencing Arrays Dynamically. Strings and I/O: Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability& Equality, Passing Strings To & From Methods, StringBuffer Classes and StringBuilder Classes. IO package: Understanding StreamsFile class and its methods, Creating, Reading, Writing using classes: Byte and Character streams, FileOutputStream, FileInputStream, FileWriter, FileReader, InputStreamReader, PrintStream, PrintWriter. Compressing and Uncompressing File.

Unit-4

Exception Handling, Threading, Networking and Database Connectivity: Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

Text Books:

1. E. Balagurusamy, “Programming with Java”, TMH, 4/Ed,

Reference books:

1. Herbert Schildt, “The Complete Reference to Java”, TMH, 10/Ed.

CORE – 6 PRACTICAL: JAVA PROGRAMMING LAB

1. To find the sum of any number of integers entered as command line arguments.
2. To find the factorial of a given number.
3. To convert a decimal to binary number.
4. To check if a number is prime or not, by taking the number as input from the keyboard.
5. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
6. Write a program that show working of different functions of String and StringBufferclasses like setCharAt(), setLength(), append(), insert(), concat()and equals().
7. Write a program to create a – “distance” class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
8. Modify the – “distance” class by creating constructor for assigning values (feetandinches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
9. Write a program to show that during function overloading, if no matching argument is found, then Java will apply automatic type conversions (from lower to higher data type)
10. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword.
11. Write a program to show the use of static functions and to pass variable length arguments in a function.

14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program – “DivideByZero” that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
21. Write a program to demonstrate priorities among multiple threads.
22. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed(), mouseReleased() & mouseDragged().
23. Write a program to demonstrate different keyboard handling events.

CORE – 7: DISCRETE MATHEMATICAL STRUCTURES

OBJECTIVES

- To learn the mathematical foundations for Computer Science.
 - Topics covered essential for understanding various courses.
-

Unit-1

Logics and Proof: Propositional Logic, Propositional Equivalences, Predicates and Quantifiers Nested Quantifiers, Rules inference, Mathematical Induction.

Sets and Functions: Sets, Relations, Functions, Closures of Equivalence Relations, Partial ordering well ordering, Lattice, Sum of products and product of sums principle of Inclusions and Exclusions

Unit-2

Combinatory: Permutations, Combinations, Pigeonhole principle

Recurrence Relation: Linear and Non-linear Recurrence Relations, Solving Recurrence Relation using Generating Functions.

Unit-3

Graphs: Introduction to graphs, graphs terminologies, Representation of graphs, Isomorphism,

Connectivity & Paths: Connectivity, Euler and Hamiltonian Paths, Introduction to tree, tree traversals, spanning tree and tree search: Breadth first search, Depth first search, cut-set, cut-vertex.

Unit-4

Modeling Computation: Finite State Machine, Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), Grammars and Language, Application of Pumping Lemma for Regular Language.

Text Books:

1. “Discrete Mathematics and its Applications with Combinatory and Graph Theory” 7th edition by Kenneth H. Rosen.

Reference Books:

1. Elements of Discrete Mathematics by C. L. Liu and D.P. Mohapatra, TMH, 2012
2. J. P Tremblay, R. Manohar, “Discrete Mathematical Structures with Applications to Computer Science”, TMH, 1997.
3. A Modern Approach to Discrete Mathematics and Structure by J. K. Mantri & T. K Tripathy ,Laxmi Publication

CORE – 7 PRACTICAL: DISCRETE MATHEMATICAL STRUCTURES LAB

Write the following programs using C/ C++

1. Tower of Hanoi
2. Graph representation using Adjacency List.
3. Graph representation using Adjacency Matrix.
4. String Matching using finite state machine.
5. Detecting whether a number is even or odd using Finite State Machine.
6. To identify keywords such as char, const, continue using Finite State Machine.
7. To find the power set for a given set.
8. To find GCD of two numbers using recursion.
9. To find Binomial coefficients.
10. To find Permutation and Combination result for a given pair of values n and r.
11. To check a number is prime or not.
12. To calculate the Euclidean distance between two points.
13. To find the Roots of polynomials.
14. Find the shortest path pair in a plane.

CORE-8: OPERATING SYSTEM

OBJECTIVES

- To understand Operating system structure and services.
 - To understand the concept of a Process, memory, storage and I/O management.
-

Unit-1

Introduction to Operating System, System Structures: Operating system services, system calls, system programs, Operating system design and implementation, Operating system structure.

Unit-2

Process Management: Process Concept, Operations on processes, Process scheduling and algorithms, Inter-process Communication, Concepts on Thread and Process, Deadlocks: Deadlock detection, deadlock prevention, and deadlock avoidance fundamentals.

Unit-3

Memory Management Strategies: Swapping, Contiguous Memory Allocation, Paging, Segmentation, Virtual Memory Management: Concepts, implementation (Demand Paging), Page Replacement, Thrashing.

Unit-4

Storage Management: File System concept, Access Methods, File System Mounting, File Sharing and File Protection, Implementing File Systems, Kernel I/O Systems.

Text book:

1. Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, Eighth Edition, Wiley Student Edition 2009.

Reference book:

1. Modern Operating System, Tanenbaum, Pearson, 4/Ed. 2014
2. Richard F Ashley, Linux with Operating System Concepts, Chapman and Hall/CRC Published August 26, 2014
3. Richard Blum, Linux Command Line and Shell Scripting Bible, O' Reilly

CORE-8 PRACTICAL: OPERATING SYSTEM LAB

1. Write a program (using *fork()* and/or *exec()* commands) where parent and child execute:
 - a) same program, same code.
 - b) same program, different code.

- c) before terminating, the parent waits for the child to finish its task.
2. Write a program to report behavior of Linux kernel including kernel version, CPU type and model. (CPU information)
 3. Write a program to report behavior of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
 4. Write a program to print file details including owner access permissions, file access time, where file name is given as argument.
 5. Write a program to copy files using system calls.
 6. Write a program using C to implement FCFS scheduling algorithm.
 7. Write a program using C to implement Round Robin scheduling algorithm.
 8. Write a program using C to implement SJF scheduling algorithm.
 9. Write a program using C to implement non-preemptive priority based scheduling algorithm.
 10. Write a program using C to implement preemptive priority based scheduling algorithm.
 11. Write a program using C to implement SRTF scheduling algorithm.
 12. Write a program using C to implement first-fit, best-fit and worst-fit allocation strategies.

CORE – 9: COMPUTER NETWORKS

OBJECTIVES

- To learn how do computers and terminals actually communicate with each other.
 - To understand the parts of a communication network and how they work together.
-

Unit-1

Introduction to Data Communications and Network Models: Protocols and Standards, Layers in OSI Models, Analog and Digital Signals, Transmission Modes, Transmission Impairment, Data Rate Limits, Performance, Digital Transmission, Network Devices & Drivers: Router, Modem, Repeater, Hub, Switch, Bridge (fundamental concepts only).

Unit-2

Signal Conversion: Digital-to-Digital Conversion, Analog-to-Digital Conversion, Digital-to-analog Conversion, Analog-to-analog Conversion.

Transmission Media: Guided Media, Unguided Media, Switching Techniques: Packet Switching, Circuit Switching, Datagram Networks, Virtual-Circuit Networks, and Structure of a Switch.

Unit-3

Error Detection and Correction: Checksum, CRC, Data Link Control: Framing, Flow and Error Control, Noiseless Channels, Noisy channels, (Stop and Wait ARQ, Sliding Window Protocol, Go Back N, Selective Repeat) HDLC, Point-to-Point Protocol. Access Control: TDM, CSMA/CD, and Channelization (FDMA, TDMA, and CDMA).

Unit-4

Network Layer: Logical Addressing, IPv4 Addresses, IPv6 Addresses, Virtual-Circuit Networks: Frame Relay and ATM, Transport Layer: Process-Process Delivery: UDP, TCP. Application layers: DNS, SMTP, POP, FTP, HTTP, Basics of WiFi (Fundamental concepts only), Network Security: Authentication, Basics of Public Key and Private Key, Digital Signatures and Certificates (Fundamental concepts only).

Text Books:

Data Communications and Networking, Fourth Edition by Behrouza A. Forouzan, TMH.

1.

Reference Books:

Computer Networks, A. S. Tanenbaum, 4th edition, Pearson Education.

1.

CORE – 9 PRACTICAL: COMPUTER NETWORKS LAB

Use C/C++/ any Network Simulator

1. Simulate Even Parity generator and checker.
2. Simulate two dimensional Parity generator and checker.
3. Simulate checksum generator and checker.
4. Simulate Hamming code method.
5. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
6. Simulate and implement stop and wait protocol for noisy channel.
7. Simulate and implement go back n sliding window protocol.
8. Simulate and implement selective repeat sliding window protocol.
9. Simulate and implement distance vector routing algorithm.

CORE-10: DATABASE SYSTEMS

OBJECTIVES

- To learn the fundamental elements of database system.
 - To learn the basic concepts of relational database management systems.
 - To learn various SQL commands.
-

Unit-1

Introduction to Database and Database Users, Database System Concepts and Architecture: data Models, schema, and instances, Conceptual Modeling and Database Design: Entity Relationship (ER) Model: Entity Types, Entity Sets, Attributes, Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, ER Naming Conventions. Enhanced Entity-Relationship (EER) Model.

Unit-2

Database Design Theory and Normalization: Functional Dependencies, Normal Forms based on Primary Keys, Second and third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

Unit-3

Relational data Model and SQL: Relational Model Concepts, Basic SQLs, SQL Data Definition and Data types, Constraints in SQL, Retrieval Queries in SQL, INSERT, DELETE, UPDATE Statements in SQL, Relational Algebra and Relational Calculus: Unary Relational Operations: SELECT and PROJECT, Binary Relation: JOIN and DIVISION.

Unit-4

Introduction to Transaction Processing Concepts and Theory: Introduction to Transaction Processing, Transaction and System Concepts, Properties of Transactions, Recoverability, Serializability, Concurrency Control Techniques, Locking techniques for Concurrency Control, Concurrency Control based on Time-Stamp Ordering.

