ANDHRA KESARI UNIVERSITY

Programme: B.C.A. (HONOURS) Computer Applications (Major)
w.e.f. AY 2023-24

COURSE STRUCTURE

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
I	I	1	Fundamentals of Commerce	3+2	4
		2	Business Organization	3+2	4
	II	3	Office Automation Tools	3	3
			Office Automation Tools Lab	2	1
		4	Programming in C	3	3
			Programming in C Lab	2	1
II	III	5	Database Management System	3	3
			Database Management System Lab	2	1
		6	Data Structures	3	3
			Data Structures Lab	2	1
		7	Object Oriented Programming Through JAVA	3	3
			Object Oriented Programming Through JAVA Lab	2	1
		8	Software Engineering	3	3
			Software Engineering Lab	2	1
	IV	9	Python Programming	3	3
			Python Programming Lab	2	1
		10	Operating Systems	3	3
			Operating Systems Lab	2	1
		11	Mobile Application Development using Android	3	3
			Mobile Application Development using Android Lab	2	1

SEMESTER-III

COURSE 5: DATABASE MANAGEMENT SYSTEM

Theory Credits: 3 3 hrs/week

Course Objectives:

- Graduates will have the expertise in analyzing real time problems and providing appropriate solutions related to Computer Science & Engineering.
- Graduates will have the knowledge of fundamental principles and innovative technologies to succeed in higher studies and research.
- Graduates will continue to learn and to adapt technology developments combined with deep awareness of ethical responsibilities in profession.

Course Outcomes:

- An ability to apply Knowledge of computing and mathematics in Computer Science & Engineering.
- An ability to analyze a problem, identify and define the computing requirements appropriate to its solution.
- An ability to design, implement and evaluate a computer-based system to meet desired needs with appropriate societal considerations.
- An ability to conduct investigations, interpret data and provide conclusions in investigating complex problems related to Computer Science & Engineering.
- An ability to engage in continuing professional development and life-long learning.

UNIT- I

Overview of Database Systems: Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications.

Data Models: Introduction; types of data models, Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Case Study:

1. Describe the differences between Database systems and File based systems
2. Study about database models and their advantages and dis-advantages

UNIT- II

Relational Model: Introduction to relational model, Codd’s rules, concepts of domain, attribute, tuple, relation, constraints (Domain, Key constraints, integrity constraints) and their importance , concept of keys (super key, candidate key, primary key, surrogate key, foreign key) , relational Algebra & relational calculus.

Normalization: Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency(1NF, 2NF and 3 NF), Boyce-codd normal form(BCNF)

Case Study:
Describe Relational model and normalization for database design

UNIT - III:

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set,relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams,

BASIC SQL: Database schema, data types, DDL operations (create, alter, drop, rename), DML operations (insert, delete, update), basic SQL querying (select and project) using where clause, arithmetic & logical operations, aggregation, grouping, ordering.

Case Study:

1. Examine issues in data storage and query processing using SQL.
2. Create, maintain and manipulate a relational database using SQL

UNIT - IV

SQL: Nested queries/ sub queries, implementation of different types of joins, SQL functions(Date, Numeric, String, Conversion functions), Creating tables with relationship, implementation of key and integrity constraints, views, relational set operations , Transaction Control Language: commit, Rollback, Savepoint , DCL :Grant, Revoke

Case Study:

1. Try to convert some sample data to information and show how it can you be used in decision making.

UNIT –V

PL/SQL: Introduction , Structure , Control Structures , Cursors , Procedure , Function , Packages , Exception Handling ,Triggers.

Transaction processing Concepts : Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm.

Case Study:
Outline the role and issues in Transaction management of data such as efficiency, privacy, security.
<u>Database management systems Text Books</u>

- Database Management Systems, 3rd Edition , Raghurama Krishnan, Johannes Gehrke, TMH
- Database System Concepts,5th Edition , Silberschatz, Korth, TMH

SEMESTER-III
COURSE 5: DATABASE MANAGEMENT SYSTEM

Practical Credits: 1 2 hrs/week

List of Experiments

SQL :

Cycle-I: Aim: Marketing company wishes to computerize their operations by using following tables.

