SCHOOL OF ENGINEERING AND TECHNOLOGY

SRI PADMAVATI MAHILA VISVAVIDYALAYAM

(WOMEN'S UNIVERSITY)

TIRUPATI – 517502, ANDHRA PRADESH

Accredited by NAAC with 'A' Grade

ISO 9001:2021 Certified

Syllabus

for

Minor Degree Programme – R20

Department of Computer Science and Engineering

20CSTM01: BASICS OF PYTHON

Credits – 3 **Sessional Marks: 30**
L:T:P :: 3:0:0 **University Exam Marks: 70**

Course Objectives

To expose the students to the following:

1. The core syntax and semantics of Python programming language.
2. The various operators and control flow statements.
3. Various data structures and functions.
4. Modules, packages in python, object-oriented concepts, exception handling.
5. Advanced concepts like multithreading, graphics and various test cases.

Course Outcomes

After successful completion of course the student should be able to

CO1. Understand the basics of python programming languages
CO2. Develop simple programs with control structures
CO3. Apply data structures and make use of functions in developing programs/applications using python.
CO4. Build module level applications which may include packages and exception handling mechanisms.
CO5. Demonstrate projects that make use of libraries and generate test cases for the projects.

UNIT I

Python Programming Introduction - History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL (Shell) Running Python Scripts
Data Types - Variables, Assignment, Keywords, Input-Output, Indentation-Types - Integers, Strings, Booleans.

UNIT II

Operators - Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators
Expressions - Expressions and order of evaluations Control Flow- if, if-elif-else, for, while, break, continue.

UNIT III

Data Structures - Lists - Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences, Comprehensions
Functions - Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful, Functions (Function Returning Values) Scope of the Variables in a Function - Global and Local Variables

UNIT IV

MODULEs - Creating Modules, import statement, from. Import statement; name spacing, Python packages, Introduction to PIP, Installing Packages via PIP, Using Python Packages Object Oriented Programming OOP in Python: Classes, 'self-variable', Methods, Constructor, Method, Inheritance, Overriding Methods, Data hiding.
Error and Exceptions - Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions

UNIT V

Brief Tour of the Standard Library - Operating System Interface - String Pattern Matching, Mathematics, Internet Access, Dates and Times, Data Compression, Multithreading, GUI Programming, Turtle Graphics.

Testing -Why testing is required? Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.

Textbooks

1. Vamsi Kurama, "Python Programming: A Modern Approach", Pearson Publications, Kindle Edition, 2017.
2. Mark Lutz, "Learning Python: Powerful Object-Oriented Programming", Orielly Publishers, Fifth Edition, 2013.

Reference Books

1. Allen Downey, "Think Python", Green Tea Press, Second Edition, 2012.
2. Wesley J. Chun, "Core Python Programming", Pearson, Second Edition, 2006
3. Kenneth A. Lambert and B. L. Juneja, "Fundamentals of Python", Cengage Publications, 2015

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	L	-	-	H	M	-	-	-	-	-	-	-	H	-	-
CO2	-	L	M	H	-	-	-	-	-	-	-	-	-	H	L
CO3	L	-	M	H	-	-	-	-	-	-	-	-	-	H	L
CO4	-	-	L	M	H	-	-	-	-	-	-	-	H	M	L
CO5	-	-	-	-	H	L	-	M	-	-	-	-	L	M	H

20CSPM01: BASICS OF PYTHON LAB

Credits – 1 **Sessional Marks: 30**
L:T:P :: 3:1:0 **University Exam Marks: 70**

Course Objectives

To expose the students to the following:

1. Develop skills to solve the problems.
2. The core syntax and semantics of Python programming language.
3. How to develop various applications using python programming language

Course Outcomes

After successful completion of course the student should be able to

CO1 Develop simple applications using python.
CO2 Make use of functions in python scripts.
CO3 Deploy applications and packages necessary for applications.

LIST OF PROGRAMS

1. a) Run instructions in Interactive interpreter and a Python Script.
   b) Write a program to purposefully raise Indentation Error and correct it.
2. a) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem).
   b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.
3. a) Write a Program for checking whether the given number is a even number or not.
   b) Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . , 1/10.
   c) Write a program using a for loop that loops over a sequence. What is sequence?
   d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
4. a) Find the sum of all the primes below two million. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
   b) By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.
5. a) Write a program to count the numbers of characters in the given string and store them in a dictionary data structure
   b) Write a program to use split and join methods in the given string and trace a birthday with a dictionary data structure.
6. a) Write a program to combine two lists into a dictionary.
   b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?
7. a) Write a program to print each line of a file in reverse order.
   b) Write a program to compute the number of characters, words and lines in a file.
8. a) Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.
   Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius .If (distance between two balls centers) <= (sum of their radii) then (they are colliding)
   b) Find mean, median, mode for the given set of numbers in a list.

