BACHELORS WITH COMPUTER APPLICATIONS AS MAJOR

2nd SEMESTER

CAP222J: COMPUTER APPLICATIONS _ PROGRAMMING FUNDAMENTALS THROUGH C

CREDITS: THEORY: 4; PRACTICAL: 2

COURSE OBJECTIVES:

1. *To demonstrate the use of flowcharts and algorithms for problem solving*
2. *To introduce the concepts of structured programming*
3. *To familiarize the student with the syntactic constructs of C*
4. *To enable to the students to translate algorithms into C programs*

THEORY (4 CREDITS)

UNIT – I **(15 Lectures)**

Programming Languages: History and Role of Programming Languages, Syntax and semantics, source code and object code, datatypes, variables, constants, declaration, Structured Data Types. Sequence Control: Implicit and Explicit. Sequence control between Statements. Subprogram Control: Simple call return and recursive subprogram. Language Paradigms: Simple Procedural Languages, Block Structured Programming Languages, Object Based Languages, Functional Languages, Logic Programming Languages.

Flowcharts, Flowchart Elements, Problem Solving Through Flowcharts.

Algorithms, Characteristics of an Algorithm, Algorithms for basic problems.

UNIT – II

Introduction to C Programming **(15 Lectures)**

History and overview of C, Basic structure of a C Program, Compilation, Execution and Debugging of programs in C.

Keywords, Identifiers and Datatypes. Variables and Constants. Comments. Console I/O using printf() and scanf(). Typecasting.

Operators – Arithmetic, Logical, Relational, Increment Decrement and Assignment Operators. Expressions. Operator Precedence.

Conditional Statements (If, If-Else, If-Else If, Nested If, Switch).

UNIT – III **(15 Lectures)**

Looping, Functions and Pointers.

Loops (while, do-while, for). Break and Continue. Nested Loops.

Functions: Declaring, Defining and Calling. Call by Value, Call by Reference. Function Arguments and Return Values.

Pointers: Declaring and Initializing. Accessing value of a pointer variable. Pointer Expressions. Pointer Increments and Scale Factors. Pointers and Arrays. Passing Pointers to Functions.

UNIT – IV

Arrays, Strings, Structures and Unions

Declaring, Initializing 1-D arrays and 2-D arrays. Accessing Elements of an Array, Memory Layout of Arrays. Passing Arrays to Functions, Command Line Arguments.

Character Arrays and String. Declaring and Initializing Strings, Reading and Writing Strings, String Handling Function (strlen, strcat, strcmp, strcpy).

Structures and Unions: Declaring, initializing and using simple structures and unions; Manipulating individual members of structures and unions, Array of structures, Passing structures to functions.

Dynamic Memory Allocation using malloc and free.

TEXTBOOKS:

1. Terence Pratt, Programming Languages Design and Implementation (Pearson/Prentice Hall)
2. Balagurusamy, Programming in ANSI C, 8th Edition (McGraw Hill)

REFERENCES:

1. Kanetkar - Let us C (BPB Publications)
2. Ghezzi, Jazayeri - Programming Language Concepts (Wiley)
3. Srivastava – C in Depth (BPB Publications)
4. Beej's Guide to C Programming
5. Byron Gottfried - Schaum's Outline of Programming with C (McGraw-Hill)

LABORATORY COURSE (02 CREDITS):

1. WAP to print the sum of digits of an integer.
2. WAP to print the product of digits of an integer.
3. WAP to reverse a number.
4. WAP to compute the sum of the first n terms of the following series S = 1+1/2+1/3+1/4+……
5. WAP to compute the sum of the first n terms of the following series S =1-2+3-4+5…….
6. Write a program to check whether a given string is Palindrome or not. Convert this program into to a function that checks if a given string is a palindrome.
7. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
8. WAP to compute the factors of a given number.
9. Write a macro that swaps two numbers. WAP to use it.
10. WAP to print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
*******
*********
```

11. WAP to perform following actions on an array entered by the user:

i) Print the even-valued elements
ii) Print the odd-valued elements
iii) Calculate and print the sum and average of the elements of the array

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.

12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. 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.
15. Write a program to create an array of user-defined size dynamically using malloc() function. Display the sum of values entered in it and use free() to release its memory.
16. Write a 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) Reverse the string

***

BACHELORS WITH APPLIED COMPUTING AS MINOR

(FOR STUDENTS WITH MAJOR IN COMPUTER APPLICATIONS / INFORMATION TECHNOLOGY)

2nd SEMESTER

ACP222N: APPLIED COMPUTING _ WEB DESIGNING

CREDITS: 4 + 2

UNIT-I

Markup Languages, Introduction to HTML5, Development Environment Setup, Anatomy of an HTML Tag, Basic Structure of HTML Document, HTML Content Models, Meta-Tags, Formatting Tags, Text Level Formatting, Lists, Hyperlinks, Image and Image Maps, Table Tags, HTML Comment tag. Block and inline elements, redirecting to another URL, creating division-based layouts. Forms: creating basic form, using check boxes, textboxes and option buttons, input validation and additional input types in HTML5, HTML multimedia basics. HTML DOM structure.

UNIT II:

Need for CSS. Different approaches to style sheets, Anatomy of a CSS Rule. Element, Class, and ID Selectors. Combining Selectors, Pseudo-Class Selectors. Style Placement, Conflict Resolution, Styling Text Wildcard Selectors (*, ^ and $) in CSS. Web fonts. Working with Browser Developer Tools.CSS Box Model: - background, margin, padding, Float and z-index properties, Relative and Absolute Element Positioning. Basic Introduction to Bootstrap Framework.

UNIT-III

Introduction to Javascript, Different approaches to place Javascript code in an HTML File. JS identifiers, Reserved Words, Optional Semicolons, Comments, Literals. Types, Values and Variables: Numbers, Text, Booleans. Nulls and undefined. Type Conversions. Variable Declaration and Assignment. Const, let and var.
Expressions and Operators: Arithmetic, Relational, Logical, Assignment and Evaluation Expressions. Conditionals: if, else if and switch. Loops: while and for. Break, continue, return and yield.
Functions: Defining, Invoking, Function Arguments and Parameters. Functions as Values.

UNIT-IV

Objects: Creating Objects, Querying and setting Properties, Deleting and Testing Properties. Serializing Objects. Arrays: Creating, Reading, Writing arrays. Array length. Iterating Arrays, Strings as Arrays.
The Document Object Model, Program Input and Output, Browser Events and Event Handling.

RECOMMENDED BOOKS:

1. Jennifer Robbins - Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (5e, 2018, O'Reilly Media)
2. Terry Felke-Morris - Web Development & Design Foundations with HTML5 (8e, 2017, Pearson)
3. Eric Meyer, Estelle Weyl - CSS The Definitive Guide (4e, 2018, O'Reilly Media)
4. David Sawyer McFarland - CSS The Missing Manual (4e, 2015, O'Reilly Media)
5. David Flanagan - JavaScript_ The Definitive Guide (7e, 2020, O'Reilly Media)
6. Cay S. Horstmann - Modern JavaScript for the Impatient (Addison-Wesley Professional, 2020)

WEB DESIGNING LAB

1. Design a Home page for your college
2. Design a web page with links to different pages and allow navigation between web pages.
3. Design a web page using Images
4. Use a HTML table to design a page with a header, sidebar, main content and footer.
5. Design a user registration form using different HTML form controls
6. Design a web page with buttons that can handle different page events using JS event handlers.
7. Use Java Script to change the image displayed in an img tag when a button on the page is clicked.
8. Use bootstrap to add formatting to your home page.
9. Write a JavaScript program with proper GUI to perform unit conversion using the onChange event.
10. Design the interface of a login page using HTML and CSS.
11. Design a simple "To Do" Application using HTML/CSS/JavaScript.
12. Design Basic Calculator using HTML/CSS/JavaScript.
13. Design and develop a simple "Tic-Tac-Toe Game" using HTML/CSS/JavaScript.
14. Remove a specific table row using Java Script.
15. Set value in input text using Java Script.
16. Set a value in a span using Java Script.

BACHELORS WITH COMPUTER APPLICATIONS AS MINOR

(FOR STUDENTS WITH SUBJECTS OTHER THAN COMPUTER APPLICATION / INFORMATION TECHNOLOGY AS MAJOR)

2nd SEMESTER

CAP222N: COMPUTER APPLICATIONS _ PROGRAMMING FUNDAMENTALS THROUGH C

CREDITS: THEORY: 4; PRACTICAL: 2

COURSE OBJECTIVES:

1. *To demonstrate the use of flowcharts and algorithms for problem solving*
2. *To introduce the concepts of structured programming*
3. *To familiarize the student with the syntactic constructs of C*
4. *To enable to the students to translate algorithms into C programs*

THEORY (4 CREDITS)

UNIT – I **(15 Lectures)**

Programming Languages: History and Role of Programming Languages, Syntax and semantics, source code and object code, datatypes, variables, constants, declaration, Structured Data Types. Sequence Control: Implicit and Explicit. Sequence control between Statements. Subprogram Control: Simple call return and recursive subprogram. Language Paradigms: Simple Procedural Languages, Block Structured Programming Languages, Object Based Languages, Functional Languages, Logic Programming Languages.

Flowcharts, Flowchart Elements, Problem Solving Through Flowcharts.

Algorithms, Characteristics of an Algorithm, Algorithms for basic problems.

UNIT – II

Introduction to C Programming **(15 Lectures)**

History and overview of C, Basic structure of a C Program, Compilation, Execution and Debugging of programs in C.

Keywords, Identifiers and Datatypes. Variables and Constants. Comments. Console I/O using printf() and scanf(). Typecasting.

Operators – Arithmetic, Logical, Relational, Increment Decrement and Assignment Operators. Expressions. Operator Precedence.

Conditional Statements (If, If-Else, If-Else If, Nested If, Switch).

UNIT – III **(15 Lectures)**

Looping, Functions and Pointers.

Loops (while, do-while, for). Break and Continue. Nested Loops.

Functions: Declaring, Defining and Calling. Call by Value, Call by Reference. Function Arguments and Return Values.

Pointers: Declaring and Initializing. Accessing value of a pointer variable. Pointer Expressions. Pointer Increments and Scale Factors. Pointers and Arrays. Passing Pointers to Functions.

UNIT – IV

Arrays, Strings, Structures and Unions

Declaring, Initializing 1-D arrays and 2-D arrays. Accessing Elements of an Array, Memory Layout of Arrays. Passing Arrays to Functions, Command Line Arguments.

Character Arrays and String. Declaring and Initializing Strings, Reading and Writing Strings, String Handling Function (strlen, strcat, strcmp, strcpy).

Structures and Unions: Declaring, initializing and using simple structures and unions; Manipulating individual members of structures and unions, Array of structures, Passing structures to functions.

Dynamic Memory Allocation using malloc and free.

TEXTBOOKS:

1. Terence Pratt, Programming Languages Design and Implementation (Pearson/Prentice Hall)
2. Balagurusamy, Programming in ANSI C, 8th Edition (McGraw Hill)

REFERENCES:

1. Kanetkar - Let us C (BPB Publications)
2. Ghezzi, Jazayeri - Programming Language Concepts (Wiley)
3. Srivastava – C in Depth (BPB Publications)
4. Beej's Guide to C Programming

5. Byron Gottfried - Schaum's Outline of Programming with C (McGraw-Hill)

LABORATORY COURSE (02 CREDITS):

1. WAP to print the sum of digits of an integer.
2. WAP to print the product of digits of an integer.
3. WAP to reverse a number.
4. WAP to compute the sum of the first n terms of the following series S = 1+1/2+1/3+1/4+……
5. WAP to compute the sum of the first n terms of the following series S =1-2+3-4+5…….
6. Write a program to check whether a given string is Palindrome or not. Convert this program into to a function that checks if a given string is a palindrome.
7. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
8. WAP to compute the factors of a given number.
9. Write a macro that swaps two numbers. WAP to use it.
10. WAP to print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
*******
*********
```