Table Name: Client- Master
Description: Used to store client information

Column Name	Data Type	Size	Attribute
CLIENT_NO	Varchar2	6	Primarykey
NAME	Varchar2	20	Not null
ADDRESS1	Varchar2	30	
ADDRESSS	Varchar2	30	
CITY	Varchar2	15	
PINCODE	Varchar2	8	
STATE	Varchar2	15	
BAL_DUE	Number	10,2	

Table Name: Product_Master
Description: Used to store product information

ColumnName	Data Type	Size	Attribute
PRODUCT_NO	Varchar2	6	Primarykey
DESCRIPTION	Varchar2	15	Not null
PROFIT _PERCENT	Number	4,2	Not null
UNIT_MEASUE	Varchar2	10	
QTY_ON_ HAND	Number	8	
REORDER_LVL	Number	8	
SELL_PRICE	Number	8,2	Not null, cannot be 0
COST _PRICE	Number	8,2	Not null,cannot be 0

Table Name: Salesman_master
Description: Used to store salesman information working for the company.

ColumnName	Data Type	Size	Attribut e
SALESMAN_NO	Varchar2	6	Primary key
SALESMAN_NAME	Varchar2	20	Not null
ADDRESS1	Varchar2	30	
ADDRESS2	Varchar2	30	
CITY	Varchar2	20	
PINCODE	Number	8	
STATE	Vachar2	20	
SAL_AMT	Number	8,2	Not null, cannotbe0
TGT_TO_GET	Number	6,2	Not null, cannotbe0
YTD_SALES	Number	6,2	Not null
REMARKS	Varchar2	20	

Table Name: SALES-ORDER Description:Used to store client's orders

ColumnName	Data Type	Size	Attribut e
ORDER_NO	Varchar2	6	Primarykey
CLIENT_NO	Varchar2	6	ForeignKey
ORDER _DATE	Date		
DELY_ADDRESS	Varchar2	25	
SALESMAN_NO	Varchar2	6	ForeignKey
DELY_TYPE	Char	1	Delivery:part(p)/full(f)anddefault'F'
BILL_YN	Char	1	
DELY_DATE	Date		Can'tbe lessthanorderdate
ORDER _STATUS	Varchar2	10	Values("InProcess","Fulfilled", "Back Order", "Cancelled.

Table Name: SALES_ORDER_DETAILS

Description:Used to store client’s order with details of each product ordered.

ColumnName	Data Type	Size	Attribute
ORDER_NO	Varchar2	6	Primary key references SALES_ORDER table
PRODUCT_NO	Varchar2	6	Foreign Key references SALES_ORDER_table
QTY_ ORDERED	Number	8	
QTY_ DISP	Number	8	
PRODUCT_RATE	Number	10,2	Foreign Key

Solve the following queries by using above tables.

1. Retrieve the list of names, city and the state of all the clients.
2. List all the clients who are located in ‘Mumbai’ or ‘Bangalore’.
3. List the various products available from the product_master table.
4. Find the names of salesman who have a salary equal to Rs.3000.
5. List the names o fall clients having ‘a’ as the second letter in their names.
6. List all clients whose Baldue is greater than value 1000.
7. List the clients who stay in a city whose first letter is ‘M’.
8. List all information from sales-order table for orders placed in the month of July.
9. List the products whose selling price is greater than 1000 and less than or equal to 3000.
10. Find the products whose selling price is greater than 1000 and also find the new selling price as original selling price 0.50.

Cycle-II Supplier

Aim: A manufacturing company deals with various parts and various suppliers supply these parts. It consists of three tables to record its entire information. Those are as follows.

Supplier (Supplier_No, Sname, City, status) Part(Part_no, pname, color, weight, city, cost) Shipment (supplier_No, Part_no, city) JX(project_no, project_name, city) SPJX(Supplier_no, part_no, project_no,city)

1. Get supplier numbers and status for suppliers in Chennai with status>20.
2. Get project names for projects supplied by supplier ’S’.
3. Get colors of parts supplied by supplier S·.
4. Get part numbers for parts supplied to any project in Mumbai.
5. Find the id’s of suppliers who supply a red or pink parts.

Cycle–III EmployeeDatabase

Aim: An enterprise wishes to maintain a database to automate its operations. Enterprise divided into a certain departments and each department consists of employees. The following two tables describes the automation schemas.

Emp(Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, Deptno)
Dept(Deptno, Dname, Loc)

1. List the details of employees who have joined before the end of September '81.
2. List the name of the employee and designation of the employee, who does not report to anybody.
3. List the name,salary and PF amount of all the employees(PF is calculated as10%of salary)
4. List the names of employees who are more than 2 years old in the organization.
5. Determine the number of employees, who are taking commission.
6. Update the employee salary by 20%,whose experience is greater than 12 years.
7. Determine the department does not contain any employees.
8. Create a view, which contains employee name and their manager names working in sales department.
9. Determine the employees, whose total salary is like the minimum salary of any department.
10. List the department numbers and number of employees in each department.