Text Book:

1. Fundamentals of Database Systems, 6th edition, Ramez Elmasri, Shamkant B. Navathe, Pearson Education

Reference Book:

1. An Introduction to Database System, Date C. J. - Pearson Education, New Delhi - 2005

CORE-10 PRACTICAL: DATABASE SYSTEMS LABS

Create and use the following database schema to answer the given queries.

EMPLOYEE Schema

Field	Type	NULL	KEY	DEFAULT
Eno	Char(3)	NO	PRI	NIL
Ename	Varchar(50)	NO		NIL
Job_type	Varchar(50)	NO		NIL
Manager	Char(3)	Yes	FK	NIL
Hire_date	Date	NO		NIL
Dno	Integer	YES	FK	NIL
Commission	Decimal(10,2)	YES		NIL
Salary	Decimal(7,2)	NO		NIL

DEPARTMENT Schema

Field	Type	NULL	KEY	DEFAULT
Dno	Integer	No	PRI	NULL
Dname	Varchar(50)	Yes		NULL
Location	Varchar(50)	Yes		New Delhi

Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than Rs2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of Rs.1500 and Rs.2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is 'A'.
14. Query to display Name of all employees either have two 'R's or have two 'A's in their name and are either in Dept No = 30 or their Mangers Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.
16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns <Salary> monthly but wants <3*Current Salary>. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with 'J', 'A' and 'M'.

21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Department Name of all employees who have an 'A' in their name.
25. Query to display Name, Job, Department No. and Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manger's Name and the Manager's employee no; along with the Employees Name who do not have a Manager.
27. Query to display Name, Department No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies Rs100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees.
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department.

CORE – 11: WEB TECHNOLOGY

OBJECTIVES

- To learn the fundamentals of web designing.
 - To design and develop standard and interactive web pages.
 - To learn some popular web scripting languages.
-

Unit-1

Web Essentials: Clients, Servers and Communication:

The Internet – Basic Internet protocols – The WWW, HTTP request message – response message, web clients web servers – case study.

Introduction to HTML: HTML, HTML domains, basic structure of an HTML document –creating an HTML document, mark up tags, heading, paragraphs, line breaks, HTML tags. Elements of

HTML, working with text, lists, tables and frames, working with hyperlink, images and multimedia, forms and controls

Unit-2

Introduction to cascading style sheets: Concepts of CSS, creating style sheet, CSS properties, CSS styling (background, text format, controlling fonts), working with the block elements and objects. Working with lists and tables, CSS ID and class. Box model (introduction, border properties, padding properties, margin properties), CSS colour, grouping, Dimensions, display, positioning, floating, align, pseudo class, Navigation bar, image sprites.

Unit-3

JavaScripts: Client side scripting, what is JavaScript, simple JavaScript, variables, functions, conditions, loops and repetitions. JavaScripts and objects, JavaScript own objects, the DOM and web browser environment, forms and validations.

DHTML: Combining HTML, CSS, JavaScripts, events and buttons, controlling your browser.

Unit-4

PHP: Starting to script on server side, PHP basics, variables, data types, operators, expressions, constants, decisions and loop making decisions. Strings – creating, accessing strings, searching, replacing and formatting strings. Arrays: Creation, accessing array, multidimensional arrays, PHP with Database.

Text Book:

1. Web Technologies – Black Book – DreamTech Press
2. Matt Doyle, Beginning PHP 5.3 (Wrox-Wiley publishing)
3. John Duckett, Beginning HTML, XHTML, CSS and JavaScript.

Reference Book:

1. HTML, XHTML and CSS Bible, 5ed, Wiley India-Steven M. Schafer.

CORE – 11 PRACTICAL: WEB TECHNOLOGY LAB

1. Acquaintance with elements, tags and basic structure of HTML files.
2. Practicing basic and advanced text for formatting.
3. Practice use of image, video and sound in HTML documents.
4. Designing of web pages- Document layout, list, tables.
5. Practicing Hyperlink of web pages, working with frames.
6. Working with forms and controls.
7. Acquaintance with creating style sheet, CSS properties and styling.
8. Working with background, text, font, list properties.
9. Working with HTML elements box properties in CSS.

10. Develop simple calculator for addition, subtraction, multiplication and division operation using java script.
11. Create HTML page with java script which takes integer number as a input and tells whether the number is odd or even.
12. Create HTML page that contains form with fields name, Email, mobile number, gender, favorite colour and button; now write a java script code to validate each entry. Also write a code to combine and display the information in text box when button is clicked.
13. Write a PHP program to check if number is prime or not.
14. Write a PHP program to print first ten Fibonacci numbers.
15. Create a MySQL data base and connect with PHP.
16. Write PHP script for string and retrieving user information from my SQL table.
 - a. Write a HTML page which takes Name, Address, Email and Mobile number from user (register PHP).
 - b. Store this data in MySQL data base.
 - c. Next page display all user in HTML table using PHP (display .PHP).
17. Using HTML, CSS, Javascript, PHP, MySQL, design a authentication module of a web page.

CORE – 12: SOFTWARE ENGINEERING

OBJECTIVES:

- To learn the way of developing software with high quality and the relevant techniques.
 - To introduce software engineering principles for industry standard.
 - To focus on Project management domain and Software risks management.
-

Unit-1

Introduction: Evolution of Software to an Engineering Discipline, Software Development Projects, Exploratory Style of Software Development, Emergence of Software Engineering, Changes in Software Development Practices, Computer Systems Engineering.

Software Lifecycle Models: Waterfall Model and its Extensions, Rapid Application Development (RAD), Agile Development Models, Spiral Model.

Unit-2

Software Project Management: Software Project Management Complexities, Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO, Halstead's Software Science, Staffing Level Estimation, Scheduling, Organization and Team Structures, Staffing, Risk Management, Software Configuration Management.

Unit-3

Requirement Analysis and Specification: Requirements Gathering and Analysis, Software Requirement Specifications, Formal System Specification Axiomatic Specification, Algebraic Specification, Executable Specification and 4GL.

Software Design: Design Process, Characterize a Good Software Design, Cohesion and Coupling, Layered Arrangements of Modules, Approaches to Software Design (Function Oriented & Object-Oriented).

Unit-4

Coding and Testing: Coding: Code Review, Software Documentation, Testing, Unit Testing, Black Box and White Box Testing, Debugging, Program Analysis Tools, Integration Testing, System Testing, Software Maintenance.

Text Book:

1. Fundamental of Software Engineering, Rajib Mall, Fifth Edition, PHI Publication, India.

Reference Books:

1. Software Engineering– Ian Sommerville, 10/Ed, Pearson.
2. Software Engineering Concepts and Practice – Ugrasen Suman, Cengage Learning India Pvt, Ltd.
3. R. Misra, C. Panigrahi, B. Panda: Principles of Software Engineering & System Design, YesDee Publication

CORE – 12 PRACTICAL: SOFTWARE ENGINEERING LAB

S. No.	Practical Title
1.	<ul style="list-style-type: none">• Problem Statement,• Process Model
2.	Requirement Analysis: <ul style="list-style-type: none">• Creating a Data Flow• Data Dictionary, Use Cases
3.	Project Management: <ul style="list-style-type: none">• Computing FP• Effort• Schedule, Risk Table, Timeline chart
4.	Design Engineering: <ul style="list-style-type: none">• Architectural Design• Data Design, Component Level Design
5.	Testing: <ul style="list-style-type: none">• Basis Path Testing

Sample Projects:

1. **Criminal Record Management:** Implement a criminal record management system for jailers, police officers and CBI officers.
2. **Route Information:** Online information about the bus routes and their frequency and fares
3. **Car Pooling:** To maintain a web based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation system
8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

CORE – 13: COMPUTER GRAPHICS

OBJECTIVES

- To be able to learn the core concepts of Computer Graphics.
 - To be able to create effective programs for solving graphics problems.
-

Unit-1

Computer Graphics: A Survey of Computer graphics, Overview of Graphics System: Video Display Devices, Raster-Scan Systems, Input Devices, Hard-Copy Devices, Graphics Software.

Unit-2

Graphics Output Primitives: Point and Lines, Algorithms for line, circle & ellipse generation, Filled-Area Primitives. Attributes of Graphics Primitives: Point, line, curve attributes, fill area attributes, Fill methods for areas with irregular boundaries.

Unit-3

Geometric Transformations (both 2-D & 3-D): Basic Geometric Transformations, Transformation Matrix, Types of transformation in 2-D and 3-D Graphics: Scaling, Reflection, shear transformation, rotation, translation. 2-D, 3-D transformation using homogeneous coordinates.

Unit-4

Two Dimensional Viewing: Introduction to viewing and clipping, Viewing transformation in 2-D, Viewing pipeline, Clipping Window, Clipping Algorithms: Point clipping, Line clipping and Polygon clipping.

Text books

1. Mathematical Elements for Computer Graphics, D. F. Rogers & J. A. Adams, MGH, 2/ed.
2. Donald Hearn & M. Pauline Baker, “Computer Graphics with OpenGL”, Pearson Education.

Reference books

1. D. Hearn and M. Baker, “Computer Graphics with Open GL”, Pearson, 2/ed.
2. D. F. Rogers, “Procedural Elements for Computer Graphics”, MGH

CORE – 13 PRACTICAL: COMPUTER GRAPHICS LAB**Develop the programs using C/C++ or Java**

1. Write a program to implement Bresenham’s line drawing algorithm.
2. Write a program to implement mid-point circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to fill a polygon using Scan line fill algorithm.
6. Write a program to apply various 2D transformations on a 2D object (use homogenous coordinates).
7. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.

CORE – 14: NUMERICAL TECHNIQUES

OBJECTIVES:

- To learn various numerical techniques.
 - To be able to implement different numerical techniques using programming language.
-

Unit-1

Floating point representation and computer arithmetic, Significant digits, Errors: Round-off error, Local truncation error, Global truncation error, Order of a method, Convergence and terminal conditions, Efficient computations.