11. WAP to perform following actions on an array entered by the user:

i) Print the even-valued elements
ii) Print the odd-valued elements
iii) Calculate and print the sum and average of the elements of the array

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.

12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. 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.
15. Write a program to create an array of user-defined size dynamically using malloc() function. Display the sum of values entered in it and use free() to release its memory.
16. Write a 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) Reverse the string

***

2nd SEMESTER
COMPUTER APPLICATIONS
(WEB DEVELOPER)
SKILL ENHANCEMENT COURSE (SEC)

WDP222S: JAVA SCRIPT AND CSS BASICS

CREDITS: THEORY: 2, PRACTICAL: 2

THEORY (2 CREDITS)

UNIT 1 – JAVA SCRIPT (15 LECTURES)

Introduction, Script Tag, Data Types, Variables, Literals, Expressions, Operators, Conditional Statements (if, if-else, if-else-if-else), switch-case, Looping Statements (while, for, do-while), Array, Associative Arrays, Functions, Event Handling, Javascript Objects (Browser, Document, Window etc.)

UNIT 2 – CSS (15 LECTURES)

DHTML introduction, Style Sheets-Embedded Styles, Inline Styles, External Style Sheets, Using Classes, Style Sheet Properties- Fonts Properties, Background and Colour Properties, Text Properties, Box Properties, Classification Properties-Display Property, Whitespace Property, , CSS Units, URL's , DIV and SPAN Tags, Dynamic Positioning, Layering, DHTML Events.

REFERENCE BOOKS:

1. Java Script Bible Wrox Publications
2. DHTML BPB Publications

PRACTICAL (2 CREDITS)

LAB SHEET- JAVA SCRIPT AND CSS BASICS

1. Write a JavaScript program to display the current day and time in the following format
2. Write a JavaScript program to get the current date
3. Write a JavaScript program that accept two integers and display the larger
4. Write a JavaScript conditional statement to sort three numbers. Display an alert box to show the result
5. Write JavaScript to demonstrate loops: while, for, do-while
6. Write a JavaScript for loop that will iterate from 0 to 15. For each iteration, it will check if the current number is odd or even, and display a message to the screen
7. Write a JavaScript function to check whether an `input` is an array or not
8. Write a JavaScript program to sort the items of an array
9. WAP to show blinking effect on a web page using JavaScript.
10. Write CSS for *Fonts, Background, Color, Text*
11. Design a digital clock using JavaScript and CSS.
12. Design a calculator using HTML & JavaScript.
13. Write a JavaScript program to demonstrate Event Handling.
14. WAP to validate Email Address in JavaScript.
15. Write a program to demonstrate exception handling in JS.

BACHELORS WITH DIGITAL AND TECHNOLOGICAL SOLUTIONS AS VALUE ADDED COURSE
1st to 2nd SEMESTERS

DTS024V: DIGITAL AND TECHNOLOGICAL SOLUTIONS

CREDITS: 02

COURSE OBJECTIVES:

- *To gain familiarity with digital paradigms;*
- *To sensitize about role & significance of digital technology;*
- *To provide know how of communications & networks;*
- *To bring awareness about the e-governance and Digital India initiatives;*
- *To provide a flavour of emerging technologies - Cloud, Big Data, AI, 3D printing.*

COURSE OUTCOME:

- *Knowledge about digital paradigm;*
- *Realization of importance of digital technology, digital financial tools, e-commerce;*
- *Know-how of communication and networks;*
- *Familiarity with the e-governance and Digital India initiatives;*
- *An understanding of use & applications of digital technology;*
- *Basic knowledge of machine learning and big data.*

COURSE CONTENTS:

UNIT I

Introduction & Evolution of Digital Systems. Role & Significance of Digital Technology. Information & Communication Technology & Tools. Computer System & it's working, Software and its types. Operating Systems: Types and Functions. Problem Solving: Algorithms and Flowcharts.

Communication Systems: Principles, Model & Transmission Media. Computer Networks & Internet: Concepts & Applications, WWW, Web Browsers, Search Engines, Messaging, Email, Social Networking. Computer Based Information System: Significance & Types. E-commerce & Digital Marketing: Basic Concepts, Benefits & Challenges.

UNIT II

Digital India & e-Governance: Initiatives, Infrastructure, Services and Empowerment. Digital Financial Tools: Unified Payment Interface, Aadhar Enabled Payment System, USSD, Credit / Debit Cards, e-Wallets, Internet Banking, NEFT/RTGS and IMPS, Online Bill Payments and PoS. Cyber Security: Threats, Significance, Challenges, Precautions, Safety Measures, & Tools, legal and ethical perspectives.

Emerging Technologies & their applications: Overview of Cloud Computing, Big Data, Internet of Things, Virtual Reality, Blockchain & Cryptocurrency, Robotics, Machine Learning & Artificial Intelligence, 3-D Printing. Digital Signatures.

TEXT BOOK

- F S Masoodi, Z S Masoodi and K B Dar, Digital and Technological Solutions, BPB Publications.

REFERENCE BOOKS

- V. Rajaraman, Introduction to Information Technology, 3rd Edition, PHI;
- E Balagurusamy, Fundamentals of Computers, Tata Mc GrawHill;
- Behrouz A. Forouzan, Data Communications and Networking, McGraw Hill;
- Pramod Kumar, Anuradha Tomar, R. Sharmila, Emerging Technologies in Computing Theory, Practice, and Advances, Edition 2021, Chapman and Hall/CRC Imprint;
- Buvya, Broberg, and Gosciniski, Cloud Computing- Principals and Paradigms, Wiley
- Russel and Norving, Artificial Intelligence- A Modern Approach, Pearson Education;
- Samuel Greengard, Internet of Things, MIT Press;
- C.S.V. Murthy, E-commerce Concepts, Models, Strategies;
- Hurwith, Nugent Halper, Kaufman, Big Data for dummies, Wiley & Sons - Wiley.

SEMESTER 1st to 3rd
MULTI-DISCIPLINARY COURSE
CAP022I COMPUTER APPLICATIONS (INTRODUCTION TO COMPUTERS)

CREDITS: 3

COURSE OBJECTIVES

1. *To introduce the fundamentals of computing devices and reinforce computer vocabulary, particularly with respect to personal use of computer hardware and software, the Internet, networking and mobile computing.*
2. *To provide hands-on use of Microsoft Office 2013 applications Word, Excel, Access and PowerPoint. Completion of the assignments will result in MS Office applications knowledge and skills.*
3. *To describe the organization and operation of a computer processor, primary and secondary memory, peripheral devices and to give computer specifications*

UNIT – I: COMPUTER BASICS

Introduction: Characteristics of Computer, Classification of Computers, Architecture and Chronology, Applications of Computer. Block Diagram of Computer.
Commonly used Terms: Hardware, Software, Firmware, Units of Measurement of Storage, Input/ Output Devices, Types of Memory, Generation of Computer Languages, and Introduction to Internet & E-Mail.

UNIT – II: OPERATING SYSTEM BASICS & GUI USING MS-WINDOWS.

Application Software and System Software, Open-Source Software and Proprietary Software.
Computer Languages and its types (Machine Language, Assembly Language, High Level Language) Translators, Compiler, Interpreter. Operating System and its functions.

UNIT –III: INTRODUCTION TO MS OFFICE

MS Word Basics: Basics of Word Processing, Text Selection, Opening Documents and Creating Documents, Saving Documents/Quitting Documents, Printing Documents. Using the Interface (Menu Toolbars), Editing Text (Copy, Delete, Move Etc.). Finding and replacing text. Special check Feature/ Auto correct Feature, Grammar check Facility, Formatting and Editing. Mail Merge, Bullets & Numbering, Borders and Shadings.

MS EXCEL BASIC:
Worksheet Workbook, Workspace Basics, Data Entry in cell, Entry of Numbers, Text and Formulate, Moving Data in the Worksheet, Selecting Data Range, Using the Interface (Toolbars, Menus), Editing basics, working with Workbooks Saving and Quitting, Cell Reference, Formatting, Editing.

MS POWER POINT BASICS:
Use of existing templates, fonts and drawing. Hands of MS PowerPoint, Creation of animated slides.

REFERENCE BOOKS:

1. Computer today, Donald H. Sanders, McGraw Hill Publishing Company.
2. Microcomputers Software and Applications, Dennis P. Curtin and Leslie R. Portel, PHI.
3. Data Processing: An Introduction, Donald P. Spencer and Charles R. Merril Pub. And Co.
4. Computers and Their Applications, Larry Joel Goldestein, PHI.
5. Windows-2000, kethy, Tata McGraw Hill Publishing Company.