PL/SQL PROGRAMS

1. Writea PL/SQL program to check the given string is palindrome o rnot.
2. The HRD manager has decide to raise the employee salary by 15% write a PL/SQL block to accept the employee number and update the salary of that employee. Display appropriate message based on the existence of the record in Emp table.
3. Write a PL/SQL program to display top 10rows in Emp table based on their job and salary.
4. Write a PL/SQL program to raise the employee salary by 10% for department number 30 people and also maintain the raised details in the raise table.
5. Create a procedure to update the salaries of Employees by 20%, for those who are not getting commission
6. Write a PL/SQL procedure to prepare an electricity bill by using following table. Table used: Elect

Name	Null?	Type
MNNO	NOT NULL	NUMBER(3)
CNAME		VARCHAR2(20)
CUR_READ		NUMBER(5)
PREV_READ		NUMBER(5)
NO_ UNITS		NUMBER(5)
AMOUNT		NUMBER(8,2)
SER_TAX		NUMBER(8,2)
NET_AMT		NUMBER(9,2)

7. Create a trigger to avoid any transactions(insert, update, delete) on EMP table on Saturday & Sunday.

SEMESTER-III

COURSE 6: DATA STRUCTURES

Theory Credits: 3 3 hrs/week

Course Objectives:

- The objective of the course is to make a student to implement data structures and organize and manage data, based on data structures for efficient access.

Course Outcomes:

- Identify data structures suitable to solve any specific problem.
- Identifying various data structures and their real-time applications
- Identifying the use of Time and Space Complexity.
- Implementing different sorting & searching techniques.

Syllabus

Unit-I

Introduction and Overview- Elementary Data Organization, Data Structures classification, Data Structure Operations, Algorithms: Complexity, Time-Space Tradeoff.

Preliminaries-Mathematical Notation and Functions, Algorithmic Notation, Control Structures used in algorithms, Complexity of Algorithms. Other Asymptotic Notations, Sub algorithms, Variables, Data Types.

Case Study:

1. Calculate the space complexity of a given code

```
int tot (int a, int b)
{
        int c;
        c = a + b;
        return c;
}
```

Unit-II

Arrays, Records and Pointers – Linear Arrays, Representation and Traversing Linear Arrays, Inserting and Deleting. Passing an array to function, Pointer & Arrays Multidimensional Arrays, Sparse Matrices.

Case Study:

1. Application of arrays in the real world

Unit- III

Linked Lists – Representation, Dynamic Memory Allocation, Traversing, Searching, Insertion, Deletion, Header Linked Lists, Two-Way Lists

Stacks- Stacks, Operations on stacks, Array representation of stacks, Linked List representation of stacks, Arithmetic Expressions, Polish notation, Recursion.

Case Study:

1. Linked list verses Arrays.
2. Towers of Hanoi.

Unit- IV

Queues, Linked representation of Queues, Deques, Priority Queues.

Sorting - Insertion Sort, Bubble Sort, Selection sort, Quick Sort, Merge sort, Heap Sort, Searching – Linear Search, Binary Search.

Case Study:

1. Application of Queues.
2. Comparison of sorting algorithms.

Unit- V

Trees- Binary trees, Representing and traversing binary trees, Traversal algorithms using stacks.
Binary Search Trees, Searching, Insertion and Deletion in Binary Search Trees,

Graphs- Terminology, Sequential representation of Graphs, Linked representation of Graphs, Operations on Graphs, Traversing a Graph.

Case Study:

1. Applications of Binary Tree.
2. Warshall's Algorithm.

Text books:

1. Data Structures by Seymour Lipschutz, McGraw Hill(Schaum's Outlines).
2. Data Structures using C , Second edition , Dr. Reema Thareja, Oxford University Press.

REFERENCE BOOKS:

1. Data Structures & Algorithms Using C, Khanna Publishers
2. Theory and Problems of Data Structures by Seymour Lipschutz, McGraw Hill (Schaum's Outlines)
3. Data Structures & Algorithms in C by M.A.Weiss, Addison Wisley.
4. Data Structures Using C, Reema Thareja, oxford.

SEMESTER-III
COURSE 6: DATA STRUCTURES

Practical Credits: 1 2 hrs/week

List of Lab Experiments

1. Write a C program to Implement matrix multiplication.
2. Write a C program to Implement stack using arrays.
3. Write a C program to Implement queue using arrays.
4. Write a C program to Implement circular queue using arrays.
5. Write a C program to Implement dequeue using arrays.
6. Write a C program to Implement single linked list using the methods create(), insert(), search(), delete() and display().
7. Write a C program to Implement double linked list.
8. Write a C program to Implement stack using linked list.
9. Write a C program to Implement queue using linked list.
10. Give a solution to towers of Hanoi using C program.
11. Write a C program to Implement bubble sort.
12. Write a C program to Implement selection sort.
13. Write a C program to Implement insertion sort.
14. Write a C program to Implement merge sort.
15. Write a C program to Implement quick sort.