Unit-2

Bisection method, Secant method, Regula–Falsi method Newton–Raphson method, Newton’s method for solving nonlinear systems.

Unit-3

Interpolation: Lagrange’s form and Newton’s form Finite difference operators, Gregory Newton forward and backward differences Interpolation Piecewise polynomial interpolation: Linear interpolation.

Unit-4

Numerical integration: Trapezoid rule, Simpson's rule (only method), Newton–Cotes formulas, Gaussian quadrature, Ordinary differential equation: Euler's method Modified Euler's methods, Runge-Kutta second methods

Text books

1. S.S. Sastry, "Introductory Methods of Numerical Analysis", EEE , 5/ed.
2. M.K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Methods for Scientific and Engineering Computation, New Age International Publisher, 6/e (2012)

Reference books

1. Numerical Analysis: J. K. Mantri & S. Prahan, Laxmi Publication.
2. Introduction to Numerical Analysis, Josef Stoer and Roland Bulirsch, Springer.

CORE – 14 PRACTICAL: NUMERICAL TECHNIQUES LAB

Implement using C/ C++ or MATLAB/ Scilab

1. Find the roots of the equation by bisection method.
2. Find the roots of the equation by secant/Regula–Falsi method.
3. Find the roots of the equation by Newton's method.
4. Find the solution of a system of nonlinear equation using Newton's method.
5. Find the solution of tri-diagonal system using Gauss Thomas method.
6. Find the solution of system of equations using Jacobi/Gauss-Seidel method.
7. Find the cubic spline interpolating function.
8. Evaluate the approximate value of finite integrals using Gaussian/Romberg integration.
9. Solve the boundary value problem using finite difference method.

DSE-1: UNIX PROGRAMMING

OBJECTIVES:

- To learn the basics of UNIX OS, UNIX commands and File system.
 - To familiarize students with the Linux environment.
 - To learn fundamentals of shell scripting and shell programming.
 - To be able to write simple programs using UNIX.
-

Unit-1

Introduction: Unix Operating systems, Difference between Unix and other operating systems, Features and Architecture, Installation, Booting and shutdown process, System processes (an

overview), External and internal commands, Creation of partitions in OS, Processes and its creation phases – Fork, Exec, wait, exit.

Unit-2

User Management and the File System: Types of Users, Creating users, Granting rights, User management commands, File quota and various file systems available, File System Management and Layout, File permissions, Login process, Managing Disk Quotas, Links (hard links, symbolic links)

Unit-3

Shell introduction and Shell Scripting: Shell and various type of shell, Various editors present in Unix, Different modes of operation in vi editor, Shell script, Writing and executing the shell script, Shell variable (user defined and system variables), System calls, Using system calls, Pipes and Filters.

Unit-4

Unix Control Structures and Utilities: Decision making in Shell Scripts (If else, switch), Loops in shell, Functions, Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep).

Text Books:

1. Sumitabha, Das, Unix Concepts And Applications, Tata McGraw-Hill Education, 2017, 4/Ed.

Reference Books:

1. Nemeth Synder & Hein, Linux Administration Handbook, Pearson Education, 2010, 2/ Ed.

DSE–1 PRACTICAL: UNIX PROGRAMMING LAB

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify “cal” command to display calendars of the specified months.
3. Write a shell script to modify “cal” command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message “Entered login name is invalid”.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of “who” command along with the total number of users.
7. Write a shell script to display the multiplication table of any number.

8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
9. Write a shell script to find the sum of digits of a given number.
10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
11. Write a shell script to find the LCD (least common divisor) of two numbers.
12. Write a shell script to perform the tasks of basic calculator.
13. Write a shell script to find the power of a given number.
14. Write a shell script to find the greatest number among the three numbers.
15. Write a shell script to find the factorial of a given number.
16. Write a shell script to check whether the number is Armstrong or not.

DSE-2: DATA MINING

OBJECTIVES:

- To introduce the basic concepts of data warehousing, data mining, Issues, and Implication.
 - To learn the core topics like Association rules, Classification & Prediction and Clustering techniques.
 - To make a study on the Applications and Trends in Data Mining.
-

Unit-1

Data Warehouse Fundamentals: Introduction to Data Warehouse, OLTP Systems, OLAP, Differences between OLTP and OLAP, Characteristics of Data Warehouse, Functionality of Data Warehouse, Advantages and Applications of Data Warehouse, Advantages, Applications, Top-Down and Bottom-Up Development Methodology, Tools for Data warehouse development, Data Warehouse Types, Data cubes

Unit-2

Introduction to Data Mining: Data mining, Functionalities, Data Preprocessing: Preprocessing the Data, Data cleaning, Data Integration and Transformation, Data reduction, Discretization and Concept hierarchies.

Unit-3

Mining Association Rules: Basics Concepts – Single Dimensional Boolean Association Rules from Transaction Databases, Multilevel Association Rules from transaction databases, Multi

dimension Association Rules from Relational Database and Data Warehouses. Apriori Algorithm, FP-Tree algorithm

Unit-4

Classification and Prediction: Introduction, Issues, Decision Tree Induction, Naïve Bayesian Classification, Classification based on Concepts from Association Rule Mining, Classifier Accuracy.

Text Books:

1. J. Han and M. Kamber, Data Mining Concepts and Techniques, Elsevier, 2011

Reference Books:

1. K.P. Soman ,Shyam Diwakar, V.Ajay ,2006, Insight into Data Mining Theory and Practice, Prentice Hall of India Pvt. Ltd - New Delhi.
2. Data Mining Techniques, Arun K. Pujari, Universities Press, 2006
3. Modern Approaches of Data Mining: Theory & Practice, M. Panda, S. Dehuri, M. R. Patra, Narosa Publishing House, 2018.

DSE–2 PRACTICAL: DATA MINING LAB

Using Scilab/ MATLAB/ C/ Python/ R

1. Build a Data Warehouse and perform its operations.
2. Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets.
3. Demonstrate performing classification on data sets.
4. Demonstrate performing clustering on data sets.
5. Demonstrate performing Regression on data sets.
6. Credit Risk Assessment. Sample Programs using German Credit Data.
7. Sample Programs using Hospital Management System.

DSE-3: DATA SCIENCE

OBJECTIVES:

- To learn emerging issues related to various fields of data science.
 - To understand the underlying principles of data science, exploring data analysis.
 - To learn the basics of R Programming.
-

Unit-1

Data Scientist's Tool Box: Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio.

Unit-2

R Programming Basics: Overview of R, R data types and objects, reading and writing data, Control structures, functions, scoping rules, dates and times, Loop functions, debugging tools, Simulation, code profiling.

Unit-3

Getting and Cleaning Data: Obtaining data from the web, from APIs, from databases and from colleagues in various formats, basics of data cleaning and making data "tidy".

Unit-4

Exploratory Data Analysis: Essential exploratory techniques for summarizing data, applied before formal modeling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize high-dimensional data.

Text Books

1. Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontline" by Schroff/O'Reilly, 2013.

Reference Books

1. Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to Know About Data Mining and Data-Analytic Thinking by O'Reilly, 2013.
2. John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight" by John Wiley & Sons, 2013.
3. Eric Segel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die", 1st Edition, by Wiley, 2013.

DSE-3 PRACTICAL: DATA SCIENCE LAB

1. Write a program that prints "Hello World" to the screen.
2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n
3. Write a program that prints a multiplication table for numbers up to 12.
4. Write a function that returns the largest element in a list.
5. Write a function that computes the running total of a list.
6. Write a function that tests whether a string is a palindrome.
7. Implement linear search.
8. Implement binary search.
9. Implement matrices addition, subtraction and Multiplication
10. Fifteen students were enrolled in a course. Their ages were:

20 20 20 20 20 21 21 21 22 22 22 22 23 23 23

- i. Find the median age of all students under 22 years
- ii. Find the median age of all students
- iii. Find the mean age of all students
- iv. Find the modal age for all students
- v. Two more students enter the class. The age of both students is 23. What is now mean, mode and median?

DSE-4: PROJECT WORK / E-COMMERCE

DSE-4: E-COMMERCE

OBJECTIVES

- To introduce the concepts of *electronic commerce*.
 - To make the user understand how *electronic commerce* is affecting business enterprises, governments, consumers and people in general.
-

Unit-1

Introduction to E-Commerce: Definition and scope of E-Commerce and M-Commerce, E-Commerce trade cycle, Electronic Markets, Internet Commerce, Benefits and Impacts of E-Commerce.

Elements of E-Commerce: Various elements, e-visibility, e-shops, Delivery of goods and services, Online payments, After-sales services, Internet E-Commerce security.

Unit-2

EDI and Electronic Payment Systems: Introduction and definition of EDI, EDI layered Architecture, EDI technology and standards, EDI communications and transactions, Benefits and applications of EDI with example, Electronic Payment Systems: credit/debit/smart cards, e-credit accounts, e-money.

Unit-3

Introduction to EC models: Inter-organization and intra-organization E-Commerce, E-Commerce Models: B2B, B2C, C2B, C2C, G2C, C2G

E-Business: Introduction to Internet bookshops, Grocery Suppliers, Software Supplies and support, Electronic newspapers, Virtual auctions, Online share dealing, e-diversity.

Unit-4

E-Security and Legal Issues: Security concerns in E-Commerce, Privacy, integrity, authenticity, non-repudiation, confidentiality, SSL, Digital Signatures and fire walls, IT Act 2000, Cyber-crimes and cyber laws

Mobile Commerce and Future of E-Commerce: Introduction to Mobile Commerce, Benefits of Mobile Commerce, Impediments of M-Commerce, M-Commerce framework, Emerging and future trends.

Text Books:

1. G.S.V.Murthy, E-Commerce Concepts, Models, Strategies, Himalaya Publishing House.
2. Henry Chan, Raymond Lee, Tharam Dillon, Elizabeth Chang, "E-Commerce Fundamentals and Applications, Wiley Student Edition.