BACHELORS WITH COMPUTER APPLICATIONS AS MAJOR
4th SEMESTER

CAP422J1 COMPUTER APPLICATIONS _ DBMS

CREDITS: THEORY (3) PRACTICAL (1)

Course Learning Outcomes:

- *To introduce the core concept of Relational Database.*
- *To enable students to design the databases for a wide variety of Real-World problems*
- *To introduce the concept and process of Database Normalization*
- *To enable the student to learn DML, DDL, DCL commands using SQL*

Unit I:

Introduction to Databases, Database Users, Characteristics of Database approach, Applications of DBMS, Advantages and Disadvantages, Database System Concepts and Architecture, Data Models, Schemas, Instances, Three-Schema Architecture and Data Independence, Database System, Centralized and Client/Server Architecture for DBMS,

Unit II:

Data Modelling using Entity-Relationship Mode, Entity Types, Entity Sets, Attributes, and Keys, Relationship Types, Relationship Sets, ER Diagrams, Relational Data Model and Relational Model Constraints, Functional Dependencies, Normalization: 1NF, 2NF, 3NF

Unit III:

Relational Algebra and Relational Calculus, Introduction to DDL, DML, DCL. Introduction to transactions, Transaction states ACID properties, Concurrency Control Techniques: 2-phase locking, time stamp ordering.

TEXTBOOK:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010

REFERENCES:

1. An introduction to database systems by Desai, Bipin C
2. PL/SQL by Ivan Bayross

DBMS- LAB

Creation of Tables in SQL (DDL)

1. Overview of using SQL tool, Data types in SQL,
2. Create Schema and insert at least 5 records for each table.
3. Queries for Altering Tables (with add and modify clause) , Dropping Tables
4. Queries for DROP RENAME TRUNCATE and Describe a Table
5. SQL Constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT

Practicing DML Commands

6. Insert, Select, Update, Delete
7. Select Command with WHERE, IN, BETWEEN AND LIKE clause
8. SELECT with AND, OR, NOT
9. Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING.

Joining Tables

10. Practice the Inner Join, Left Join,
11. Practice the Right Join, Cross Join

Create the following Database Schema and Practice the queries

EMP (eno, ename, bdate, title, salary, dno)
PROJECT (pno, pname, budget, dno)
DEPT (dno, dname, mgreno)
WORKSON (eno, pno, resp, hours)

12. Write an SQL query that returns the project number and name for projects with a budget greater than 100,000.
13. Write an SQL query that returns all works on records where hours worked is less than 10 and the responsibility is 'Manager
14. Write an SQL query that returns the employees (number and name only) who have a title of 'EE' or 'SA' and make more than 35,000.
15. Write an SQL query that returns the employees (name only) in department 'D1' ordered by decreasing salary.
16. Write an SQL query that returns the departments (all fields) ordered by ascending department name.
17. Write an SQL query that returns the employee's name, department name, and employee title.
18. Write an SQL query that returns the project name, hours worked, and project number for all works on records where hours > 10.
19. Write an SQL query that returns the project name, department name, and budget for all projects with a budget < 50,000.
20. Write an SQL query that returns the employee numbers and salaries of all employees in the 'Consulting' department ordered by descending salary.
21. Write an SQL query that returns the employee's name, project name, employee title, and hours for all works on records.

BACHELORS WITH COMPUTER APPLICATIONS AS MAJOR

4th SEMESTER

CAP422J2 COMPUTER APPLICATIONS _ OOPs with C++

CREDITS: THEORY (4) PRACTICAL (2)

COURSE LEARNING OUTCOME:

The student should understand object-oriented programming and C++ concepts, improve his/her problem-solving skills and should:

- *Be able to explain the difference between object-oriented programming and procedural programming.*
- *Be able to program using C++ features such as composition of objects, operator overloading, inheritance and polymorphism, file I/O, etc.*
- *Be able to build C++ classes using appropriate encapsulation and design principles.*
- *Be able to apply object-oriented techniques to solve bigger computing problems.*

UNIT-1 (15 HOURS)

Introduction to Object Oriented Programming

Comparison of Procedural Programming and OOP, Benefits of OOP, Abstraction, Encapsulation, Inheritance, Polymorphism, Difference between C and C++.

Elements of C++ Language:

Tokens and identifiers, Variables and Constants, Reference variables, Basic data types in C++, Streams in C++. Types of operators in C++.

Decision and Control Structures:

if statement, if-else statement, switch statement, Loop: while, do-while, for. Break and continue.

Pointers and structures.

UNIT-2 (15 HOURS)

Functions:

Inline function, function overloading

Introduction to Classes and Objects:

Classes in C++, class declaration, declaring objects, Defining Member functions, Inline member function, Array of objects, Objects as function argument, Static data member and member function, Friend function and friend class.

Constructors and Destructors:

Constructors, Instantiation of objects, Default constructor, Parameterized constructor, Constructor overloading, Copy constructor and its use, Destructors.

UNIT-3 (15 HOURS)

Operator Overloading:

Overloading unary and binary operators.

Inheritance:

Derived class and base class, access specifiers, type conversion, accessing the base class member, Types of Inheritance, Virtual base class, Abstract class

UNIT-4 (15 HOURS)

Virtual Functions and Polymorphism:

Virtual functions, pure virtual functions; Polymorphism. Compile time polymorphism, Run time polymorphism

File Handling:

Opening and Closing a file, File modes, Functions for I/O operations.

Introduction to standard template library.

Components of STL, Containers: Vector and lists.

Text Book:

i. E. Balagurusamy: *Object oriented programming with C++*

REFERENCE:

i. Bjarne Stroustrup: *The C++ programming language.*

ii. Robert Lafore: *Object oriented programming in C++*

iii. Yashwant Kanathker: *Let's C++*

iv. Schildt. H: *C++ The Complete Reference*

PRACTICALS (2 CREDITS: 30 HOURS)

UNIT -1

1. Write a C++ program using decision making and loop structure for the given situation.
2. Use the structure in C++ program for solving the given problem.
3. Create C++ programs to perform the given arithmetic operations using pointers.
4. Write a program to Demonstrate the use of Streams in C++
5. Write a menu driven program in C++ using switch statement.

UNIT -2

1. Develop programs that implements a class and use it with objects.
2. Develop relevant friend functions to solve the given problem.
3. Use function overloading to solve the given problem.
4. Write a program to implement all types of constructors (constructor overloading).
5. Write program to delete the given object using destructor in C++ program.

UNIT -3

1. Write a program to demonstrate operator overloading for Unary operator.
2. Write a program to demonstrate operator overloading for Binary operator.'
3. Write a program for implementing single, multi-level, and multiple inheritance.
4. Write a C++ program using virtual base class.
5. Develop a program to demonstrate use of Pointer to derived class

UNIT -4

1. Write a C++ program demonstrating the use of pure virtual function.
2. Implement run time polymorphism using virtual functions in the given C++ program.
3. Develop C++ program to perform read/write operation from/to the given file.
4. Write a C++ program to illustrate the iterators in vector
5. Develop C++ program to demonstrate the use of list containers.

BACHELORS WITH COMPUTER APPLICATIONS AS MAJOR
4th SEMESTER
CAP422J3 COMPUTER APPLICATIONS _ COMPUTING MATHEMATICS
CREDITS: THEORY (4) PRACTICAL (2)

COURSE LEARNING OUTCOMES:

- *To introduce elements of 10+2 level mathematics to students of Computer Applications who are from a medical or arts background*
- *To cover fundamental concepts of matrices and determinants*
- *To cover fundamental concepts of calculus.*
- *To acquire fundamental knowledge regarding the problems of approximation and errors in Computer based numerical problems solving.*

UNIT 1: MATRICES AND DETERMINANTS:

Matrix, Types of Matrices, Matrix Operations: Add, Sub, Multiply, Divide, Transpose. Determinants, Minors and Cofactors, Properties of Determinants. Elementary Transformations. Solutions of Linear Equations by Matrix Method and Determinants.

UNIT 2: FUNCTIONS AND LIMITS

Functions, Concept, Domain and Range, Types of Functions
Limits and Continuity: Limit of a Function, Algebra of Limits, Simplification and Evaluation of Limits. Standard Limits, Continuity, Geometric Meaning

UNIT 3: DIFFERENTIATION

Ab-Initio Method, Derivatives of some Common Functions, Rules to find Derivatives, Second Order Derivatives, Anti-Derivative of a Function.

UNIT 4: APPROXIMATION AND ERRORS:

Mathematical Modelling and Engineering Problem-Solving, Role of Computers and Software. Significant Figures, Accuracy and Precision. Numerical Error. Floating Point Representation. Computer Arithmetic. Round-Off, Truncation Errors and Error Propagation.

TEXTBOOK:

1. Wiley's Problems in Mathematics for JEE Main and Advanced (Vol 1 + 2)
2. Introduction to Linear algebra by Gilbert Strang, 6th Edition

REFERENCES:

1. Numerical Methods for engineers. S C Chapra and R P Canale , McGrow Hill International Edition.
2. Numerical Methods for Scientific & Engineering Computation, M. K. Jain, S.R.K.
3. Numerical Methods in Science & Engineering Prog. - By Dr. B. S. Grawal, Khanna Pub., New Delhi.
4. R S Aggarwal, Senior Secondary School Mathematics for Class XI/XII (Bharati Bhawan)
5. R.D Sharma – Mathematics for Class 11th / 12th (4 Vols.)
6. Computer Oriented Numerical Methods, R. S. Salaria., Khanna Publisher.

TUTORIALS:

1. Show how to use determinants to calculate the area of a triangle from its coordinates.
2. Show how to use determinants to determine that a set of given points are collinear.
3. Show how to use determinants to solve a non-homogenous system of linear equations
4. Show how to use determinants to solve a homogenous system of linear equations
5. Show that the value of a determinant remains unchanged if its rows and columns are interchanged
6. Show that if each element of a row or a column of a determinant is multiplied by a constant k then the value of the new determinant is k times the value of the original determinant
7. Show the different ways of representing a mathematical function. Also, given a mathematical function, show how to plot its graph.
8. Use limits to determine the tangent to the curve at a given point
9. Use limits to determine the instantaneous velocity of a ball dropped from a height at a given time.
10. Use differentiation to compute the rate of change of one quantity from the rate of change of another, more easily measurable quantity (e.g. the rate of increase of the radius of a balloon from the rate of change of its volume)
11. Show how to use the derivative of a function as a measure of rate of change
12. Show how to use the derivative to calculate the maxima and minima of a function
13. Write an algorithm for finding the roots of a quadratic equation
14. Develop a simple mathematical model to determine the terminal velocity of a free-falling body (a parachutist) near the earth's surface. Present a numerical solution for this problem.