SEMESTER-III

COURSE 7: OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Theory Credits: 3 3 hrs/week

Course Objectives:

To make the students understand the fundamentals of Java programming.

- To expose the students to Window based applications using AWT
- To make the students to design appropriate Exception Handling in Java
- To make the students to understand the concepts of Threads Files and
- I/O Streams, Applets Networking in java.

Course Outcomes:

The student would become competent enough to write, debug, and document well-structured java applications

- Demonstrate good object-oriented programming skills in Java
- Able to describe recognize, apply, and implement selected design patterns in Java
- Understand the capabilities and limitations of Java
- Be familiar with common errors in Java and its associated libraries
- Develop excellent debugging skills

UNIT - I

Introduction to OOPS: Paradigms of Programming Languages – Basic concepts of Object Oriented Programming – Differences between Procedure Oriented Programming and Object Oriented programming - Benefits of OOPs – Application of OOPs. Java: History – Java features – Java Environment – JDK – API. Introduction to Java: Creating and Executing a Java program – Java Tokens- Java Virtual Machine (JVM) – Command Line Arguments –Comments in Java program. Elements: Constants – Variables – Data types - Scope of variables – Type casting – Operators: Special operators – Expressions – Evaluation of Expressions.

Case Study:

1. Study the evolution of JAVA, why it was developed, and how it changed the software industry scenario.

UNIT - II

Control Structures: The if Statement, Nested ifs, The if-else-if Ladder and, Looping Statements: The while Loop, The do-while Loop, for loop and its variations and Nested Loops. Jumping Statements: Break, continue Statement.

Class and objects: Defining a class – Methods – Creating objects – Accessing class members – Constructors – Parameterized Constructors, Adding a Constructor. Arrays: One Dimensional Array – Creating an array – Array processing – Multidimensional.

Case Study:

1. Study the difference between the looping structured in JAVA And Programming in C.
2. Study the limitation of Constructors in JAVA.

UNIT - III

Inheritance: Defining inheritance –types of inheritance– Method overloading – Static members – Nesting of Methods – this keyword - Overriding methods – Final variables and methods – Final classes – Final methods - Abstract methods and classes – Visibility Control.

Interfaces: Defining interface – Extending interface - Implementing Interface - Accessing interface variables. Strings: Constructing Strings, Operating on Strings, Arrays of Strings

Case Study:

1. Study the inheritance types available in JAVA and try to identify the limitations.

UNIT – IV

Packages: Java API Packages – Defining a Package, System Packages – Naming Conventions – Creating & Package Member Access – Adding Class to a Package.
Multithreading: Creating Threads – Life of a Thread – Defining & Running Thread – Thread Methods – Thread Priority – Synchronization –Implementing Runnable interface – Thread Scheduling.

Case Study:

1. Study the advantages of Package compared to Libraries in Procedural languages.

UNIT – V

Exception Handling: Limitations of Error handling – Advantages of Exception Handling - Types of Errors – Basics of Exception Handling - Syntax of Exception Handling Code, Multiple Catch Statements, Using finally Statement, Throwing Our Own Exceptions

Applets: Introduction, Java applications versus Java Applets, Applet Life-cycle, Working with Applets, The HTML Applet Tag.

Case Study:

1. Study and present the limitation of Applets in Web application development.

TEXT BOOKS:

1. **Object Oriented Programming through Java,** Universities Press, by P. Radha Krishna.
2. E. Balagurusamy, "*Programming with Java*", TataMc-Graw Hill, 5th Edition.

REFERENCES:

1. Herbert Schildt, "The complete reference Java", TataMc-Graw Hill, 7th Edition.

SEMESTER-III

COURSE 7: OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Practical Credits: 1 2 hrs/week

List of Lab Experiments

1 Write a program to print Biggest of 3 Numbers using Logical Operators.

2 Write a program to Test the Prime number.

3 Write a program to create a Simple class to find out the Area and perimeter of rectangle and box using super and this keyword.

4 Write a program to design a class account using the inheritance and static that show all function of bank(withdrawal, deposit).

5 Write a program to design a class using abstract methods and classes.

6 Write a program to design a string class that perform string method (equal, reverse the string, change case).

7 Write a program to handle the exception using try and multiple catch block.

8 Write a program that import the user define package and access the member variable of classes that contained by package.