9. a) Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
   b) Write a function dups to find all duplicates in the list.
   c) Write a function unique to find all the unique elements of a list.
10. a) Write a function cumulative_product to compute cumulative product of a list of numbers.
   b) Write a function reverse to reverse a list. Without using the reverse function.
11. a) Write a program that defines and print a matrix.
   b) Write a program to perform addition of two square matrices.
   c) Write a program to perform multiplication of two square matrices.
12. a) Install packages requests, flask and explore them.
   b) Write a script that imports requests and fetch content from the page.
   c) Write a simple script that serves a simple HTTPResponse and a simple HTML Page.
   d) Write a program to implement class for ATM and include functions required for it.

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	L	-	M	L	H	-	-	-	-	-	-	L	H	-	-
CO2	-	L	M	-	H	-	-	-	-	-	-	H	H	-	L
CO3	L		M	-	H	-	-	-	-	-	-	H	-	M	H

20CSTM02: FUNDAMENTALS OF DATA STRUCTURES

Credits – 3	**Sessional Marks: 30**
L:T:P :: 3:0:0	**University Exam Marks: 70**

Course Objectives

To expose the students to the following:

1. Basic concepts of object-oriented programming, data structures and algorithms.
2. Linear data structure such as arrays, lists, stacks, queues and their applications.
3. Non-linear data structure such as Trees, graphs and their applications.
4. Various sorting, searching's and hashing techniques with their performance comparison.
5. Algorithms for solving problems with the help of fundamental data structures.

Course Outcomes

After successful completion of course the student should be able to

CO1. Acquire knowledge in object-oriented concepts and basics of data structures

CO2. Explain the significance and analyse various data structures and dynamic memory management techniques.

CO3. Design algorithms to perform operations with linear and Non-linear data structures.

CO4. Analyse and illustrate various techniques for searching, sorting and hashing.

CO5. Choose appropriate data structure to solve real world problem efficiently

UNIT I

Introduction to Object Oriented Programming: An overview of Object-oriented programming, classes and objects, constructors, destructors, templates, Data Abstraction, Inheritance, Overloading functions and operators, Polymorphism, Friend Functions, Inline Functions, Exception Handling.

UNIT II

Introduction to Data Structures: Definition, Classification, Applications, Arrays

Linked Lists - Singly linked list, doubly linked list, Circular linked list–operations and applications, Dynamic memory management.

Hashing – Indexing, Separate Chaining, Open Addressing and Double Hashing

UNIT III

Stacks and Queues: ADT Stack and its operations, Applications of Stacks: Expression Conversion and evaluation, ADT queue, Types of Queues: Simple Queue, Circular Queue, Priority Queue - Operations and applications.

UNIT IV

Nonlinear Data structure - Graphs: Basic Terminologies and Representations, Graph search and traversal algorithms-applications. **Trees:** Basic Tree Terminologies, Different types of Trees: Binary Tree, Binary Search Tree, B Tree, B+ Trees, and Applications of various tree structures.

UNIT V

Sorting: Introduction to internal sorting, Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, external sorting.

Searching: linear search, binary search

Textbooks

1. Yashavant Kanetkar, “Data Structures through C”, Third Edition BPB Publications, 2019
2. Herbert Schildt, “Complete Reference C++”, Fifth edition, Tata McGraw-Hill, 2015.
3. Ellis Horowitz, SartajSahni, “Fundamentals of Data Structures in C++”, Second Edition, Galgotia publication, 2007

Reference Books

1. BjarneStroustrup, “The C++ Programming Language”, 4th Edition, Addison-Wesley Professional, 2013.
2. Samanta D, “Classic Data Structures”, prentice hall India, 2/e, 2008.
3. Adam Drozdek, “Data Structures and Algorithms in C++”, 4th Edition, Cengage Learning, 2012.
4. SartajSahni, “Data Structures, Algorithms and Applications in C++”, Second Edition, Silicon Press, 2005.

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	H	L	-	-	-	-	-	-	-	-	-	H	-	-
CO2	M	H	L	-	-	-	-	-	-	-	-	-	H	-	-
CO3	L	M	H	-	-	-	-	-	-	-	-	-	H	-	-
CO4	-	H	M	L	-	-	-	-	-	-	-	-	H	-	-
CO5	-	M	L	H	-	-	-	-	-	-	-	-	-	-	H

20CSPM02: FUNDAMENTALS OF DATA STRUCTURES LAB

Credits – 1 **Sessional Marks: 40**
L:T:P :: 0:0:2 **University Exam Marks: 60**

Course Objectives

To expose the students to the following:

1. Simple linear and non-linear data structures.
2. Identify and apply the suitable data structure for the given real-world problem.
3. Various sorting and searching methods

Course Outcomes

After successful completion of course the student should be able to

CO1 Identify and implement the appropriate data structure for a given problem.
CO2 Determine and simulate the appropriate searching and sorting techniques for a given problem.
CO3 Implement various graph traversal techniques.
CO4 Design solutions for real life computational problems