REFERENCES:

- Stewart, Calculus Concepts and Contexts (2nd Edition)
- R.S. Aggarwal, Mathematics XI/XII
- Numerical Methods for engineers. S C Chapra and R P Canale

BACHELOR WITH APPLIED COMPUTING AS MINOR

(FOR STUDENTS WITH MAJOR IN COMPUTER APPLICATIONS / INFORMATION TECHNOLOGY)

4th SEMESTER

ACP422N APPLIED COMPUTING _ IOT FUNDAMENTALS

CREDITS: THEORY: 3; PRACTICAL: 1

COURSE LEARNING OUTCOMES:

- *Understand the fundamental characteristics of IOT, including its physical design, basic components, and the concepts of things, sensing, and actuators.*
- *Explore various application areas of IoT such as home automation, smart cities, medical, logistics, environment, analytics, and smart grids.*
- *Gain insights into IoT protocols used for communication and data exchange within IoT ecosystems.*
- *Develop hands-on skills in working with hardware platforms like Raspberry Pi and Arduino, and learn how to implement basic sensors for monitoring temperature, humidity, proximity, gas, air quality, and ultrasonic sensors.*

THEORY (3 CREDITS)

UNIT - 1 (15 Hours)

Introduction: Definition & Characteristics of IOT, Physical Design of IOT, Basic Components of IoT, Thing, Sensing & Actuators, Vision, Physical Parameters. IOT Protocols.
Application Areas of IoT: Home Automation, Smart Cities, Medical, Logistics, Environment, Analytics. Smart Grids.

UNIT - 2 (15 Hours)

IOT Communication Models, APIs, IoT Architecture: Basic Architecture: 3 layer and 5-layer Architecture, ITU-IoT Reference Model, Machine to Machine Communication, IoT Gateways, Wireless Sensor Networks. Technologies: Bluetooth Low Energy (BLE), ZigBee: Architecture, Comparison with other wireless standards. LoraWAN.

UNIT - 3 (15 Hours)

Electronic Product Code (EPC), Near Field Comm. (NFC), 6LoWPAN, End to End Reliability: MQTT, SCADA. Hardware and Software Platforms: Hardware: Raspberry Pi, ESP8266 Wifi Module, Arduino. Implementation of Basic Sensors (temperature, humidity, proximity, gas, air quality, Ultrasonic sensors)
Internet of Things Privacy and Security Issues, Steps towards a security platform in IoT

TEXT BOOKS:

1. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1stEdition, VPT, 2014. (ISBN-13: 978-8173719547)
2. Internet of Things (IoT), Dr. Kamlesh Lakhwani, Dr. Hemant Kumar Gianey, Joseph Kofi Wireko, Kamal Kant Hiran

REFERENCES:

1. Schwartz, Marco. “Internet of Things with Arduino Cookbook”. Packt Publishing Ltd, 2016.
2. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”, 1st Edition, Academic Press, 2014. (ISBN-13: 978-0124076846)
3. Hakima Chaouchi “The Internet of Things Connecting Objects to the Web” by Wiley publications

PRACTICAL (1 CREDIT: 30 HOURS)

LEARNING OUTCOME*: These practical's cover a range of fundamental concepts in IoT, including hardware interfacing, sensor integration, data communication, cloud integration, and real-world applications. Students can gain hands-on experience and a solid understanding of the core principles of IoT through these practical exercises.*

1. Study the fundamental of IOT software's and components. Install Arduino IDE development platform.
2. Connect a microcontroller chip on a breadboard, establish power connections, and verify basic functionality. Gather the necessary components: microcontroller chip (e.g., Arduino Uno), breadboard, jumper wires, USB cable for power, and an LED.
3. To practice connecting analog input and output components to the microcontroller using the breadboard.
4. Write a program to Read sensor data using analog or digital pins.
5. LED Blinking Using Arduino: To interface LED/Buzzer with Arduino/Raspberry Pi/ ESP8266 and write a program to turn ON & OFF.
6. To interface Push button/Digital sensor (IR/LDR) with Arduino/ ESP8266/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection. Upload the program to the board and observe the blinking.
7. To interface DHT11 sensor with Arduino/Raspberry Pi/ ESP8266 and write a program to print temperature and humidity readings and Display sensor readings on the serial monitor.
8. IoT Cloud Platform Integration: Create an account on a popular IoT cloud platform ThingSpeak.
9. Write a program to Send temperature and humidity sensor data to the cloud and visualize it on a dashboard.
10. Write a Program to interface motor using relay with Arduino/Raspberry Pi/ ESP8266 and write a program to turn ON motor when push button is pressed.
11. Build a web server using Arduino and Ethernet/Wi-Fi shield using ESP8266 Wifi module. Develop a web page to control LEDs remotely.
12. Use buttons on the webpage to toggle LED states.
13. Log data at regular intervals to an SD card or cloud platform. Create graphs to visualize the logged data over time.
14. Set up an MQTT broker (e.g., Mosquitto) on a computer or cloud server. Program an Arduino to publish sensor data and subscribe to commands via MQTT.
15. Create a soil moisture sensing system using Arduino and moisture sensors. Send data to the cloud to monitor soil conditions remotely.
16. Design a basic home automation system with Arduino and relays.

BACHELORS WITH COMPUTER APPLICATIONS AS MINOR
4th SEMESTER

CAP422N COMPUTER APPLICATIONS _ DBMS

CREDITS: THEORY (3) PRACTICAL (1)

Course Learning Outcomes:

- *To introduce the core concept of Relational Database.*
- *To enable students to design the databases for a wide variety of Real-World problems*
- *To introduce the concept and process of Database Normalization*
- *To enable the student to learn DML, DDL, DCL commands using SQL*

Unit I:

Introduction to Databases, Database Users, Characteristics of Database approach, Applications of DBMS, Advantages and Disadvantages, Database System Concepts and Architecture, Data Models, Schemas, Instances, Three-Schema Architecture and Data Independence, Database System, Centralized and Client/Server Architecture for DBMS,

Unit II:

Data Modelling using Entity-Relationship Mode, Entity Types, Entity Sets, Attributes, and Keys, Relationship Types, Relationship Sets, ER Diagrams, Relational Data Model and Relational Model Constraints, Functional Dependencies, Normalization: 1NF, 2NF, 3NF

Unit III:

Relational Algebra and Relational Calculus, Introduction to DDL, DML, DCL. Introduction to transactions, Transaction states ACID properties, Concurrency Control Techniques: 2-phase locking, time stamp ordering.

TEXTBOOK:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010

REFERENCES:

1. An introduction to database systems by Desai, Bipin C
2. PL/SQL by Ivan Bayross

DBMS- LAB

Creation of Tables in SQL (DDL)

1. Overview of using SQL tool, Data types in SQL,
2. Create Schema and insert at least 5 records for each table.
3. Queries for Altering Tables (with add and modify clause) , Dropping Tables
4. Queries for DROP RENAME TRUNCATE and Describe a Table
5. SQL Constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT

Practicing DML Commands

6. Insert, Select, Update, Delete
7. Select Command with WHERE, IN, BETWEEN AND LIKE clause
8. SELECT with AND, OR, NOT
9. Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING.

Joining Tables

10. Practice the Inner Join, Left Join,
11. Practice the Right Join, Cross Join

Create the following Database Schema and Practice the queries

EMP (eno, ename, bdate, title, salary, dno)
PROJECT (pno, pname, budget, dno)
DEPT (dno, dname, mgreno)
WORKSON (eno, pno, resp, hours)

12. Write an SQL query that returns the project number and name for projects with a budget greater than 100,000.
13. Write an SQL query that returns all works on records where hours worked is less than 10 and the responsibility is 'Manager
14. Write an SQL query that returns the employees (number and name only) who have a title of 'EE' or 'SA' and make more than 35,000.
15. Write an SQL query that returns the employees (name only) in department 'D1' ordered by decreasing salary.
16. Write an SQL query that returns the departments (all fields) ordered by ascending department name.
17. Write an SQL query that returns the employee's name, department name, and employee title.
18. Write an SQL query that returns the project name, hours worked, and project number for all works on records where hours > 10.
19. Write an SQL query that returns the project name, department name, and budget for all projects with a budget < 50,000.
20. Write an SQL query that returns the employee numbers and salaries of all employees in the 'Consulting' department ordered by descending salary.
21. Write an SQL query that returns the employee's name, project name, employee title, and hours for all works on records.

BACHELOR WITH COMPUTER APPLICATIONS AS MAJOR (CT-1)
6th SEMESTER

CAP622J1: COMPUTER APPLICATIONS _ PYTHON PROGRAMMING

CREDITS: THEORY (3) PRACTICAL (1)

LEARNING OUTCOMES:

After completing this course, the learner shall be able to:

1. *Understand the fundamentals of programming through Python*
2. *Transform a solution from a subjective world into an objective world.*
3. *Work with and manipulate different basic data structures available in Python*
4. *Use the List comprehensions and generators in their programs*
5. *Apply basic object-oriented concepts to design classes and objects in Python*
6. *Use concepts of Inheritance and Polymorphism in their programs*
7. *Perform basic file operations for text and CSV files*
8. *Use existing inbuilt Python modules*
9. *Develop custom modules*
10. *Use basic functionalities provided by packages Numpy and Pandas*
11. *Develop visualizations using different plotting functions available in matplotlib*

UNIT-I: (15 LECTURES)

Origins and History of Python, Structure of a Python Program, Interpreter shell, Indentation, Comments, Identifiers and keywords, Literals, Basic operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment Operator, Bit wise operator)

Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables. Creating Python Programs: Input and Output Statements, Control statements: -branching, looping, range function, Exit function, break, continue and pass.

UNIT: II (15 LECTURES)

Functions: Defining a Function, Calling a Function, def Statements with Parameters, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Return Values and return Statements, The None Value, Local and Global Scope, The global Statement.

The List Data Type: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods. List comprehension.

UNIT: III (15 LECTURES)

The Dictionary Data Type: Creating Dictionary, Accessing and Modifying key: value Pairs in Dictionaries, Built-In Functions used on Dictionaries, Dictionary Methods

Tuples and Sets, Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Set Methods, Traversing of Sets

Strings, Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Files: Types of Files, Creating and Reading Text Data

TEXT BOOKS

1. Gowrishankar S, Veena A, "Introduction to Python Programming", 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372
2. Downey, A.B., (2015), Think Python How to think like a Computer Scientist, 3rd edition.
3. Taneja, S. & Kumar, N., (2017), Python Programming- A Modular Approach. Pearson Education.