9 Write a program that show the implementation of interface.

10 Write a program to create a thread that implement the runable interface.

11 Write a program to draw the line, rectangle, oval, text using the graphics method.

12 Write a program to create menu using the frame.

13 Write a program to create dialog box.

14 Write a program to implement the flow layout and border layout.

15 Write a program to create Frame that display the student information.

SEMESTER-III

COURSE 8: SOFTWARE ENGINEERING

Theory Credits: 3 3 hrs/week

Course Objectives: The Objective of the course is to assist the student in understanding the basic theory of software engineering, and to apply these basic theoretical principles to a group software development project.

Course Outcomes

1. Ability to gather and specify requirements of the software projects.
2. Ability to analyze software requirements with existing tools
3. Able to differentiate different testing methodologies
4. Able to understand and apply the basic project management practices in real life projects
5. Ability to work in a team as well as independently on software projects

Syllabus

UNIT I

Introduction to Software Engineering: Definitions - Size Factors - Quality and Productivity Factors – Managerial Issues.

Planning a software project: Defining the problem - Developing a Solution Strategy - Planning the Development Process - Planning an Organization structure - Other Planning Activities.

UNIT – II

Software Cost Estimation: Software cost factors - Software Cost.

Estimation Techniques – Staffing level Estimation- Estimating Software Maintenance Costs - The Software Requirements, Specification - Formal Specification Techniques - Languages andProcessors for Requirements Specification.

UNIT – III

Software design: Fundamental Design Concepts - Modules and Modularization Criteria – Design Notations -Design Techniques - Detailed Design Considerations.

Real-Time and Distributed System Design - Test Plans - Milestones, walkthroughs, and Inspections.

UNIT IV

User interface design and real time systems: User interface design - Human factors - Humancomputer interaction - Human - Computer Interface design - Interface design - Interface standards.

UNIT V

Software quality and testing: Software Quality Assurance - Quality metrics - Software Reliability - Software testing - Path testing – Control Structures testing - Black Box testing - Integration, Validation and system testing - Reverse Engineering and Reengineering.

CASE Tools: Projects management, tools - analysis and design tools – programming tools - integration and testing tool - Case studies.

REFERENCE BOOKS:

1. R.Fairley, Software Engineering Concepts, Tata McGraw-Hill, 1997.
2. R.S. Pressman, Software Engineering, Fourth Ed., McGraw Hill, 1997.
3. Software Engineering, H. Sommervill Ian , Addition Wesley Pub. Co.
4. Software Engineering: An object Oriented Perspective by Braude, E.J., Willey, 2001

Student Activity:

1. Visit any financial organization nearby and prepare requirement analysis report
2. Visit any industrial organization and prepare risk chart

SEMESTER-III
COURSE 8: SOFTWARE ENGINEERING

Practical Credits: 1 2 hrs/week

(Using Object Oriented Analysis and Design (OOAD))

Case Studies:

1. Student Marks Analysis System
2. E-Commerce Management System
3. Inventory Control System
4. Food Delivery Management system
5. Logistics Management System

Choose any two of above case studies and do the following exercises for that Case Study

1. Write the complete problem statement
2. Write the software requirements specification document
3. Draw the entity relationship diagram
4. Draw the data flow diagrams
5. Draw use case diagrams
6. Draw activity diagrams for all use cases
7. Draw sequence diagrams for all use cases
8. Draw collaboration diagram
9. Assign objects in sequence diagrams to classes and make class diagram.

Note: 1. To draw dataflow diagrams using Microsoft Visio Software, SmartDraw, etc…
2. To draw UML diagrams using Rational Rose Software, Star UML, etc.

SEMESTER-IV

COURSE 9: PYTHON PROGRAMMING

Theory Credits: 3 3 hrs/week

Unit-I

Getting Started with Python: Introduction to Python , Python Keywords , Identifiers , Variables , Comments, Data Types , Operators, Input and Output , Type Conversion , Debugging . Flow of Control, Selection , Indentation , Repetition , Break and Continue Statement , Nested Loops .

Strings- String Operations , Traversing a String , String handling Functions.

Case Study:

1. Study the features that make Python different from Procedural Languages.

Unit-II

Functions: Functions, Built-in Functions, User Defined Functions, recursive functions, Scope of a Variable

Python and OOP: Defining Classes, Defining and calling functions passing arguments, Inheritance, polymorphism, Modules – date time, math, Packages.

Exception Handling- Exception in python, Types of Exception, User-defined Exceptions.