Reference Books:

1. Gray P. Schneider , Electronic commerce, International Student Edition.

DSE-4 Practical: E-Commerce Tutorial**SEC – 1: PYTHON PROGRAMMING**

OBJECTIVES:

- To enable the students to understand the basic principles of the Python Language.
 - To use the tools to do simple programs in python.
-

Unit-1

Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation.

Unit-2

Techniques of Problem Solving: Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.

Unit-3

Overview of Programming: Structure of a Python Program, Elements of Python

Introduction to Python: Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator)

Unit-4

Creating Python Programs: Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments.

Text Books

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011

Reference Books

1. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012

Online References:

1. Python Tutorial/Documentation www.python.or 2015

2. <http://docs.python.org/3/tutorial/index.html>
3. <http://interactivepython.org/courselib/static/pythonds>
4. <http://www.ibiblio.org/g2swap/byteofpython/read/>

Software Lab based on Python Programming:

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users choice.
2. Write a Program to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria:
 - Grade A: Percentage ≥ 80
 - Grade B: Percentage ≥ 70 and < 80
 - Grade C: Percentage ≥ 60 and < 70
 - Grade D: Percentage ≥ 40 and < 60
 - Grade E: Percentage < 40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. Write a Program to display the first n terms of Fibonacci series.
5. Write a Program to find factorial of the given number.
6. Write a Program to find sum of the following series for n terms: $1 - 2/2! + 3/3! - \dots - n/n!$
7. Write a Program to calculate the sum and product of two compatible matrices.

SEC-2: ANDROID PROGRAMMING

OBJECTIVES:

- To learn the basics of Android Programming.
 - To develop simple Android applications.
-

Unit-1

Introduction: History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture.

Unit-2

Overview of object oriented programming using Java: OOPs Concepts: Inheritance, Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java Virtual Machine.

Unit-3

Development Tools: Installing and using Eclipse with ADT plug-in, Installing Virtual machine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creating an android project – Hello Word, run on emulator, Deploy it on USB-connected Android device.

User Interface Architecture: Application context, intents, Activity life cycle, multiple screen sizes.

Unit-4

User Interface Design: Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners (Combo boxes), Images, Menu, Dialog.

Database: Understanding of SQLite database, connecting with the database.

Text Books:

1. Android application development for java programmers. By James C. Sheusi. Publisher: Cengage Learning, 2013.

Reference Book:

1. James C. Sheusi, “Android application Development for Java Programmers”, Cengage Learning, 2013.
2. M. Burton, & D. Felker, “Android Application Development for Dummies”, 2/e, Wiley India.

Online References:

1. <http://www.developer.android.com>
2. <http://docs.oracle.com/javase/tutorial/index.htm> (Available in the form of free downloadable ebooks also).
3. <http://developer.android.com/guide/components/fundamentals.html>
4. <http://developer.android.com/training/multiscreen/screensizes.html>
5. <http://developer.android.com/guide/topics/ui/controls.html>

Software Lab based on Android Programming:

1. Create “Hello World” application. That will display “Hello World” in the middle of the screen in the emulator. Also display “Hello World” in the middle of the screen in the Android Phone.
2. Create an application with login module. (Check username and password).
3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.
4. Create a menu with 5 options and selected option should appear in text box.
5. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.
6. Create an application with three option buttons, on selecting a button colour of the screen will change.
7. Create and Login application as above. On successful login, pop up the message.
8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.

AEC–1: Environmental Science

AEC–2: English Communication/MIL

GE/IC–1: PRINCIPLES OF MANAGEMENT

OBJECTIVES

- To understand the basic principles of management.
 - To provide a basis of understanding towards working of business organization through the process of management.
-

Unit-1

Nature of Management: Meaning, Definition, it's nature purpose, importance & Functions, Management as Art, Science & Profession- Management as social System Concepts of management-Administration-Organization.

Evolution of Management Thought: Contribution of F.W.Taylor, Henri Fayol ,Elton Mayo, Chester Barhard & Peter Drucker to the management thought. Various approaches to management (i.e. Schools of management thought)Indian Management Thought.

Unit-2

Functions of Management (Part-I)

Planning - Meaning - Need & Importance, types levels– advantages & limitations, Forecasting - Need & Techniques, Decision making - Types - Process of rational decision making & techniques of decision making,

Organizing - Elements of organizing & processes: Types of organizations, Delegation of authority - Need, difficulties in delegation – Decentralization,

Unit-3

Functions of Management (Part-II)

Staffing - Meaning & Importance, Direction - Nature – Principles, Communication - Types & Importance, Motivation - Importance – theories, Leadership - Meaning - styles, qualities & functions of leaders

Controlling-Need, Nature, importance, Process & Techniques, Coordination - Need, Importance.

Unit-4

Strategic Management

Definition, Classes of Decisions, Levels of Decision, Strategy, Role of different Strategist, Relevance of Strategic Management and its Benefits, Strategic Management in India.

Text Books:

1. Horold Koontz and Iteinz Weibrich, Essential of Management, McGrawhills International
2. K.Aswhathapa, Essential of Business Administration, Himalaya Publishing House

Reference Books:

1. L.M.Parasad Principles & practice of management - Sultan Chand & Sons - New Delhi
2. Tripathi, Reddy, Principles of Management, Tata McGraw Hill

GE/IC–1 TUTORIAL/PRACTICAL: PRINCIPLES OF MANAGEMENT TUTORIAL

GE/IC–2: STATISTICS

OBJECTIVES

- To understand the concept of population and sample.
- To use frequency distribution to make decision.
- To understand and to calculate various types of averages and variation.

Unit-1

Statistical Methods: Definition and scope of Statistics, concepts of statistical population and sample. **Data:** quantitative and qualitative, attributes, variables, scales of measurement nominal, ordinal, interval and ratio. **Presentation:** tabular and graphical, including histogram.

Unit-2

Measures of Central Tendency: mathematical and positional. **Measures of Dispersion:** range, quartile deviation, mean deviation, standard deviation, coefficient of variation, Moments, absolute moments, factorial moments, skewness and kurtosis, Sheppard's corrections.

Unit-3

Bivariate data: Definition, scatter diagram, simple, partial and multiple correlation (3-variables only), rank correlation. Simple linear regression.

Unit-4

Principle of least squares and fitting of polynomials and exponential curves. Theory of attributes: Independence and association of attributes, consistency of data, measures of association and contingency, Yule's coefficient of colligation.

Text Books:

1. S.C. Gupta, Fundamentals of Statistics, Sultan chand & sons, Delhi.
2. A.M.Goon, M.K.Gupta and B. Dasgupta, Fundamentals of Statistics, The World Press, Kolkata.

Reference Books:

1. S.P. Gupta, Statistical Methods, Sultan Chand and sons New Delhi

GE/IC-2 TUTORIAL/PRACTICAL: STATISTICS LAB

List of Practicals

1. Graphical representation of data.
2. Problems based on measures of central tendency.
3. Problems based on measures of dispersion.
4. Problems based on combined mean and variance and coefficient of variation.
5. Problems based on moments, skewness and kurtosis.
6. Fitting of polynomials, exponential curves.
7. Karl Pearson's correlation coefficient.
8. Correlation coefficient for a bivariate frequency distribution.
9. Lines of regression, angle between two lines of regression and estimated values of variables.
10. Spearman rank correlation with and without ties.
11. Partial and multiple correlations.
12. Planes of regression and variances of residuals for given simple correlations.

GE/IC-3: BUSINESS ACCOUNTING

OBJECTIVES

- To impart the basic business accounting knowledge.
-

Unit-1

Introduction: Financial Accounting-definition and Scope, objectives of Financial Accounting, Accounting v/s Book Keeping terms used in accounting, users of accounting information and limitations of Financial Accounting.

Conceptual Framework: Accounting Concepts, Principles and Conventions, Accounting Standards concept, objectives, benefits, brief review of Accounting Standards in India, Accounting Policies, Accounting as a measurement discipline, valuation Principles, accounting estimates

Unit-2

Recording of transactions: Voucher system; Accounting Process, Journals, Subsidiary Books, Ledger, Cash Book, Bank Reconciliation Statement, Trial Balance.

Depreciation: Meaning, need & importance of depreciation, methods of charging depreciation.

Unit-3

Preparation of final accounts: Preparation of Trading and Profit & Loss Account and Balance Sheet of sole proprietary business

Unit-4

Introduction to Company Final Accounts: Important provisions of Companies Act, 1956 in respect of preparation of Final Accounts, Understanding of final accounts of a Company.

Computerized Accounting: Computers and Financial application, Accounting Software packages, An overview of computerized accounting system - Salient features and significance, Concept of grouping of accounts, Codification of accounts, Maintaining the hierarchy of ledger, Generating Accounting Reports.

Text Books :

1. Anil Chowdhry, "Fundamentals of Accounting & Financial Analysis", Pearson Education
2. Rajesh Agarwal, R. Srinivasan, "Accounting Made Easy", TMH

Reference Books:

1. Amrish Gupta, "Financial Accounting for Management", Pearson Education
2. S. N. Maheshwari, "Financial Accounting for Management: Vikas Publishing House

GE/ IC-3 PRACTICAL/TUTORIAL: BUSINESS ACCOUNTING TUTORIAL

GE/IC-4: BUSINESS ECONOMICS

OBJECTIVES:

- To introduce the economic concepts.
 - To familiarize with the students the importance of economic approaches in managerial decision making.
 - To understand the applications of economic theories in business decisions.
-

Unit-1:

Demand, Supply and Market equilibrium: individual demand, market demand, individual supply, market supply, market equilibrium; Elasticity of demand and supply: Price elasticity of demand, income elasticity of demand, cross price elasticity of demand, elasticity of supply; Theory of consumer behavior: cardinal utility theory, ordinal utility theory (indifference curves, budget line, consumer choice, price effect, substitution effect, income effect for normal, inferior and giffen goods), revealed preference theory.