REFERENCE BOOKS

1. Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data", 1st Edition, O'Reilly Media, 2016. ISBN-13: 978-1491912058
2. Aurelien Geron, "Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", 2nd Edition, O'Reilly Media, 2019. ISBN – 13: 978-9352139057.
3. Wesley J Chun, "Core Python Applications Programming", 3rd Edition, Pearson Education India, 2015. ISBN-13: 978-9332555365

4. Miguel Grinberg, “Flask Web Development: Developing Web Applications with Python”, 2nd Edition, O'Reilly Media, 2018. ISBN-13: 978-1491991732.
5. Brown, M. C. (2001). The Complete Reference: Python, McGraw Hill Education.
6. Dromey, R. G. (2006), How to Solve it by Computer, Pearson Education.
7. Guttag, J. V. (2016), Introduction to computation and programming using Python. MIT Press.
8. Liang, Y.D. (2013), Introduction to programming using Python. Pearson Education.

LAB WORK: PYTHON PROGRAMMING

1. Install and configure Python IDE
2. Write simple Python program to display message on screen
3. Write a program to demonstrate different number data types & arithmetic operations in Python.
4. Write a Python program to demonstrate use of conditional statements:
5. Write Python program to demonstrate use of looping statements
6. Demonstrate the following control transfer statements in Python with suitable examples.
   i) break
   ii) continue
   iii) pass
7. Write a python program to demonstrate the working of following functions in Python.
   i) id( )
   ii) type( )
   iii) range( )
8. Write Python programs to demonstrate the following:
   i) input( )
   ii) print( )
   iii) ‘sep’ attribute
   iv) ‘end’ attribute
9. Write a program to create a menu driven program with the following options using functions:
   Addition 2. Subtraction 3. Multiplication 4. Division
10. Write Python program to perform following operations on Lists:
    a) Create list
    b) Access list
    c) Update list (Add item, Remove item)
    d) Delete list
11. Write Python program to perform following operations on Tuples:
    a) Create Tuple
    b) Access Tuple
    c) Update Tuple
    d) Delete Tuple
12. Write Python program to perform following operations on Sets:
    a) Create Set
    b) Access Set elements
    c) Update Set
    d) Delete Set
13. Write Python program to perform following operations on Dictionaries:
    a) Create Dictionary
    b) Access Dictionary elements
    c) Update Dictionary
    d) Delete Set
    e) Looping through Dictionary
14. Write a Python program to demonstrate slicing operations on lists and strings.
15. Write a Python program to perform read and write operations on a file.

BACHELOR WITH COMPUTER APPLICATIONS AS MAJOR (CT-2)

6th SEMESTER

CAP622J2: COMPUTER APPLICATIONS _ COMPUTER ORGANIZATION & ARCHITECTURE

CREDITS: THEORY (4) PRACTICAL (2)

COURSE LEARNING OUTCOMES:

After the course the students are expected to be able to:

- *Explain design of the various functional units and components of computers.*
- *Explain the basics of organizational and architectural issues of a digital computer and Classify and compute the performance of machines.*
- *Explain principles of computer organization and the basic architectural concepts.*
- *Demonstrate memory management and I/O techniques.*
- *To have better idea on how to write assemble language programs.*
- *Explain key aspects of Computer Organization & Architecture by enabling them to perform the experiments with support of a design and simulation.*

UNIT 1: BASIC COMPUTER ORGANISATION (15 HOURS)

Digital Computers, Von Neumann Computer, Basic organisation of a Computer, Computer Generations, Instruction Codes, Stored Program Organization, Computer registers, Computer instructions, Instruction cycle, Fetch and Decode, Common Bus System, Bus Interconnection Structures

UNIT 2: ARITHMETIC LOGIC UNIT (15 HOURS)

General Register Organisation, Control Word, Stack Organisation, Instruction Formats, Addressing Modes (Direct, Indirect, Register, Indexed), Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), Instruction Pipelining, Control Unit Implementation: Hardwired and Microprogrammed, Control Memory.

UNIT 3: MEMORY ORGANIZATION (15 HOURS)

Memory Hierarchy, Memory Technology, Main memory, RAM and ROM Chips, Memory Address Map, Memory Connection to CPU. RAM Technologies, SRAM, DRAM, Read Only Memory(ROM), PROM, EEPROM, Flash Memory, Auxiliary memory, Magnetic Disks, Magnetic Tape, Cache memory, Cache Memory Mappings (Direct, Associative, Set-Associative)

UNIT 4: INPUT-OUTPUT ORGANIZATION (15 HOURS)

Peripheral Devices, Input-Output Interface, I/O Bus and Interface Modules, Isolated and Memory Mapped I/O, Modes of Transfer, Programmed I/O, Interrupt-Initiated I/O, Direct Memory Access (DMA), Bus Arbitration, DMA Controller, DMA Transfer, Input-Output Processor (IOP). Introduction to parallel Processing, Multi-processing systems. Multicore architecture.

TEXT BOOK:

1. M.Morris Mano-Computer System Architecture, Revised Third Edition, Pearson Education

REFERENCE BOOK

1. William Stallings-Computer Organization and Architecture, Tenth Edition, Pearson Education
2. Carl Hamacher-Computer Organization, Fifth Edition, Tata McGraw Hill
3. Ramesh S Gaonkar-Microprocessor Architecture, Programming, and Applications with the 8085, Prentice Hall

LIST OF PRACTICALS:

Computer Organization and architecture lab consist of performing various experiments in GNU Sim (An open source and platform independent simulator for 8085 microprocessor).

1. Write the working of 8085 simulator GNUsim8085.
2. Write the working of 8086 simulator EMU8086.
3. Discuss basic architecture of 8085
4. Discuss basic architecture of 8086
5. Discuss the various registers available in 8085.
6. Study the complete instruction set of 8085 and write the instructions in the instruction set of 8085 along with examples.
7. Write an assembly language code in GNUsim8085 to demonstrate immediate addressing.
8. Write an assembly language code in GNUsim8085 to demonstrate indirect addressing.
9. Write an assembly language code in GNUsim8085 to demonstrate indexed addressing.
10. Write an assembly language code in GNUsim8085 to implement data transfer instruction.

11. Write an assembly language code in GNUsim8085 to store numbers in reverse order in memory location.
12. Write an assembly language code in GNUsim8085 to implement arithmetic instruction.
13. Write an assembly language code in GNUsim8085 to perform addition of two 8 bit numbers.
14. Write an assembly language code in GNUsim8085 to perform addition of two 16 bit numbers.
15. Write an assembly language code in GNUsim8085 to find 1's & 2's complement of a 8 bit number.
16. Write an assembly language code in GNUsim8085 to add two 8 bit numbers stored in memory and also storing the carry.
17. Write an assembly language code in GNUsim8085 to find the factorial of a number.
18. Write an assembly language code in GNUsim8085 to implement logical instructions.
19. Write an assembly language code in GNUsim8085 to implement stack and branch instructions.
20. Write a program using in GNUsim8085 for Decimal addition and subtraction of two Numbers.
21. Write a program using in GNUsim8085 for Hexadecimal addition and subtraction of two Numbers.
22. Write a program using 8085 Microprocessor for addition and subtraction of two BCD numbers.
23. Write a program to perform multiplication of two 8 bit numbers using 8085.
24. Write a program to perform division of two 8 bit numbers using 8085.
25. Write a program to find the largest and smallest number in an array of data using 8085 instruction set.
26. Write a program to arrange an array of data in ascending and descending order.
27. Write an assembly language code in EMU8086 to store numbers in reverse order in memory location.
28. Write an assembly language code in EMU8086 to implement arithmetic instruction.
29. Write an assembly language code in EMU8086 to perform addition of two 8 bit numbers.
30. Write an assembly language code in EMU8086 to perform addition of two 16 bit numbers.

BACHELOR WITH COMPUTER APPLICATIONS AS MAJOR (CT-3)
6th SEMESTER
CAP622J3: COMPUTER APPLICATIONS _ PROBABILITY AND STATISTICS
CREDITS: THEORY (4) PRACTICAL (2)

COURSE LEARNING OUTCOMES:

1. *Learn the language and core concepts of probability theory.*
2. *Understand basic principles of statistical inference.*
3. *Become an informed consumer of statistical information and have a good knowledge of what expectation and variance mean and be able to compute them.*

PREREQUISITE: Fundamental Mathematics

UNIT 1: (15 LECTURES)

Introduction to Probability: Random experiment, sample space, trial, event. Simple probability, Compound Probability, mutually exclusive events, Addition theorem, independent events, Multiplication theorem, Dependent events, Conditional probability, Bayes' Theorem, Partitions and Total probability law. Exploring Univariate Data: Types of data, Mean, Mode and Median.

UNIT 2: (15 LECTURES)

Standard Deviation and Variance, Range and Finding Outliers. Counting, Random variables, probability mass function, probability density function. distributions, quantiles, mean-variance, Joint distributions, covariance, correlation, independence, and Central limit theorem.

Discrete Distributions, Random Variables, Binomial Distributions, Geometric Distributions Continuous Distributions, Density Curves, The Normal Distribution

UNIT 3: (15 LECTURES)

Multivariate Data, Scatter Plots, Correlation, The Least Squares Regression Line, Residuals, Non-Linear Models, Relations in Categorical Data, Margins of Error and Estimates, Confidence Interval for a Proportion, Confidence Interval for the Difference of Two Proportions, Confidence Interval for a Mean, Confidence Interval for the Difference of Two Means.

UNIT 4: (15 LECTURES)

Tests of Significance, Inference for the Mean of a Population, Sample Proportions, Inference for a Population Proportion, Comparing Two Means, Comparing Two Proportions, Goodness of Fit Test, and Two-way Tables. Simple correlation (Pearson's correlation coefficient), Simple linear regression, Prediction, error in prediction, principle of least square.

TUTORIALS:

1. Two dice are rolled. Find the sample space for this experiment.
2. What is the probability of drawing a red card from a standard deck of 52 cards?
3. If two events are mutually exclusive, and the probability of either event A or event B occurring is 0.4, what is P(A) + P(B)?
4. If two events are independent, and the probability of event A is 0.3 and the probability of event B is 0.6, what is P(A and B)?
5. If A and B are mutually exclusive events, and P(A) = 0.2, what is P(B)?
6. Calculate the conditional probability of event A given that event B has occurred if P(A) = 0.4 and P(B) = 0.3.
7. Using Bayes' Theorem, find the probability of event A if P(B/A) = 0.6, P(A) = 0.4, and P(B)= 0.5.
8. Define a random variable and provide an example of one.
9. If X is a discrete random variable with the following probability mass function: P(X = 1) = 0.2, P(X = 2) = 0.4, and P(X = 3) = 0.4, find E(X), the expected value of X.
10. In a deck of cards, what is the probability of drawing a face card (jack, queen, or king)?
11. If you roll a fair six-sided die three times, what is the probability of getting three 6s in a row?
12. If there are 25 students in a class, and 5 of them wear glasses, what is the probability that a randomly selected student does not wear glasses?
13. If a bag contains 8 white balls and 6 black balls, what is the probability of drawing a white ball and then a black ball (without replacement)?
14. A box contains 12 chocolates, of which 3 are dark chocolate. What is the probability of randomly selecting a dark chocolate?
15. If you have a deck of cards and draw 2 cards with replacement, what is the probability that both cards are aces?