Case Study:

1. Present a report of how Exception handling is different from JAVA Exceptional Handling.

Unit-III

List: Introduction to List, List Operations, Traversing a List, List Methods and Built-in Functions.

Tuples and Dictionaries, Introduction to Tuples, Tuple Operations, Tuple Methods and Built-in Functions, Nested Tuples. Introduction to Dictionaries, Dictionaries are Mutable, Dictionary Operations, Traversing a Dictionary, Dictionary Methods and Built-in functions.

Case Study:

1. What are the special features of dictionaries and try to analyze about the same features in any other language.

Unit-IV

Introduction to NumPy, Array , NumPy Array , Indexing and Slicing , Operations on Arrays , Concatenating Arrays , Reshaping Arrays , Splitting Arrays , Statistical Operations on Arrays.

Data Handling using Pandas , Introduction to Python Libraries, Series, DataFrame, Importing and Exporting Data between CSV Files and DataFrames, Pandas Series Vs NumPy ndarray.

Case Study:

1. Present a paper on advanced features of NumPy and Pandas.

Unit-V

Plotting Data using Matplotlib: Introduction, Plotting using Matplotlib –Line chart, Bar chart, Histogram, Scatter Chart, Pie Chart.

GUI Programming and Database Connectivity Using Python. Graphical User Interfaces. Usingthe Tkinter Module, Creating Label, Text, Buttons, info Dialog Boxes, Radiobutton, Checkbutton, Getting Input, Importing MySQL for Python , Connecting with a database, Forming a query in MySQL, Passing a query to MySQL.

Case Study:

1. Present a paper on the features and advantages of MySQL compared to other commercial Databases.

References:

1. Mark Lutz, Learning Python,5th Ed. O‟REILLY
2. Core Python Programming by Dr. R. Nageswara Rao
3. Problem Solving and Python Programming by E. Balaguru Swamy
4. Python programming: using problem solving approach by Reema Thareja.
5. Albert Lukaszewski ,MySQL for Python,Packet Publishing

SEMESTER-IV
COURSE 9: PYTHON PROGRAMMING

Practical Credits: 1 2 hrs/week

Lab Programs

1. Write a Program to check whether given number is Armstrong or not.
2. Write a Program to check whether given number is perfect or not.
3. Write a program to find factorial of given number using recursive function
4. Write a program to implement inheritance and polymorphism
5. Demonstrate a python code to print try, except and finally block statements
6. Write a program to demonstrate String handling functions
7. Write a program to input n numbers from the user. Store these numbers in a tuple. Print the maximum and minimum number from this tuple.
8. Write a program to enter names of employees and their salaries as input and store them in a dictionary
9. Write a program to implement statistical operations on arrays using numPy
10. Write a program to import and export CSV file to DataFrame.
11. Create the DataFrame Sales containing year wise sales and perform basic operation on it.
12. Visualize the plots using matplot lib.
13. Create GUI interface with different types button and labels
14. Create GUI interface and connect with MySQL database and perform CRUD(Create, Read, Update and Delete) operations.

SEMESTER-IV

COURSE 10: OPERATING SYSTEMS

Theory Credits: 3 3 hrs/week

Course Objectives:

1. To know the basic Structure, Components and Organization of Operating System.
2. To learn the notation of a Process- a Program in Execution, Management, Scheduling and Classic Problems of Synchronization.
3. To gain knowledge in various Memory Management Techniques.
4. To understand Unix Operating System and Various File operations.

Course Outcomes:

The students will be able to:

1. Understand the main components and Structure of Operating System & their functions.
2. Analyze various ways of Process Management & CPU Scheduling Algorithms.
3. Evaluate various device and resources like Memory, Time and CPU Management techniques in distributed systems.
4. Apply different methods for Preventing Deadlocks in a Computer System.
5. Create and build an Application/Service over the UNIX operating system.

Syllabus

Unit I

Introduction: What is Operating System? ,History and Evolution of OS, Basic OS Functions, Computer System Architecture, Operating System Structure.

System Structures: Operating System Services, User Operating System Interface, System Calls, Types of System Calls, Overview of UNIX Operating System, Basic Features of Unix Operating System.

Case Study :

1. Understanding and listing the basic differences between UNIX OS and Windows OS in usage, user interface, features etc.

Unit II

Process Management: Process Concept, Operation on Processes, Communication in Client-Server Systems.

Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, CPU Scheduling in UNIX.

Case Study:

1. Present your understanding on how CPU Scheduling is different in WINDOWS compared to UNIX/LINUX.

Unit III

Synchronization: Process Synchronization, Semaphores: Usage, Implementation, The Critical Section Problem., Classic problems of synchronization.