Unit-2:

Producer and optimal production choice: optimizing behavior in short run (geometry of product curves, law of diminishing margin productivity, three stages of production), optimizing behavior in long run (isoquants, isocost line, optimal combination of resources) Costs and scale: traditional theory of cost (short run and long run, geometry of cot curves, envelope curves), modern theory of cost (short run and long run), economies of scale, economies of scope.

Unit-3:

Theory of firm and market organization: perfect competition (basic features, short run equilibrium of firm/industry, long run equilibrium of firm/industry, effect of changes in demand, cost and imposition of taxes); monopoly (basic features, short run equilibrium, long run equilibrium, effect of changes in demand, cost and imposition of taxes, comparison with perfect competition, welfare cost of monopoly), price discrimination, multiplant monopoly; monopolistic competition (basic features, demand and cost, short run equilibrium, long run equilibrium, excess capacity); oligopoly (Cournot's model, kinked demand curve model, dominant price leadership model, prisoner's dilemma)

Unit-4:

Factor market: demand for a factor by a firm under marginal productivity theory (perfect competition in the product market, monopoly in the product market), market demand for a factor, supply of labour, market supply of labour, factor market equilibrium.

Text Books:

1. Yogesh Maheswari, Managerial Economics, PHI Learning, New Delhi.
2. G. S. Gupta, Managerial Economics, Tata Mcgraw-Hill, New Delhi.

Reference Books:

1. Moyer & Harris, Managerial Economics, Cengage Learning, New Delhi.

2. Geetika, Ghosh & Choudhury, Managerial Economics, Tata Mcgrawhill, New Delhi.
3. Dominick Salvatore, Principles of Microeconomics, Oxford University Press, (5th Ed.)

GE/IC-4 TUTORIAL/PRACTICAL: BUSINESS ECONOMICS TUTORIAL EQUIPMENT

1. Desktop Computer

Core i5 (minimum 8th Generation Processor, 8 GB RAM, 2 TB HDD)

Number of Desktops: 30 (or as per student strength). It must be connected through structured Local Area Network (LAN).

2. Software

LibreOffice, Scilab, C, C++, Java, Assembler, VHDL, Linux/ Unix Prolog etc. , preferably Open Source Software.

FACULTY TRAINING

Most of the Colleges are offering BCA under self-financing mode. Faculty training is required for all Core Courses as well as Discipline Specific Elective Courses.

Faculty training may be organized for the following Courses in phased manner (six month before the beginning of the Subject in the concerned semester).

- i. Digital Logic
- ii. Computer Organization
- iii. Data Structure
- iv. Operating System
- v. Database Systems
- vi. Java Programming
- vii. Web Technology
- viii. Unix Shell Programming
- ix. Data Mining
- x. Data Science
- xi. Programming in Python
- xii. Android Programming

UTKAL UNIVERSITY

Syllabus for Integrated Masters of Computer Applications (IMCA)-4th&5th Years

(This is being followed by the Model Syllabus of BCA Programme of Higher Education, Govt. of Odisha)

Syllabus for Integrated Masters of Computer Applications (IMCA)-4th&5th Years				
PAPER ID	PAPER TITLE	FULL MARK		CREDIT
		Mid-Term	End-Term	
SEVENTH SEMESTER				
IMCA-7.1	Core Course-15 Combinatorics and Graph Theory	30	70	4
IMCA-7.2	Core Course-16 Compiler Design	30	70	4
IMCA-7.3	Core Course-17 Distributed System	30	70	4
IMCA-7.4	Core Course-18 Network Security	30	70	4
IMCA-7.5	Discipline Specific Elective-5	30	70	4
IMCA-7.6	Compiler Design Lab		50	2
IMCA-7.7	Network Security Lab		50	2
EIGHTH SEMESTER				
IMCA-8.1	Core Course-19 Algorithms Design and Analysis	30	70	4
IMCA-8.2	Core Course-20 Theory of Computation	30	70	4
IMCA-8.3	Core Course-21 Advanced Computer Architecture	30	70	4
IMCA-8.4	Core Course-22 Artificial Intelligence	30	70	4
IMCA-8.5	Discipline Specific Elective-6	30	70	4
IMCA-8.6	Algorithms Design and Analysis Lab		50	2
IMCA-8.7	Artificial Intelligence Lab		50	2
NINETH SEMESTER				
IMCA-9.1	Core Course-23 Cloud Computing	30	70	4
IMCA-9.2	Core Course-24 Internet of Things	30	70	4
IMCA-9.3	Core Course-25 Machine Learning	30	70	4
IMCA-9.4	Core Course-26 Block Chain Technology	30	70	4
IMCA-9.5	Discipline Specific Elective-7	30	70	4

IMCA-9.6	Internet of Things Lab		50	2
IMCA-9.7	Machine Learning Lab		50	2
TENTH SEMESTER				
IMCA-10.1	Project Work / Dissertation		300 *	12

List of Discipline Specific Elective Papers #

Discipline Specific Elective-5	
IMCA-7.5(1)	Multimedia and Applications
IMCA-7.5(2)	Bio-Informatics
IMCA-7.5(3)	Wireless Sensor Networks
Discipline Specific Elective-6	
IMCA-8.5(1)	Digital Image Processing
IMCA-8.5(2)	Human Computer Interaction
IMCA-8.5(3)	Optimization Techniques
Discipline Specific Elective-7	
IMCA-9.5(1)	Mobile Computing
IMCA-9.5(2)	Parallel Processing
IMCA-9.5(3)	Simulation and Modeling
<p>* Mark Distribution for Project Work / Dissertation: Report (200), Presentation (50), Viva Voce (50)</p> <p># A student can opt for only one paper from among the papers mentioned under the respective Discipline Specific Elective groups</p>	

IMCA-7.1: Combinatorics and Graph Theory

Objectives

The objective of the course is to explain basic concepts in combinatorial graph theory Define how graphs serve as models for many standard problems discuss the concept of graph, tree, Euler graph, cut set and Combinatorics. See the applications of graphs in science, business and industry.

Learning outcomes

Upon Completing the Course, Students will be able to:

- Understand and explain the basic concepts of graph theory.
- Apply the basic concepts of mathematical logic.
- Analyze the basic concepts of mathematical logic.
- Evaluate some real time problems using concepts of graph theory.

UNIT-I

Introductory concepts - What is graph – Application of graphs – finite and infinite graphs – Incidence and Degree – Isolated vertex, pendent vertex and Null graph. Paths and circuits – Isomorphism, sub graphs, walks, paths and circuits, Connected graphs, disconnect graphs.

UNIT-II

Euler graphs, Hamiltonian paths and circuits, Dirac's theorem for Hamiltonicity, Travelling salesman problem. Directed graphs – types of digraphs, Digraphs and binary relation. Trees – properties, pendent vertex, Distance and centres - Rooted and binary tree, counting trees, spanning trees.

UNIT-III

Vertex Connectivity, Edge Connectivity, Cut set and Cut Vertices, Fundamental circuits, Planar graphs, Different representation of planar graphs, Euler's theorem, Geometric dual, Combinatorial dual.

UNIT-IV

Matrix representation of graphs- Adjacency matrix, Incidence Matrix, Circuit matrix, Fundamental Circuit matrix and Rank, Cut set matrix, Path matrix. Graphs theoretic algorithms - Algorithm for computer representation of a graph, algorithm for connectedness and components, spanning tree, shortest path.

UNIT-V

Partitions, counting functions, number of partitions into odd or unequal parts. Necklaces, Euler's function, set of symmetries, enumeration in the odd and even cases.

Recommended Books:

1. C.L. Liu, Elements of Discrete Mathematics, Tata McGraw Hill, 2nd Edition, 2000.
2. Douglas B. West, Introduction to Graph Theory, Prentice Hall India Ltd., 2001
3. N. Deo, Graph Theory with Applications to Engineering and Computer Science, PHI publication, 3rd edition, 2009

IMCA-7.2: Compiler Design

Objectives:

- To provide a thorough understanding of the internals of Compiler Design.
- To explore the principles, algorithms, and data structures involved in the design and construction of compilers.
- Topics include context-free grammars, lexical analysis, parsing techniques, symbol tables, error recovery, code generation, and code optimization.

Learning outcomes:

Upon Completing the Course, Students will be able to:

- Realize basics of compiler design and apply for real time applications.
- Introduce different translation languages
- Understand the importance of code optimization
- Know about compiler generation tools and techniques

UNIT-I

INTRODUCTION TO COMPILERS: Definition of compiler, interpreter and its differences, the phases of a compiler, role of lexical analyzer, regular expressions, finite automata, from regular expressions to finite automata, pass and phases of translation, bootstrapping, LEX-lexical analyzer generator. PARSING: Parsing, role of parser, context free grammar, derivations, parse trees, ambiguity, elimination of left recursion, left factoring, eliminating ambiguity from dangling-else grammar, classes of parsing, top down parsing - backtracking, recursive descent parsing, predictive parsers, LL(1) grammars.

UNIT-II

BOTTOM UP PARSING: Definition of bottom up parsing, handles, handle pruning, stack implementation of shift-reduce parsing, conflicts during shift-reduce parsing, LR grammars, LR parsers-simple LR, canonical LR(CLR) and Look Ahead LR (LALR) parsers, error recovery in parsing, parsing ambiguous grammars, YACC-automatic parser generator.

UNIT-III

SYNTAX DIRECTED TRANSLATION: Syntax directed definition, construction of syntax trees, S-attributed and L-attributed definitions, translation schemes, emitting a translation.

INTERMEDIATE CODE GENERATION: intermediate forms of source programs– abstract syntax tree, polish notation and three address code, types of three address statements and its implementation, syntax directed translation into three-address code, translation of simple statements, Boolean expressions and flow-of-control statements.

UNIT-IV

TYPE CHECKING: Definition of type checking, type expressions, type systems, static and dynamic checking of types, specification of a simple type checker, equivalence of type expressions, type conversions, overloading of functions and operators. **RUN TIME ENVIRONMENTS:** Source language issues, Storage organization, storage-allocation strategies, access to non-local names, parameter passing, symbol tables and language facilities for dynamic storage allocation.