16. If you flip a coin three times, what is the probability of getting exactly two heads?
17. If a jar contains 30 red marbles and 20 blue marbles, what is the probability of drawing a red marble on the first try and a blue marble on the second try (without replacement)?
18. If you roll a fair six-sided die four times, what is the probability of getting at least one 1?
19. Imagine you are flipping a fair coin. How can you use probability distributions to represent the outcomes of this coin toss experiment?
20. Suppose you have a standard six-sided die. What is the probability mass function for this die, and what is the probability of rolling a 3?
21. You are conducting a survey where each respondent can either say "Yes" or "No" to a question. If you expect 70% of people to say "Yes," can you use the binomial distribution to calculate the probability of getting exactly 4 "Yes" responses out of 10 respondents?
22. You are modeling the outcome of a single basketball free throw attempt, where a player either makes it (success) or misses it (failure). Is this scenario best represented by a Bernoulli distribution? Why or why not?
23. Imagine you are tracking the number of customer arrivals at a small coffee shop per hour. Can you explain when and why you might use a Poisson distribution to model this situation?
24. Consider the heights of adult males in a population. Why is the normal distribution often used to describe this data, and what are the defining characteristics of the normal distribution?
25. If a student's test score has a z-score of -1.5, what does this mean in terms of their performance compared to the rest of the students? How can you use percentiles to describe their ranking within the class?
26. Suppose you have a population of 100 test scores with a mean of 75 and a standard deviation of
10. If you take random samples of 30 test scores each and calculate the sample means, what is the expected mean of the sample means, and what is the standard error of the sample means? You're conducting a survey to estimate the average income of a population.
27. From a sample of 50 individuals, you find a sample mean income of 800000 and a sample standard deviation of 70000. Calculate a 95% confidence interval for the population mean income.
28. In a chi-square goodness-of-fit test, you expect a uniform distribution of colors in a bag of marbles, but you observe the following counts: Red (25), Blue (30), Green (20), and Yellow (25). Calculate the chi-square test statistic and determine if the observed distribution significantly differs from the expected uniform distribution at a 5% significance level
29. Suppose you have a dataset of 50 test scores, and the scores are normally distributed with an unknown mean (μ) and a known standard deviation (σ) of 15. If the maximum likelihood estimation gives you a mean of 65, what is the likelihood function for this dataset?
30. In a survey of 200 people, 120 said they prefer tea over coffee. Calculate a 95% confidence interval for the proportion of people in the entire population who prefer tea.

TEXTBOOK:

1. Probability and Statistics in Engineering (4th Edition) - W. Hines, D. Montgomery, D. Goldsman, C. Borror- Wiley Publication.

2. Introduction to Probability and Statistics for Engineers and Scientists (3rd Edition) - Sheldon M. Ross, Elsevier Academic Press.

REFERENCES:

1. Mood A.M. Graybill F.A and Boes D.C. (1974): Introduction to the Theory of Statistics. McGraw Hill.
2. Snedecor G.W and Cochran W.G. (1967); Statistical Methods. Lowa State University Press.
3. Cooke, Cramer and Clarke (1996): Basis Statistical Computing, Chapman and Hall. 4. David S. (1996): Elementary Probability, Oxford House.
4. Meyer P.L (1970): Introductory Probability and Statistical application, Addison Wesley.

BACHELOR WITH APPLIED COMPUTING AS MINOR
SIXTH SEMESTER
(FOR STUDENTS WITH MAJOR IN COMPUTER APPLICATIONS / INFORMATION TECHNOLOGY)

ACP622N: APPLIED COMPUTING _ ARTIFICIAL INTELLIGENCE

CREDITS: THEORY (3) PRACTICAL (1)

MAX. MARKS: 100 MIN. MARKS: 36

COURSE LEARNING OUTCOMES:

1. *Study the concepts of Artificial Intelligence.*
2. *Learn the methods of solving problems using Artificial Intelligence.*
3. *Learn the knowledge representation techniques, reasoning techniques and planning.*

THEORY (3 CREDITS)

PREREQUISITE: Data Structures and Analysis of Algorithm

UNIT 1: AI AND TURING TEST: (15 LECTURES)

Introduction to Artificial Intelligence, Background and Applications, Turing Test, Rational Agent approaches to AI, Introduction to Intelligent Agents, structure, behaviour and environment.

UNIT 2: SEARCHING PROBLEM SPACE: (15 LECTURES)

Problem Characteristics, Production Systems, Informed vs Uninformed Strategies, Hill Climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm.
The Wumpus World Environment, Acting and reasoning in the Wumpus world, Representation, Reasoning, Logic, Prepositional Logic and First-Order Logic, Rules of Inferencing: Resolution Principle.

UNIT 3: UNCERTAIN KNOWLEDGE AND REASONING: (15 LECTURES)

Acting under Uncertainty, handling uncertain knowledge, Uncertainty and rational decisions, Basic Probability Notation, Prior probability, Conditional probability, joint probability distribution, Bayes' Rule and Its Use, Applying Bayes' rule: The simple case, Normalization, Dampster's Shafer's Theorem.

TUTORIAL (1 CREDIT):

1. Given a real life situation
   - identify the characteristics of the environment
   - identify the percepts available to the agent
   - identify the actions that the agent can execute
   - suggest the performance measures to evaluate the agent
   - recommend the architecture of the desired intelligent agent

2. Undertake a comparative study on popular programming languages used in AI.
3. Translate commonly used English sentences into their equivalent logical expressions.
4. Given a problem description, formulate it in terms of a state space search problem. Analyze the given problem and identify the most suitable search strategy for the problem. Make an analysis of the properties of proposed algorithms in terms of
   - time complexity
   - space complexity
   - optimality

5. Write a Program to Implement Breadth First Search.
6. Write a Program to Implement Depth First Search.
7. Write a program to implement Hill Climbing Algorithm
8. Write a program to implement A* Algorithm.
9. Write a program to implement Tic-Tac-Toe game.
10. Write a Program to Implement 8-Puzzle problem.
11. Write a Program to Implement Water-Jug problem.
12. Write a Program to Implement Alpha-Beta Pruning using Python.
13. Write a Program to Implement 8-Queens Problem.
14. Build a Fuzzy inference system for the Tipping Problem.
Given two sets of numbers between 0 and 5 (where 0 is for very poor, and 5 for excellent) that respectively represent quality of service and quality of food at restaurant, what should tip be?
15. Solve 2-input 1-output project risk prediction problem using Mamdani Inference. Make necessary assumptions.

TEXT BOOK:

1. DAN.W. Patterson, Introduction to A.I and Expert Systems - PHI, 2007.

REFERENCES:

1. Russell & Norvig. Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2nd edition. 2005.
2. Ivan Bratko,"Prolog Programming for Artificial Intelligence", 3 rd Edition, Pearson Education.
3. Rich & Knight, Artificial Intelligence - Tata McGraw Hill. 2nd edition, 1991.
4. W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House, 3rd edition, 2001.

BACHELOR WITH COMPUTER APPLICATIONS AS MINOR (CT-1)
6th SEMESTER
CAP622N: COMPUTER APPLICATIONS _ PYTHON PROGRAMMING
CREDITS: THEORY (3) PRACTICAL (1)

LEARNING OUTCOMES:
After completing this course, the learner shall be able to:

1. *Understand the fundamentals of programming through Python*
2. *Transform a solution from a subjective world into an objective world.*
3. *Work with and manipulate different basic data structures available in Python*
4. *Use the List comprehensions and generators in their programs*
5. *Apply basic object-oriented concepts to design classes and objects in Python*
6. *Use concepts of Inheritance and Polymorphism in their programs*
7. *Perform basic file operations for text and CSV files*
8. *Use existing inbuilt Python modules*
9. *Develop custom modules*
10. *Use basic functionalities provided by packages Numpy and Pandas*
11. *Develop visualizations using different plotting functions available in matplotlib*

UNIT-I: (15 LECTURES)
Origins and History of Python, Structure of a Python Program, Interpreter shell, Indentation, Comments, Identifiers and keywords, Literals, Basic operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment Operator, Bit wise operator)
Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables. Creating Python Programs: Input and Output Statements, Control statements: -branching, looping, range function, Exit function, break, continue and pass.

UNIT: II (15 LECTURES)
Functions: Defining a Function, Calling a Function, def Statements with Parameters, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Return Values and return Statements, The None Value, Local and Global Scope, The global Statement.
The List Data Type: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods. List comprehension.

UNIT: III (15 LECTURES)
The Dictionary Data Type: Creating Dictionary, Accessing and Modifying key: value Pairs in Dictionaries, Built-In Functions used on Dictionaries, Dictionary Methods
Tuples and Sets, Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Set Methods, Traversing of Sets
Strings, Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.
Files: Types of Files, Creating and Reading Text Data

TEXT BOOKS

1. Gowrishankar S, Veena A, "Introduction to Python Programming", 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372
2. Downey, A.B., (2015), Think Python How to think like a Computer Scientist, 3rd edition.
3. Taneja, S. & Kumar, N., (2017), Python Programming- A Modular Approach. Pearson Education.

REFERENCE BOOKS

1. Jake VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data", 1st Edition, O'Reilly Media, 2016. ISBN-13: 978-1491912058
2. Aurelien Geron, "Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", 2nd Edition, O'Reilly Media, 2019. ISBN – 13: 978-9352139057.
3. Wesley J Chun, "Core Python Applications Programming", 3rd Edition, Pearson Education India, 2015. ISBN-13: 978-9332555365

4. Miguel Grinberg, “Flask Web Development: Developing Web Applications with Python”, 2nd Edition, O'Reilly Media, 2018. ISBN-13: 978-1491991732.
5. Brown, M. C. (2001). The Complete Reference: Python, McGraw Hill Education.
6. Dromey, R. G. (2006), How to Solve it by Computer, Pearson Education.
7. Guttag, J. V. (2016), Introduction to computation and programming using Python. MIT Press.
8. Liang, Y.D. (2013), Introduction to programming using Python. Pearson Education.