Deadlocks: Introduction, Deadlock Characterization, Necessary and Sufficient conditions for Deadlock, Deadlock Handling Approaches : Deadlock prevention, Deadlock Avoidance and Deadlock detection and Recovery .

Case Study:

1. Present your understanding of Deadlocks and new methodologies available in new Operating Systems released in the market.

Unit IV

Memory Management: Overview, Swapping, Contiguous Memory Allocation, Paging, Paging Examples, Segmentation, Page Replacement Algorithms, Memory management in UNIX.

Case Study:

1. Present a paper on new methods used in Memory management in the present day Operating Systems .

Unit V

Files and Directories in UNIX: Files, Directory Structure, File Operations, File System Implementation: File Allocation Methods, Comparison of UNIX and Windows.

Case Study:

1. Present a Paper on how UNIX treats regular files and directories differently from other operating systems.

TEXTBOOKS

1. Operating System Concepts: Abraham Silberschatz, Peter B. Galvin, GregGagne, 8th Edition,Wiley.
2. Unix and shell Programming by B.MH Arwani, OXFORD University Press.

REFERENCEBOOKS:

1. Operating System Principles, Abraham Silberchatz, PeterB.Galvin, GregGagne 8thEdition, WileyStudentEdition.
2. Principles of Operating Systems by Naresh Chauhan, OXFORD University Press.
3. Tanenbaum A S, Woodhull A S, Operating System Design and Implementation,3rd edition, PHI 2006.
4. Unix Shell Programming-YashwantKanetkar

SEMESTER-IV

COURSE 10: OPERATING SYSTEMS

Practical Credits: 1 2 hrs/week

List of Experiments

1. Introducing the LINUX Native editor vi: Working on basics of creating and editing a text file using standard commands of vi.
2. Introduction to UNIX Operating System, Compare with Windows OS. Writing and executing simple Hello World C Program in UNIX Environment.
3. Getting hands-on on basic UNIX Commands.
4. Write a program using the following system calls of UNIX OS fork, exec, getpid, exit, wait, close, opendir, readdir ?
5. Write a Simple shell script for basic arithmetic and logical calculations?
6. Write Shell script to check the given number is even or odd?
7. Write a shell script to swap the two integers?
8. Write Shell script to perform various operations on given strings.
9. Write Shell scripts to explore system variables such as PATH, HOME etc.
10. Write a shell script to display list of users currently logged in.
11. Write a shell script to delete all the temporary files.
12. Write a shell script to find the Factorial of a Number ?
13. Write C programs to implement the following Scheduling Algorithms:
    a) First Come First Serve.
    b) Shortest Job First.
    c) Round Robin.

Reference Text Books:

1. Brian W. Kernighanand Rob Pike,“The UNIX Programming Environment” Prentice Hall India (Edition available in LRCandin the form of E Book on student resource).
2. Yashwant Kanetkar,“UNIXShellProgramming”BPBPublications(FirstEdition).

SEMESTER-IV
COURSE 11: MOBILE APPLICATION DEVELOPMENT USING ANDROID

Theory	Credits: 3	3 hrs/week

COURSE OBJECTIVES:

1. To facilitate students understanding android SDK
2. To help students to gain a basic understanding of Android application development
3. To instill working knowledge of Android Studio development tool

COURSE OUTCOMES:

The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following industry-oriented COs associated with the above-mentioned competency:

1. Identify various concepts and features of Android operating system.
2. Configure Android environment and development tools.
3. Develop rich user Interfaces by using layouts and controls.
4. Use User Interface components for android application development.
5. Create Android application using database.
6. Publish Android applications.

UNIT-I

Introduction to Android: - Overview, History, Features of Android, The Android Platform, Understanding the Android Software Stack – Android Application Architecture –The Android Application Life Cycle – The Activity Life Cycle, Creating Android Activity -Views- Layout Android SDK, Android Installation, Building you First Android application, Understanding Anatomy of Android Application, Android Manifest file.

Case Study:

i. Give a brief description of Android Architecture and its parts.
ii. List out the challenges we face while using Android?
iii. List the new features of Android in the latest version.

UNIT-II

Android Application Design Essentials: Anatomy of an Android applications, Android terminologies, Creating User Interfaces with basic views- Application Context, Activities, Services, Intents, linking activities with Intents,, Receiving and Broadcasting Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions.

Case Study:

i. Present an idea that you would like to convert it into an application in the future.