UNIT-V

CODE OPTIMIZATION: Organization of code optimizer, basic blocks and flow graphs, optimization of basic blocks, the principal sources of optimization, the directed acyclic graph (DAG) representation of basic block, global data flow analysis. **CODE GENERATION:** Machine dependent code generation, object code forms, the target machine, a simple code generator, register allocation and assignment, peephole optimization.

Recommended Books:

1. A.V Aho, M.S Lam, Ravi Sethi, and J.D Ullman, “Compilers, Techniques and Tools”, 2nd Edition, Pearson Publication.

IMCA-7.3: Distributed System

Objectives:

This Subject provides students with an in-depth knowledge about the operating system. The former treats the standard principles of single processor system, including processes, synchronization, I/O, deadlocks, Memory Management, File Management systems, security and so on. This subject covers distributed operating system in detail, including communication process, file system and memory management synchronization and so on but this time in the context of distributed systems.

Learning outcomes:

Upon Completing the Course, Students will be able to:

- Outline the potential benefits of distributed systems
- Summarize the major security issues associated with distributed systems along with the range of techniques available for increasing system security.
- Apply standard design principles in the construction of these systems
- Select appropriate approaches for building a range of distributed systems, including some that employ middleware

UNIT-I

Functions of an Operating System, Design Approaches, Review of Network Operating System and Distributed Operating System, Issue in the design of Distributed Operating System, Overview of Computer Networks, Modes of communication, System Process, Interrupt Handling, Handling Systems calls, Protection of resources, Micro-Kernel Operating System, client server architecture

UNIT-II

The Critical Section Problem, Other Synchronization Problems, Language Mechanisms for Synchronization, Axiomatic Verification of Parallel Programs, Inter process communication (Linux IPC Mechanism), Remote Procedure calls, RPC exception handling, security issues, RPC in Heterogeneous Environment, Case studies.

UNIT-III

Logical clocks, Physical clocks, Vector Clock, clock synchronization algorithms, Mutual Exclusion, Non-Token Based Algorithms – Lamport’s Algorithm, Token-Based Algorithms, Suzuki-Kasami’s Broadcast Algorithm, Election Algorithms, Dead locks in Distributed Systems, Thrashing, Resource Management (Load Balancing approach, Load Sharing approach), Process Management, process Migration, Thread, and Case studies.

UNIT-IV

Overview of shared memory, Architecture, Algorithm, Protocols, Design Issues, consistency model, Page based Distributed Shared Memory, Shared variable Distributed shared Memory, and Object based Distributed shared Memory, Heterogeneous DSM, Distributed Scheduling, Issues, Components, Algorithms Case studies.

UNIT-V

File models, File access, File sharing, file-caching, File Replication, Features of Naming system terminologies and concepts of naming, fault Tolerance, Network File System (case study), 8NFS on Linux Directory Services, Security in Distributed File system, Tools (Cuda, Swift, Globus, Condor, Amazon AWS, OpenStack, Cilk, gdb, threads, MPICH, OpenMP, Hadoop, FUSE), Case studies.

Recommended Books:

1. Andrew S. Tanenbaum & Maarten van Steen, Distributed Systems: Principles and Paradigms, Prentice-Hall(2002) ISBN0-13-088893-1
2. D. L. Galli, Distributed Operating Systems, Prentice-Hall(2000)
3. Distributed Operating Systems and Algorithms, Randy Chow, T. Johnson, Addison Wesley
4. Distributed Systems Concepts and Design, G. Coulouris, J. Dollimore, Addison Wesley

IMCA-7.4: Network Security

Objectives:

- Learn fundamentals of cryptography and its application to network security.
- Understand network security threats, security services, and countermeasures.
- Acquire background on well-known network security protocols such.
- Understand vulnerability analysis of network security.

Learning outcomes:

Upon Completing the Course, Students will be able to:

- Understand various Cryptographic Techniques
- Apply various public key cryptography techniques
- Implement Hashing and Digital Signature techniques
- Implement system level security applications

UNIT-I

Introduction: Security Goals, Cryptographic Attacks, Services and Mechanism, Techniques; Mathematics of Cryptography: Integer Arithmetic, Modular Arithmetic, Matrices; Symmetric-Key Ciphers: Substitution Ciphers, Transposition Ciphers, Stream and Block Ciphers.

UNIT-II

Groups, Rings, Field, $GF(2^n)$ Fields; Data Encryption Standard (DES); Advanced Encryption Standard (AES); Euler's Phi-Function, Chinese Remainder Theorem (CRT), Exponentiation and Logarithm.

UNIT-III

Asymmetric-Key Cryptography: RSA Cryptosystem; ElGamal Cryptosystem; Hash Function: MD, SHA; Digital Signature; Digital Signature Schemes: Digital Signature Standard (DSS).

UNIT-IV

Security at the Application Layer: E-mail, PGP, S/MIME; Security at the Transport Layer: Secure Socket Layer (SSL) Architecture, SSL Message Formats, Transport Layer Security (TLS).

UNIT-V

Security at the Network Layer: Two Modes, Two Security Protocols, Security Association, Security Policy; System Security: Users, Trust and Trusted Systems, Buffer Overflow and Malicious Software, Malicious Programs, Worms, Viruses, Intrusion Detection Systems (IDS), Firewalls.

Recommended Books:

1. Behrouz A. Frouzan: Cryptography and Network Security, **McGraw-Hill**.
2. William Stallings, "Cryptography and Network Security: Principals and Practice", Pearson Education.

IMCA-7.6: Compiler Design Lab

List of Experiments:

1. Implement a lexical analyzer in "C".
2. Use LEX tool to implement a lexical analyzer.
3. Implement a recursive descent parser for an expression grammar that generates arithmetic expressions with digits, + and *.
4. Use YACC and LEX to implement a parser for the a grammar.
5. Write semantic rules to the YACC program in problem 5 and implement a calculator that takes an expression with digits, + and * and computes and prints its value.
6. Implement the front end of a compiler that generates the three address code for a simple language with: one data type integer, arithmetic operators, relational operators, variable declaration statement, one conditional construct, one iterative construct and assignment statement.
7. Implement the back end of the compiler which takes the three address code generated in problems 7 and 8, and produces the 8086 assembly language instructions that can be assembled and run using a 8086 assembler. The target assembly instructions can be simple move, add, sub, jump. Also simple addressing modes are used.

IMCA-7.7: Network Security Lab

List of Experiments:

1. Implementation of Caesar Cipher technique.
2. Implement the Play fair Cipher.
3. Implement the Pure Transposition Cipher.
4. Implement DES Encryption and Decryption.
5. Implement the AES Encryption and decryption.
6. Implement RSA Encryption Algorithm.
7. Implementation of Hash Functions

IMCA-8.1: Algorithms Design and Analysis

Objectives:

- To have a mathematical foundation in analysis of algorithm
- To expose students to the basic concepts of algorithm design and analyze its complexity.
- To develop algorithms for problems relating to sorting, database queries optimization, graph networks, transport network, etc.

Learning outcomes:

Upon Completing the Course, Students will be able to:

- Design algorithms for various applications such as Robotics, Artificial Intelligence, Machine learning, Computer Networks, Parallel computing, etc.
- Use the appropriate data structure in context of solution of given problem.
- Develop programming skills which require solving given problems.

UNIT-I

Analysis and Design of Algorithm (Case study insertion sort and merge sort) Asymptotic Analysis, Divide and Conquer, Recurrence Relations, Strassen's Matrix Multiplication.

UNIT-II

Sorting: Quick sort, heap sort, Counting sort, lower bound for sorting, Randomized quicksort, Order Statistics.

UNIT-III

Amortized Analysis (Aggregate analysis, accounting analysis, Potential analysis), 2-3-4 tree
Advanced Data structure: Fibonacci heap, Redblack tree, hashing, data structure on disjoint set, Suffix Array Data Structure.

UNIT-IV

Dynamic Programming: Matrix Chain multiplication, LCS, TSP, Branch and Bound. Greedy Algorithm: MST: Krushkal, Prims, Dijkstra Algorithm, Huffman Coding, Maxflow matching, Computational geometry: Convex Hull, 0-1-knaplock, fractional knapsack, Back tracking (4-Queen Prob.)

UNIT-V

Complexity Class: P, PSPACE, NP, NP-Hard, NP Complete, Satisfiability, Cheque, Vertex Cover, Independent set, Exact cover, Graph Coloring, Hamiltonian, Cycle Matching. Approximation Algorithm: Vertex Cove, TSP, Independent Set, Sum of subset.

Recommended Books:

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein "Introduction to Algorithms", PHI, 3/e, 2009.
2. Sarabasse & A.V. Gelder Computer Algorithm, "Introduction to Design and Analysis", Pearson 3/e, 1999
3. E. Horowitz, S. Sahni, & S. Rajasekaran, "Fundamentals of Computer Algorithms", 2/e, University Press.
4. A.V. Aho, J.E. Hopcroft, & J.D. Ullman, "The Design and Analysis of Computer Algorithm, Pearson.

IMCA-8.2: Theory of Computation

Objectives:

- To learn the mathematical foundations of computation including automata theory
- To learn the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability.
- To learn about how really computers works and what kind of activities can be computed mechanically within a computer.

Learning outcomes:

Upon Completing the Course, Students will be able to:

- Model, compare and analyse different computational models using combinatorial methods.
- Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.

- Construct algorithms for different problems and argue formally about correctness on different restricted machine models of computation.
- Identify limitations of some computational models and possible methods of proving them.
- Have an overview of how the theoretical study in this course is applicable of application like designing the compilers.

UNIT-I

Regular Language and Finite State Machine, Non-Deterministic Finite State Machine, Equivalent Automata, Regular Expression, Properties of regular languages, Pumping Lemma and its application, Minimization of Finite Automata, Decision Properties of regular languages, Linear Grammar.

UNIT-II

Context Free languages and Push Down Machine, Non-Deterministic Push Down Machine, CFG and Derivation Trees, Properties of CFL, Chomsky Normal Form, Pumping Lemma CFL, Decision Properties of CFL.