LAB WORK: PYTHON PROGRAMMING

1. Install and configure Python IDE
2. Write simple Python program to display message on screen
3. Write a program to demonstrate different number data types & arithmetic operations in Python.
4. Write a Python program to demonstrate use of conditional statements:
5. Write Python program to demonstrate use of looping statements
6. Demonstrate the following control transfer statements in Python with suitable examples.
   i) break
   ii) continue
   iii) pass
7. Write a python program to demonstrate the working of following functions in Python.
   i) id( )
   ii) type( )
   iii) range( )
8. Write Python programs to demonstrate the following:
   i) input( )
   ii) print( )
   iii) ‘sep’ attribute
   iv) ‘end’ attribute
9. Write a program to create a menu driven program with the following options using functions:
   Addition 2. Subtraction 3. Multiplication 4. Division
10. Write Python program to perform following operations on Lists:
    a) Create list
    b) Access list
    c) Update list (Add item, Remove item)
    d) Delete list
11. Write Python program to perform following operations on Tuples:
    a) Create Tuple
    b) Access Tuple
    c) Update Tuple
    d) Delete Tuple
12. Write Python program to perform following operations on Sets:
    a) Create Set
    b) Access Set elements
    c) Update Set
    d) Delete Set
13. Write Python program to perform following operations on Dictionaries:
    a) Create Dictionary
    b) Access Dictionary elements
    c) Update Dictionary
    d) Delete Set
    e) Looping through Dictionary
14. Write a Python program to demonstrate slicing operations on lists and strings.
15. Write a Python program to perform read and write operations on a file.

BACHELOR WITH COMPUTER APPLICATIONS AS MAJOR (CT-1)
8th SEMESTER (FOR HONOURS MODE ONLY)
CAP822J1 COMPUTER APPLICATIONS _ SOFTWARE ENGINEERING
CREDITS: THEORY (3) PRACTICAL (1)

UNIT I

Concept and Nature of Software, Software Crisis, Software Engineering – Concept, Goals and Challenges, Software Engineering Approach;

Software Development Process, Process Models - Waterfall Model, Evolutionary and Throwaway Prototyping Model, Incremental and Iterative Models, Spiral Model, Agile Process Model, Component based and Aspect Oriented development.

UNIT II

Introduction to Requirements Engineering - Requirements Types: functional and nonfunctional requirements. Requirement Engineering Framework. Requirement Elicitation Process and Techniques.

Software Cost Estimation: Software Process and Project Measurement: Measures, Metrics and Indicators, Size -Oriented Metrics vs. Function - Oriented Metrics, Capability Maturity Model Integration (CMMI). COCOMO Model

UNIT III

Basics of Design Engineering - Abstraction, Architecture, Patterns, Separation of concerns, Modularity, Functional Independence, refinement, Refactoring.

Function oriented design, Design principles, Coupling and Cohesion, Design Notations & Specifications, Structured Design Methodology.

Software Testing – Basics of Software Testing, Different types of testing techniques.

TEXTBOOKS:

1. Pankaj Jalote - An Integrated approach to Software Engineering, 3rd edition
2. Software Engineering: A Practitioner's Approach Seventh edition. Roger. S. Pressman

REFERENCE BOOKS:

1. Software Engineering: Principles and practice, 3rd Edition, Hans Van Vliet, Wiley.
2. Sommmerville - Software Engineering. Pearson, 7/e, 2006.
3. James F. Peters Software Engineering – An Engineering Approach, Wiley& Sons.

Software Engineering (Tutorials)

1. Study and compare different software process models and prepare comparison-based analysis report on it. Also identify the suitable applications for the individual process model.
2. Suppose that a project was estimated to be 400 KLOC. Calculate effort & time for each of 3 modes of development. Using the following basic model equations

$$E = a(KLOC)^b$$

$$time = c(Effort)^d$$

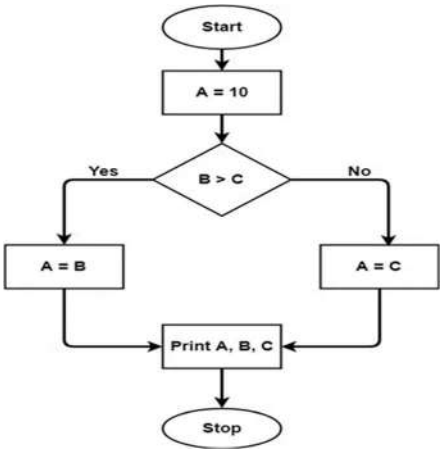
$$Personrequired = Effort/time$$

Software Product Type	**a**	**b**	**c**	**d**
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

3. Calculate the effort, duration and average persons required using the above details for basic CoCoMo model. Given
   Number of user inputs=15
   Number of user outputs=14
   Number of user inquiries=8
   Number of files =6
   Number of external interfaces=13
   The weighing factors have following values (inputs=3, outputs=5, inquiries=3, files=15, interfaces=7).

Assume that all complexity adjustment values are average and 1 function point = 25 LOC. The Values of Constant used in basic CoCoMo model. a=2.4, b=1.05, c=2.5, d=0.38.

4. Given the following values, compute function point when all complexity adjustment factor (CAF) and weighting factors are average.
   User Input = 50
   User Output = 40
   User Inquiries = 35
   User Files = 6
   External Interface = 4
5. For a Library Management System or University Admission system do the following.
   a. Identify the different requirements of the application.
   b. Identify the various elicitation techniques and their usage for the identified requirements.
   c. Classify the requirement into functional and non-functional requirements.
   d. Use various validation techniques to validate your requirement document.
6. Design the system using structured as per the instructions and applicability.
   a. Implement the design using DFD, ER diagrams and structure chart whichever applicable.
   b. Identify various processes, data store, input, output etc. of the system.
   c. Use processes at various levels to draw the DFDs.
   d. Identify various modules, input, output etc. of the system and ask students to analyze.
7. Identify the appropriate and necessary testing approach and design detailed test cases according to IEEE format for test cases.
   a. Identify various modules of the system so that they can be tested stand-alone.
   b. Identify the groups of the module that can be tested together in integration.
8. **Case Study:** You are the technical lead in your company for a new product aimed at university research scientists: an electronic lab notebook (ELN). The marketing team has given you the following information. The scientists who will use the ELN traditionally record their experiments in paper notebooks, but increasingly they are looking for digital solutions which allow them to share their information with other scientists they are working with, such as Wikis and Dropbox, which make frequent functionality changes. A small amount of market research indicates that the scientists vary considerably in what the capabilities they would like in an ELN, and the ELN solution will need to change over time as the scientists' needs change. However, the scientists all say an ELN should be easy to use and quick to use. The scientists work in labs headed by a senior academic (the Lab Head). Provision of computing resources for the lab (hardware and software) varies between labs. Some have all their needs met by the central university IT department; others manage all their own hardware and software and depend on central IT only for network connections; most are somewhere in between.
   a. Which software development process is likely to be most appropriate for developing the software system?
   b. Explain why, contrasting your choice with one other software development process.
9. Discuss in detail various testing techniques.
10. Calculate the cyclometic complexity of the following control flow graph.



BACHELOR WITH COMPUTER APPLICATIONS AS MAJOR (CT-III)
8th SEMESTER

CAP822J3: COMPUTER APPLICATIONS _ MOBILE APPLICATION DEVELOPMENT

CREDITS: THEORY (4) PRACTICAL (2)

LEARNING OUTCOMES

At the end of the course, students will:

- *Identify various concepts of mobile programming that make it unique from programming other platforms.*
- *Interpret features of Android Operating System.*
- *Critique mobile applications on their design pros and cons.*
- *Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces.*
- *Program mobile applications for the Android operating system that use basic and advanced phone features*

Unit I

Mobile Application Development, Mobile Applications and Device Platforms, Alternatives for Building Mobile Apps, Comparing Native vs. Hybrid Applications, The Mobile Application Development Lifecycle, The Mobile Application Front-End and Back-End, Key Mobile Application Services, Introduction to Java, Java Setup and Program structure, Inheritance and Exception, Handling Events, Debugging, Introduction to Classes.

Unit II

Introduction to Android: The Android Platform, Android SDK, Eclipse Installation, Android Installation, Building you First Android application, Understanding Anatomy of Android Application, Need of Android, Advanced Android Features, Tools and Software required for developing an Android application.

UNIT III

Android terminologies, Application Context, Activities, Services, Intents, Android Storing and Retrieving data, Receiving and Broadcasting Intents, Content Provider, Android Manifest File and its common settings, Using Intent Filter, Permissions, Android Networking and Web.

UNIT IV

Android User Interface Design Essentials: Fundamental UI design, User Interface Screen elements, Designing User Interfaces with Layouts, Text View, List View, Grid View, Image View, Scroll View, Drawing and Working with Animation, SQLite Database, Creating and Connection of the database.

TEXT BOOKS:

- Lauren Darcey and Shane Conder, “Android Wireless Application Development”, Pearson Education, 2nd ed. (2011)
- Jerome DiMarzio, “Beginning Android Programming with Android Studio”, 4th Edition.

REFERENCE BOOKS:

- Reto Meier, “Professional Android 2 Application Development”, Wiley India Pvt Ltd
- Mark L Murphy, “Beginning Android”, Wiley India Pvt Ltd
- Hortan, John, “Android Programming for Beginners”, Packet Publication, 2015, ISBN: 978-1-78588-326-2.

TUTORIAL-I

1. You are developing a Java application, and during the execution, an exception occurs. Explain how you would handle this exception using Java's exception handling mechanisms.
2. You are tasked with implementing a program that stores the temperatures of a week in an array. How would you use loops to find the average temperature and identify days with temperatures above a certain threshold?
3. Suppose you are designing a class to represent a car. Walk through the process of creating an object of this class, initializing its attributes, and demonstrating the use of constructors.
4. In a software system, there is a base class called "Shape" with subclasses "Circle" and "Rectangle." Explain a scenario where you would use inheritance to model different shapes and how you would apply method overriding for specific behaviors.
5. Imagine you are developing a simple banking application. Explain how you could use polymorphism to handle different types of bank accounts (e.g., savings account, checking account) through a common interface.