UNIT-III

Android User Interface Design Essentials: User Interface Screen elements, Designing User Interfaces with Layouts, Drawing and Working with Animation. Layouts, Recycler View, List View, Grid View and Web view

Input Controls: Buttons, Checkboxes, Radio Buttons, Toggle Buttons, Spinners, Input Events, Menus, Toast, Dialogs, Styles and Themes, Creating lists, and Custom lists

Case Study:

i. Present detail report on the features of Check Boxes, Radio Buttons and Toggle Buttons.

UNIT-IV

Testing Android applications: Publishing Android application, Using Android preferences, Managing Application resources in a hierarchy, working with different types of resources.

Case Study:

1. List out the special features of Android with its counterparts.

UNIT-V

Using Common Android APIs: Internal Storage, External Storage , SQLite Databases , Managing data using Sqlite, Sharing Data between Applications with Content Providers, Using Android Networking APIs, Using Android Web APIs, JSON Parsing, Using Android Telephony APIs, Deploying Android Application to the World. Google maps, Using GPS to find current location, Sensors, bluetooth/Wi-Fi Connectivity.

Case Study:

i. List out the points to keep in mind to make you application more attractive.

ii. List the controls that make you application attractive.

REFERENCE BOOKS:

1. Reto Meier, "Professional Android 2 Application Development", Wiley India Pvt Ltd
2. Mark L Murphy, "Beginning Android", Wiley India Pvt Ltd
3. "Android Application Development All in one for Dummies" by Barry Burd, Edition: I
4. "Android", Dixit, Prasanna Kumar Vikas Publications, New Delhi 2014,
ISBN: 9789325977884
5. Maclean David, Komatineni Satya,Allen Grant , "Pro Android 5", Apress Publications2015
ISBN: 978-1-4302-4680-0
6." Android Programming for Beginners" by Hortan, John, Packet Publication, 2015
ISBN: 978-1-78588-326-2
7. Lauren Darcey and Shane Conder, "Android Wireless Application Development", Pearson Education,
2nd ed. (2011)

ONLINE READING / SUPPORTING MATERIAL:
1. http://www.developer.android.com
2. http://developer.android.com/about/versions/index.html
3. http://developer.android.com/training/basics/firstapp/index.html
4. http://docs.oracle.com/javase/tutorial/index.htm (Available in the form of free downloadable ebooks also).
5. http://developer.android.com/guide/components/activities.html
6. http://developer.android.com/guide/components/fundamentals.html
7. http://developer.android.com/guide/components/intents-filters.html.
8. http://developer.android.com/training/multiscreen/screensizes.html Syllabus of BCA (Honours) under CBCS 33 9. http://developer.android.com/guide/topics/ui/controls.html
10. http://developer.android.com/guide/topics/ui/declaring-layout.html
11. http://developer.android.com/training/basics/data-storage/databases.html

SEMESTER-IV

COURSE 11: MOBILE APPLICATION DEVELOPMENT USING ANDROID

Practical Credits: 1 2 hrs/week

LIST OF EXPERIMENTS:

1. Develop a program to implement frame layout, table layout and relative layout.
2. Develop a program to implement Text View and Edit Text.
3. Develop a program to implement Auto Complete Text View.
4. Develop a program to implement Button, Image Button and Toggle Button.
5. Develop a program to implement login window using above U1 controls.
6. Develop a program to implement Checkbox.
7. Develop a program to implement Radio Button and Radio Group.
8. Develop a program to implement Progress Bar.
9. Develop a program to implement List View, Grid View, Image View and Scroll View.
10. Develop a program to implement Custom Toast Alert.
11. Develop a program to implement Date and Time Picker.
12. Develop a program to create an activity. Develop a program to implement new activity using explicit intent and implicit intent.
13. Develop a program to implement content provider.
14. Develop a program to implement service.
15. Develop a program to implement broadcast receiver.
16. Develop a program to implement sensors.
17. Develop a program to build Camera.
18. Develop a program for providing Bluetooth connectivity.
19. Perform CRUD operations using SQLite.
20. Develop a program for JSON parsing.

ANDHRA KESARI UNIVERSITY-ONGOLE, PRAKASAM DISTRICT
Single Major Programme from the Year 2023-24 Onwards
Programme- B.CA. Biochemistry -Question Paper model,
Second Year-Semester-III & IV

Time: 3 Hours Total Marks: 75

PART –A

Answer any Five of the following

5X5=25 Marks

1.

2.

3.

4.

5.

6.

7.

8.

9.

10

PART –B

Answer the following One question from each unit

5x10=50 Marks

11a.

Or

11b.

12a.

Or

12b.

13a.

Or

13b.

14a.

Or

14b.

15a.

Or

15b.