UNIT-III

Context Sensitive Languages, Turing Machine, Variants of Turing Machines, Non-Deterministic Turing machine, Enumerators, Decidable languages, Decidable problems concerning regular and context free languages. Halting Problem, Post Correspondence Problem.

UNIT-IV

Undecidable Problem, Reducibility, Undecidable problems about Turing Machine, Rice's Theorem and its application, Mapping reducibility, Computable function

UNIT-V

Complexity Theory, NP-completeness, Space Complexity, Savitch's theorem, PSPACE, PSPACE Completeness, The class L and NL, NL completeness, Intractability, Probabilistic algorithms.

Recommended Books:

1. Michael Sipser, "INTRODUCTION TO THE THEORY OF COMPUTATION"
2. J.E. Hopcroft, R.Motwani and J.D.Ullman, "INTRODUCTION TO AUTOMATA THEORY, LANGUAGES, AND COMPUTATION"

IMCA-8.3: Advanced Computer Architecture

Objective:

The aim of this module is to emphasize on the concept of a complete system consisting of asynchronous interactions between concurrently executing hardware components and device driver software in order to illustrate the behavior of a computer system as a whole.

Learning Outcomes:

At the end of this course student will be able to:

- Understand the Concept of Parallel Processing and its applications
- Implement the Hardware for Arithmetic Operations
- Analyze the performance of different scalar Computers
- Develop the Pipelining Concept for a given set of Instructions
- Distinguish the performance of pipelining and non-pipelining environment in a processor

UNIT-I

Fundamentals of Computer Design: Introduction, The Task of a Computer designer, Technology and usage trends, Cost, Performance measures. Quantitative principles of computer design, The concept of memory hierarchy.

Instruction set Principles of Examples: Introduction, Classification of Instruction Set Architectures, Memory Addressing, Operations in the Instruction Set, The DLX Architecture, Examples, Power PC and Intel 8086

UNIT-II

Pipelining: Basic concepts, Pipeline for DLX, Pipeline Hazards, Data and Control Hazards, Difficulties in implementation, Instruction Set Design and pipelining. The MIPS R 4000 Pipeline.

UNIT-III

Instruction – Level Parallelism (ILP): Concepts and challenges, Overcoming Data Hazards with Dynamic Scheduling, Reducing Branch Penalties with Dynamic Hazards Prediction, ILP with multiple issue, Hardware Support for Extracting more parallelism, Studies of ILP, The power PC 620.

UNIT-IV

Memory–Hierarch Design : Basic concepts of memory, Internal Organisation of Memory chips, Caches, Reducing Cache Misses and Miss Penalty, Reduction Hit Time, Main Memory, Virtual memory, Issues in the design of Memory Hierarchies, fallacies and Pitfalls in Memory – Hierarchy Design.

Storage System: Types of storage devices, Busses, I/O Performance Measures, Reliability, Availability and RAID, Interfacing to OS, Designing and I/O System, Unix File System Performance.

UNIT-V

Multiprocessor : Taxonomy of Parallel Architecture, Performance Matrices and Advantages for communications Mechanisms, Challenges of Parallel Processing, Characteristics of Application Domains, Centralized Shared memory architectures, Distributed shared – memory architectures, synchronization, Models of memory, Consistency, Performance measurement of Parallel machines, Memory system Issues, Design and performance of SGI challenges Multiprocessor.

Recommended Books:

1. J. L. Hennessy & D. A. Patterson – Computer Architecture – A Quantitative Approach, 2nd Edition – Morgan Kaufmann Publishers, 1996 [chapters 1, 2, 3, 4, 5, 6, 7, 8]
2. V. C. Hammacher, Z. G. Vranesic, S. G. Zaky – Computer Organisation, McGraw Hill, 1996 [chapters 1, 2, 3, 4, 5, Appendix A].
3. M. M. Mano – Computer System Architecture, 3rd Edition, PHI, 1993.
4. K. Hwang – Advanced Computer Architecture, McGraw Hill, 1993.
5. D. Sima, T. Fountain, P. Kacsak – Advanced Computer Architecture – A Design Space Approach, Addison Wesley, 1997.
6. D. A. Patterson & J. L. Hennessy – Computer Organisation & Design, Morgan Kaufmann, 1998.

IMCA-8.4: Artificial Intelligence

Objectives:

- To learn the basic concepts of AI principles and approaches.
- To develop the basic understanding of the building blocks of AI.

Learning outcomes:

Upon Completing the Course, Students will be able to:

- Have fundamental understanding of the basic concepts of artificial intelligence (AI).
- Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
- Have fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- Have knowledge of current scope and limitations, and societal implications of AI.
- Have basic foundation of machine learning.

UNIT-I

Introduction to Artificial Intelligence: The Imitation Game and Turing Test, Lady Lovelace's Objection, Learning Machines, the Evolution of AI, Water-Jug problem, Missionaries-Cannibals Problem, Monkey-Bananas Problem, Tower of Hanoi, Tower of Brahma, Cryptarithmic problem, Production System, Working Memory, Inference Engine.

State Space Search Algorithms: Depth First Search, Breadth First Search, Iterative Deepening, Uniform Cost Search, Hill-Climbing, Beam Search, Tabu Search. Heuristic Search: Greedy Best-First Search, A* algorithm, 8-Puzzle Problem, Iterative Deepening A*(IDA*) algorithm, Problem Decomposition, And-Or Graph, AO* algorithm, Rete Net, Rete algorithm, Simulated Annealing, Genetic Algorithms.

UNIT-II

Game Playing Strategies, Game Tree, MIN-MAX, Tic-Tac-Toe, Alpha-Beta Pruning, SSS* algorithm, B* Search. Constraint Satisfaction Problems, N-Queen problem.

Knowledge Representation: Types of Knowledge, presentation by Logic, propositional logic, substitutional rules, Rules of Inference: Modus Ponens, Modus Tollens, Resolution in Propositional Logic, Refutation, Predicate Logic, First Order Logic Representation, Skolemization, Natural Deduction, Resolution in Predicate Logic, Clause Form, Prenex Normal Form, Unification.

UNIT-III

Knowledge Representation using Rules: PROLOG, Facts, Rules, Defining Relations by rules and facts, Recursive rules, Atoms, Variables, Data & Structured Objects, Lists: Membership, Concatenation, adding to & Deleting from List, sublist, permutations, Operators in PROLOG, Cuts, problems with Cuts, Managing Input & Output in PROLOG, Control Facilities, Database Manipulation, bagof, setoff, and findall. Forward Reasoning, Backward Reasoning, Matching, Statistical Reasoning, Bayesian Belief Network, Dempster-Shafer Theory, Fuzzy Logic.

UNIT-IV

Planning: Planning Agent, PDDL, STRIPS approach, TWEAK approach, Blocks World Problem.

Machine learning paradigms: Introduction, machine learning systems, supervised and unsupervised learnings, inductive learning, deductive learning, clustering, support vector machines, Decision Tree, case-based reasoning and learning, Artificial neural networks: Introduction, artificial networks, single layer feed forward networks, multi layered forward networks, design issues of artificial neural networks.

UNIT-V

Natural Language Processing: Steps in NLP, Parsing, Augmented Transition Network, Semantic Analysis, Discourse and Pragmatic Processing.

Word Level Analysis: Unsmoothed N-grams, Evaluating N-grams, Skip-Gram, Word2vec, Smoothing, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

Spelling Checking: Spelling Errors, Spell checking Techniques, Soundex Algorithm.

Recommended Books:

1. "Artificial Intelligence", by Rich E, Knight K and Nair S. B., 3rd Edition, McGrawHill, 2013.
2. "A First Course in Artificial Intelligence", by Deepak Khemani, 1st Edition, McGrawHill, 2013.
3. "Artificial Intelligence: Structure and Strategies for Complex Problem solving", by George F Luger, 4th Edition, Pearson Education, 2004.
4. "Artificial Intelligence: A Modern Approach", by Russell S. J., Norvig P., 3rd Edition, Pearson, 2015.
5. "PROLOG programming for artificial intelligence", by Ivan Bratko, 3rd Edition, Addison-Wesley, 2001.
6. "Programming in PROLOG", by Clocksin W.F and Mellish C.S, Springer-Verlag, 5th Edition, 2003.

IMCA-8.6: Algorithms Design and Analysis Lab

List of Experiments:

1. Basics: Find out Big - Oh and Big – Omega of the function. Take necessary data like degree of the function, coefficients, etc.
2. Write a program to implement: a Queue, a Stack, a Queue using two Stacks and a Stack using two Queues.
3. Write an algorithm and find the efficiency of the same for following problems:
 - i. Finding Factorial – Iterative Approach
 - ii. Finding Factorial – Recursive Approach
 - iii. Printing Fibonacci Series – Iterative Approach
4. Design an algorithm and implement a program for Insertion Sort, Selection Sort, Bubble Sort, Count Sort and Linear Search.
5. Design an algorithm and implement a program for Merge Sort, Quick Sort and Binary Search.
6. Design an algorithm and implement a program to solve Knapsack Problem using greedy approach.
7. Design an algorithm and implement a program to solve Knapsack Problem and Finding Optimal Matrix Chain Order Problem using dynamic programming approach.

8. Design an algorithm and implement a program to solve Longest Common Subsequence Problem
9. Design an algorithm and write a program to implement DFS and BFS.

IMCA-8.7: Artificial Intelligence Lab

List of Experiments:

1. Write a program in prolog to implement simple facts and Queries.
2. Write a program in prolog to implement simple arithmetic.
3. Write a program in prolog to solve Monkey banana problem.
4. Write a program in prolog to solve Tower of Hanoi.
5. Write a program in prolog to solve 8 Puzzle problems.
6. Write a program in prolog to solve 4-Queens problem.
7. Write a program in prolog to solve Traveling salesman problem.
8. Write a program in prolog for Water jug problem.
9. Write a program to implement a Tic-Tac-Toe game.
10. Write a python program to implement simple Chatbot?