6. Design a scenario where encapsulation is crucial in maintaining a student database. Explain how encapsulation would help in ensuring data integrity and security.
7. In a simple game development project, there is a base class "Character" with a method "attack." Describe a situation where you would create subclasses for different types of characters (e.g., warrior, wizard) and override the "attack" method to provide unique behaviors for each character type.
8. You are a beginner in Android development, and you need to set up your development environment. Explain the steps you would take to install Eclipse, Android SDK, and configure your system for Android development.
9. Imagine you want to create a simple Android application that displays a welcome message when launched. Describe the steps you would follow, from creating a new project to running the application on an emulator or a physical device.
10. What are the key components that make up the anatomy of an Android application, and how do they interact with each other?
11. Describe how the Intent mechanism is used for communication between different components of an Android application. Provide an example scenario where Intents would be useful.
12. Why are permissions necessary in Android applications, and how are they declared in the Android Manifest file? Provide an example of a situation where a specific permission might be required.
13. You are developing a music player app, and you want to implement a feature that sends a broadcast when a new song starts playing. Describe how you would use Intents to broadcast this information, and how other components in your app or external apps might receive and respond to this broadcast.
14. Explain the concept of Intent Filters in Android and how they are used to handle implicit Intents. Provide an example scenario where an Intent Filter would be applied.
15. How do you design user interfaces in Android using layouts? Provide examples of different layout types and their use cases in creating visually appealing and responsive UIs.
16. Compare various Operating Systems with Android OS.
17. Design a basic Android application layout. Add Button and other UI elements and implement error handling for user interactions.
18. Explore and integrate advanced features of Android. Integrate advanced components, Implement functionality using features like background services or notifications.
19. Create and test Android Applications containing Activities, Services and Intents.
20. Explore the lifecycle of different Android Components. Implement inter-component communication using intents.
21. Implement data storage mechanisms in Android Applications. Use SharedPreferences for storing and retrieving simple data.
22. Implement SQLite database operations for structured data storage. Test data retrieval and manipulation using different storage options.
23. Create a content provider to share data between applications. Implement CRUD (Create, Read, Update, and Delete) operations using the content provider. Test the application’s behavior with data sharing between different components.
24. Design UI screen using different layout types (e.g., LinearLayout, RelativeLayout). Add UI elements such as TextView, ListView and ImageView. Test the UI layout on various screen sizes and orientations.
25. Create SQLite database schema with multiple tables. Implement CRUD operations (Create, Read, Update, delete) on the database. Test database operations by inserting, retrieving, updating and deleting records.
26. Add animation effects and scrollable views to Android applications. Implement animations using Android Animation API.
27. Create scrollable views using ScrollView or RecycleView. Test scrollable views in the application.
28. Create sample application with Login module where you will have to validate username and password till the username and password is not validated, Login button should remain disabled.
29. Create a budget planner application to help users track their expenses and manage their finances.
30. Design a fitness tracker application to record and monitor users exercise routines and progress.

BACHELOR WITH APPLIED COMPUTING AS MINOR
8th SEMESTER

ACP822N: APPLIED COMPUTING _ CLOUD COMPUTING

CREDITS: THEORY (3) PRACTICAL (1)

UNIT I

Overview of Cloud Computing: Definition and characteristics of cloud computing, Evolution and history of cloud computing. Types of cloud services: IaaS, PaaS, SaaS. Benefits and challenges of cloud computing. Cloud Deployment Models: Public cloud, Private cloud, Hybrid cloud, Community cloud, Considerations for choosing a deployment model. overview of major cloud service providers (AWS, Azure, Google Cloud, etc.) Business Agility: Benefits and challenges to Cloud architecture. Application availability, performance, security and disaster recovery; next generation Cloud Applications

UNIT II

Virtualization Basics: Introduction to virtualization, Types of virtualization: server, storage, network, Benefits of virtualization in cloud computing. Cloud Infrastructure, Hardware components in cloud data centers, Data center architecture: servers, storage, networking, Scalability and elasticity in cloud infrastructure, Virtual Machines and Containers, Virtual machines (VMs) vs. containers.

UNIT III

Cloud Applications: Technologies and the processes required when deploying web services; Deploying a web service from inside and outside a cloud architecture, advantages and disadvantages, Management of Cloud Services: Reliability, availability and security of services deployed from the cloud. Performance and scalability of services, tools and technologies used to manage cloud services deployment.

REFERENCES

1. Gautam Shroff, “Enterprise Cloud Computing Technology Architecture Applications”, Cambridge University Press; 1 edition, [ISBN: 9780521137355], 2010.
2. Cloud Computing: From Beginning to End by Ray J. Rafaels
3. Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing, A Practical Approach” McGraw-Hill Osborne Media; 1 edition [ISBN: 0071626948], 2009.
4. Dimitris N. Chorafas, “Cloud Computing Strategies” CRC Press; 1 edition [ISBN: 1439834539], 2010.

TUTORIAL (1 CREDIT)

S. No.	Example
1.	Guide students through signing up for an AWS Free Tier account to explore IaaS, PaaS, and SaaS offerings.
2.	Walkthrough on launching a virtual machine instance on Google Cloud Platform to understand IaaS.
3.	Show how to deploy a simple web application on Heroku to demonstrate PaaS capabilities.
4.	Create a comparison table highlighting the differences between public and private cloud services
5.	Discuss hypothetical scenarios where a company might benefit from a hybrid cloud approach.
6.	Assign a task to research and present on major cloud providers (AWS, Azure, Google Cloud) including their services and use cases.
7.	Instruct students to download and install the OpenStack DevStack on their local machines for a simplified OpenStack environment.
8.	Hands-on activity to install and set up VirtualBox to understand the basics of virtualization.
9.	Step-by-step guide on creating a virtual machine (VM) in VirtualBox, explaining VM concepts.
10.	Tutorial on installing Docker and running a simple container to explain containerization.
11.	Create a VM and a container, then compare resource usage and speed.
12.	Simulate load testing on a cloud-hosted application to observe auto-scaling in action.
13.	Walkthrough on launching different types of AWS EC2 instances to understand elasticity.
14.	Guide on deploying a basic web service (e.g., a "Hello World" application) on AWS.
15.	Instruct students to sign up for a free Heroku account if they haven't already. Guide them through the account setup process, which involves verifying their email and creating a new app on Heroku.

BACHELOR WITH COMPUTER APPLICATIONS AS MINOR (CT-1)
8th SEMESTER (FOR HONOURS MODE ONLY)
CAP822N COMPUTER APPLICATIONS _ SOFTWARE ENGINEERING
CREDITS: THEORY (3) PRACTICAL (1)

UNIT I

Concept and Nature of Software, Software Crisis, Software Engineering – Concept, Goals and Challenges, Software Engineering Approach;

Software Development Process, Process Models - Waterfall Model, Evolutionary and Throwaway Prototyping Model, Incremental and Iterative Models, Spiral Model, Agile Process Model, Component based and Aspect Oriented development.

UNIT II

Introduction to Requirements Engineering - Requirements Types: functional and nonfunctional requirements. Requirement Engineering Framework. Requirement Elicitation Process and Techniques.

Software Cost Estimation: Software Process and Project Measurement: Measures, Metrics and Indicators, Size -Oriented Metrics vs. Function - Oriented Metrics, Capability Maturity Model Integration (CMMI). COCOMO Model

UNIT III

Basics of Design Engineering - Abstraction, Architecture, Patterns, Separation of concerns, Modularity, Functional Independence, refinement, Refactoring.

Function oriented design, Design principles, Coupling and Cohesion, Design Notations & Specifications, Structured Design Methodology.

Software Testing – Basics of Software Testing, Different types of testing techniques.

TEXTBOOKS:

1. Pankaj Jalote - An Integrated approach to Software Engineering, 3rd edition
2. Software Engineering: A Practitioner's Approach Seventh edition. Roger. S. Pressman

REFERENCE BOOKS:

1. Software Engineering: Principles and practice, 3rd Edition, Hans Van Vliet, Wiley.
2. Sommmerville - Software Engineering. Pearson, 7/e, 2006.
3. James F. Peters Software Engineering – An Engineering Approach, Wiley& Sons.

Software Engineering (Tutorials)

1. Study and compare different software process models and prepare comparison-based analysis report on it. Also identify the suitable applications for the individual process model.
2. Suppose that a project was estimated to be 400 KLOC. Calculate effort & time for each of 3 modes of development. Using the following basic model equations

$$E = a(KLOC)^b$$

$$time = c(Effort)^d$$

$$Personrequired = Effort/time$$

Software Product Type	**a**	**b**	**c**	**d**
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

3. Calculate the effort, duration and average persons required using the above details for basic CoCoMo model. Given
   Number of user inputs=15
   Number of user outputs=14
   Number of user inquiries=8
   Number of files =6
   Number of external interfaces=13
   The weighing factors have following values (inputs=3, outputs=5, inquiries=3, files=15, interfaces=7).

Assume that all complexity adjustment values are average and 1 function point = 25 LOC. The Values of Constant used in basic CoCoMo model. a=2.4, b=1.05, c=2.5, d=0.38.

4. Given the following values, compute function point when all complexity adjustment factor (CAF) and weighting factors are average.
   User Input = 50
   User Output = 40
   User Inquiries = 35
   User Files = 6
   External Interface = 4
5. For a Library Management System or University Admission system do the following.
   a. Identify the different requirements of the application.
   b. Identify the various elicitation techniques and their usage for the identified requirements.
   c. Classify the requirement into functional and non-functional requirements.
   d. Use various validation techniques to validate your requirement document.
6. Design the system using structured as per the instructions and applicability.
   a. Implement the design using DFD, ER diagrams and structure chart whichever applicable.
   b. Identify various processes, data store, input, output etc. of the system.
   c. Use processes at various levels to draw the DFDs.
   d. Identify various modules, input, output etc. of the system and ask students to analyze.
7. Identify the appropriate and necessary testing approach and design detailed test cases according to IEEE format for test cases.
   a. Identify various modules of the system so that they can be tested stand-alone.
   b. Identify the groups of the module that can be tested together in integration.
8. **Case Study:** You are the technical lead in your company for a new product aimed at university research scientists: an electronic lab notebook (ELN). The marketing team has given you the following information. The scientists who will use the ELN traditionally record their experiments in paper notebooks, but increasingly they are looking for digital solutions which allow them to share their information with other scientists they are working with, such as Wikis and Dropbox, which make frequent functionality changes. A small amount of market research indicates that the scientists vary considerably in what the capabilities they would like in an ELN, and the ELN solution will need to change over time as the scientists' needs change. However, the scientists all say an ELN should be easy to use and quick to use. The scientists work in labs headed by a senior academic (the Lab Head). Provision of computing resources for the lab (hardware and software) varies between labs. Some have all their needs met by the central university IT department; others manage all their own hardware and software and depend on central IT only for network connections; most are somewhere in between.
   a. Which software development process is likely to be most appropriate for developing the software system?
   b. Explain why, contrasting your choice with one other software development process.
9. Discuss in detail various testing techniques.
10. Calculate the cyclometic complexity of the following control flow graph.