IMCA-9.1: Cloud Computing

Objective:

- Identify the technical foundations of cloud systems architectures.
- Analyze the problems and solutions to cloud application problems.
- Apply principles of best practice in cloud application design and management.
- Identify and define technical challenges for cloud applications and assess their importance.

Learning Outcome:

- Understand the fundamental principles of distributed computing.
- Understand how the distributed computing environments known as Grids can be built from lower level services.
- Understand the importance of virtualization in distributed computing and how this has enabled the development of Cloud Computing.
- Analyze the performance of Cloud Computing.
- Understand the concept of Cloud Security.
- Learn the Concept of Cloud Infrastructure Model.

UNIT-I

Overview of Computing Paradigm: P2P, Grid, Cluster, Distributed, Utility, and Cloud Computing; Evolution of computing, adopting cloud computing in business; Introduction to cloud computing: NIST Model, properties, characteristics, benefits, role of open standards.

UNIT-II

Cloud Computing Architecture: The cloud computing stack (Client/Server, Protocols, Web services), Service Models (XaaS): IaaS, PaaS, SaaS; Deployment Models: Public, Private, Hybrid, Community; Platform as a Service: Overview of PaaS, Service Oriented Architecture (SOA); cloud platform and management, computation and storage; Case Study: Google App Engine, Microsoft Azure, Salesforce.com - platform and use for public.

UNIT-III

Infrastructure as a Service: IaaS definition, Introduction to Virtualization, Hypervisors, Machine Image, Virtual Machine; Resource Virtualization of server, storage, network; Case Study: Amazon EC2- Renting, EC2 Compute Unit, platforms and storage, pricing, Customer Service Provisioning; Eucalyptus.

UNIT-IV

Software as a Service: Introduction to SaaS, Web services, Web 2.0, Web OS, Case Study on SaaS; Service Management: Service Level Agreements, Billing & Accounting, Comparing Scaling Hardware- Traditional vs. Cloud, Economics of Scaling, Managing Data- Database & Data Stores in Cloud, large scale data processing.

UNIT-V

Cloud Security: Infrastructure Security, Network level, Host level, Application Level, Data Security, Identity and Access Management, Access Control, Trust, Reputation, Risk; Authentication: Client access in Cloud, Cloud contracting model, commercial and business considerations, Service level maintenance in Infrastructure, Network, Host and Application; Case Study: Eucalyptus, Microsoft Azure, Amazon EC2.

Recommended Books:

1. R. Buyya, J. Broberg, and A. Goscinski, Cloud Computing: Principles and Paradigms, 2nd Edition, Wiley, 2011.
2. G. Shroff, Enterprise Cloud Computing - Technology, Architecture, Applications, University Press, 2010.
3. R. L. Krutz and R. D. Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley-India, 2010. L. Wasserman-All of statistics

IMCA-9.2: Internet of Things

Objective:

The Internet is evolving to connect people to physical things and also physical things to other physical things all in real time. It's becoming the Internet of Things (IoT). The course enables

student to understand the basics of Internet of things and protocols. It introduces some of the application areas where Internet of Things can be applied. Students will learn about the middleware for Internet of Things. To understand the concepts of Web of Things.

Learning Outcome:

At the end of this course student will be able to:

- Able to understand the application areas of IoT.
- Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks.
- Able to understand building blocks of Internet of Things and characteristics.

UNIT-I

IoT - What is the IoT and why is it important? Elements of an IoT ecosystem, Technology drivers, Business drivers, Trends and implications, Overview of Governance, Privacy and Security Issues.

UNIT-II

IoT PROTOCOLS - Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFID Protocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE802.15.4–BACNet Protocol– Modbus – KNX – Zigbee– Network layer – APS layer – Security

UNIT-III

IoT ARCHITECTURE - IoT Open source architecture (OIC)- OIC Architecture & Design principles- IoT Devices and deployment models- IoTivity : An Open source IoT stack - Overview- IoTivity stack architecture- Resource model and Abstraction.

UNIT-IV

WEB OF THINGS - Web of Things versus Internet of Things – Two Pillars of the Web – Architecture Standardization for WoT– Platform Middleware for WoT – Unified Multitier WoT Architecture – WoT Portals and Business Intelligence.

UNIT-V

IoT APPLICATIONS - IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications. Study of existing IoT platforms /middleware, IoT- A, Hydra etc.

Recommended Books:

1. Honbo Zhou, “The Internet of Things in the Cloud: A Middleware Perspective”, CRC Press,2012.
2. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), “Architecting the Internet of Things”, Springer, 2011.
3. David Easley and Jon Kleinberg, “Networks, Crowds, and Markets: Reasoning About a HighlyConnected World”, Cambridge University Press, 2010.

4. Olivier Hersent, David Boswarthick, Omar Elloumi , “The Internet of Things – Key applications and Protocols”, Wiley, 2012.

IMCA-9.3: Machine Learning

Objective:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To develop skills of using recent machine learning software for solving practical problems.
- To gain experience of doing independent study and research.

Learning Outcome:

At the end of this course student will be able to:

- Have a good understanding of the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.
- Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
- Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
- Be able to design and implement various machine learning algorithms in a range of real-world applications.

UNIT-I

Linear Methods for Regression and Classification: Overview of supervised learning, Linear regression models and least squares, Multiple regression, Subset selection , Ridge regression, least angle regression and Lasso , Linear Discriminant Analysis , Logistic regression , Naïve Bayes Classifier. K-nearest neighbor.

UNIT-II

Model Assessment and Selection :Bias, Variance, and model complexity, Bias-variance trade off, Optimism of the training error rate, Estimate of In-sample prediction error, Effective number of parameters, Bayesian approach and B. IC, Cross-validation ,Boot strap methods, conditional or expected test error. Dimensionality reduction (PCA, Kernel PCA, Independent Component analysis, LLE, non-negative matrix factorization, SVD), feature Selection.

UNIT-III

Additive Models, Trees, and Boosting: Generalized additive models, Regression and classification trees , Boosting methods-exponential loss and AdaBoost, Numerical Optimization

via gradient boosting , Examples (Spam data, California housing , New Zealand fish, Demographic data)

UNIT-IV

Support Vector Machines (SVM) and Neural network: SVM for classification, Reproducing Kernels, SVM for regression. Model of a neuron, Perceptron learning, Backpropagation, Methods to improve backpropagation, Introduction to deep learning.

UNIT-V

Unsupervised Learning and Random forests: Cluster analysis (k-means, Hierarchical clustering, DBSCAN, spectral clustering), Gaussian mixtures and EM algorithm, Random forests and analysis.

Recommended Books:

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman , The Elements of Statistical Learning- Data Mining, Inference, and Prediction, Second Edition , Springer Verlag, 2009.
2. G. James, D.Witten, T.Hastie, R.Tibshirani - An introduction to statistical learning with applications in R, Springer, 2013.G. Booch, I. Jacobson, J. Ramburg – UML User Guide, Pearson Education
3. C. M. Bishop –Pattern Recognition and Machine Learning, Springer, 2006
4. L. Wasserman-All of statistics

IMCA-9.4: Block Chain Technology

Objective:

This course is intended to study the basics of Blockchain technology. During this course learner will explore various aspects of Blockchain technology like application in various domains. By implementing learner will have idea about private and public Blockchain, and smart contract?

Learning Outcome:

At the end of this course student will be able to:

- Understand and explore the working of Blockchain technology (Understanding)
- Analyze the working of Smart Contracts (Analyze)
- Understand and analyze the working of Hyperledger (Analyze).
- Apply the learning of solidity and de-centralized apps on Ethereum (Apply).

UNIT-I

Introduction of Cryptography and Blockchain: What is Blockchain, Blockchain Technology Mechanisms & Networks, Blockchain Origins, Objective of Blockchain, Blockchain Challenges, Transactions And Blocks, P2P Systems, Keys As Identity, Digital Signatures, Hashing, and public key cryptosystems, private vs. public Blockchain.

UNIT-II

BitCoin and Cryptocurrency: What is Bitcoin, The Bitcoin Network, The Bitcoin Mining Process, Mining Developments, Bitcoin Wallets, Decentralization and Hard Forks, Ethereum Virtual Machine (EVM), Merkle Tree, Double-Spend Problem, Blockchain and Digital Currency, Transactional Blocks, Impact of Blockchain Technology on Cryptocurrency?

UNIT-III

Introduction to Ethereum: What is Ethereum, Introduction to Ethereum, Consensus Mechanisms, How Smart Contracts Work, Metamask Setup, Ethereum Accounts, Receiving Ether's What's a Transaction?, Smart Contracts.

UNIT-IV

Introduction to Hyperledger: What is Hyperledger? Distributed Ledger Technology & its Challenges, Hyperledger & Distributed Ledger Technology, Hyperledger Fabric, Hyperledger Composer.

UNIT-V

Solidity Programming: Solidity - Language of Smart Contracts, Installing Solidity & Ethereum Wallet, Basics of Solidity, Layout of a Solidity Source File & Structure of Smart Contracts, General Value Types (Int, Real, String, Bytes, Arrays, Mapping, Enum, address), Blockchain Applications

Recommended Books:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press (July 19, 2016).
2. Antonopoulos, Mastering Bitcoin.
3. Antonopoulos and G. Wood, Mastering Ethereum.
4. D. Drescher, Blockchain Basics. Apress, 2017.

IMCA-9.6: Internet of Things Lab

List of Experiments:

1. Introduction to Arduino platform and programming.
2. Interfacing Arduino to Zigbee module.
3. Interfacing Arduino to GSM module.
4. Interfacing Arduino to Bluetooth Module.
5. Introduction to Raspberry PI platform and python programming.
6. Interfacing sensors to Raspberry PI.
7. Communicate between Arduino and Raspberry PI using any wireless medium.
8. Setup a cloud platform to log the data.
9. Log Data using Raspberry PI and upload to the cloud platform.
10. Design an IOT based system.

IMCA-9.7: Machine Learning Lab

List of Experiments:

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

9. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.