LIST OF PROGRAMS

1. Define a class to represent a bank account which includes the following members as Data members: a) Name of the depositor b) Account Number c) Withdrawal amount d) Balance amount in the account
2. Member Functions:
    a) To assign initial values b) To deposit an amount c) To withdraw an amount after checking the balance d) To display name and balance.
3. Write the above program for handling n number of account holders using array of objects.
4. Write a C++ program to compute area of right-angle triangle, equilateral triangle, isosceles triangle using function overloading concept.
5. Write a C++ program to swap the values two integer members of different classes using friend function.
6. Define a class string and overload == to compare two strings and + operator for concatenation of two strings.
7. Consider an example of declaring the examination result. Design three classes student, exam and result. The student has data members such as roll no, name. Create the lass exam by inheriting the student class. The exam class adds data members representing the marks scored in 5 subjects. Derive the result from exam-class and it has own data members like total, avg. Write the interactive program into model this relationship
8. Write a program that uses stack operations to convert a given infix expression into its postfix Equivalent, Implement the stack using an array
9. Write a program to reverse the elements in the stack using recursion.
10. Write a program to implement the simple Queue and circular operations
11. Write a program that uses functions to perform the following: a) Create a singly linked list of integers. b) Delete a given integer from the above linked list. c) Display the contents of the above list after deletion.
12. Write a program that uses functions to perform the following: a) Create a doubly linked list of integers. b) Delete a given integer from the above doubly linked list. c) Display the contents of the above list after deletion
13. Write a program to implement Circular linked list operations

14. Determine the indegree and outdegree of all the vertices of a given graph.
15. Write programs for implementing the following graph traversal algorithms:
    a) Depth first traversal
    b) Breadth first traversal
16. Determine whether the given graph is connected graph or not.
17. Write a program that uses functions to perform the following: a) Create a binary search tree of characters. b) Traverse the above Binary search tree recursively in Postorder.
18. Write a program that uses functions to perform the following: a) Create a binary search tree of integers. b) Traverse the above Binary search tree non-recursively in inorder.
19. Write a program to implement B-tree
20. Write a program to implement B++ tree
21. Write programs for implementing the following sorting methods to arrange a list of integers in ascending order: a) Bubble sort b) Insertion sort
22. Write programs for implementing the following sorting methods to arrange a list of integers in ascending order: a) Quick sort b) Selection sort
23. Write programs for implementing the following sorting methods to arrange a list of integers in ascending order: a) Merge sort b) Heap sort
24. Write a program to implement external sorting technique.
25. Write a program to search for a given element using
    a) Linear search
    b) Binary search
26. Write a program to implement all the functions of a dictionary (ADT) using hashing.

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1		L		M	H			L					H	L	
CO2	M	M		L	H	L							H	L	
CO3		M			H	M		L					H	L	
CO4	L	M	H		H								H	L	

20CSTM03: INTRODUCTION TO DATBASE MANAGEMENT SYSTEMS

Credits – 3 **Sessional Marks: 30**
L:T:P :: 3:0:0 **University Exam Marks: 70**

Course Objectives

To expose the students to the following:
1. The basic concepts and the applications of database systems
2. The relational database design principles
3. The basics of SQL and construct queries using SQL
4. Normalize the data by using various techniques
5. Basic issues of transaction processing and concurrency control

Course Outcomes

After successful completion of course the student should be able to

CO1 Understand the knowledge of database concepts and query language
CO2 Acquire the knowledge and design ER model and relational model for database.
CO3 Design and build a simple database system and demonstrate competence with the fundamental tasks involved with modelling, designing, and implementing a DBMS.
CO4 Analyse the different Functional Dependencies and Functional Decomposition.
CO5 Apply various Normalization techniques.

UNIT I

Database System Applications, Purpose of Database Systems, View of Data – Data Abstraction – Instances and Schemas – data Models – the ER Model – Relational Model – Other Models – Database Languages – DDL – DML – database Access for applications Programs – data base Users and Administrator – Transaction Management – data base Architecture – Storage Manager – the Query Processor

UNIT II

Data base design and ER diagrams – ER Model - Entities, Attributes and Entity sets – Relationships and Relationship sets – ER Design Issues – Concept Design – Conceptual Design for University Enterprise

Introduction to the Relational Model – Structure – Database Schema, Keys – Schema Diagrams, Relational Query Languages, Relational Operations.

UNIT III

Overview of the SQL Query Language – Basic Structure of SQL Queries, Set Operations, Aggregate Functions – GROUPBY – HAVING, Nested Sub queries, Views, Triggers.

UNIT IV

Normalization – Introduction, Non-loss decomposition and functional dependencies, First, Second, and third normal forms – dependency preservation, Boyce/Codd normal form.

Higher Normal Forms - Introduction, Multi-valued dependencies and fourth normal form, Join Dependencies and Fifth normal form

UNIT V

Transaction Concepts- Transaction State- Implementation of Atomicity and Durability – Concurrent – Executions – Serializability- Recoverability – Implementation of Isolation – Testing for serializability- Lock –Based Protocols – Timestamp Based Protocols- Validation- Based Protocols – Multiple Granularity

Textbooks

1. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, "Database System Concepts", Tata McGraw Hill, Seventh Edition, 2021.
2. C.J. Date, A. Kannan, S. Swamynathan, "An Introduction to Database systems", Pearson Education, Eight Edition, 2006.

Reference Books

1. Ramez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", Pearson Education, Seventh Edition, 2016.
2. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", Tata McGraw Hill, Third Edition, 2014.

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	-	L	-	H	-	-	-	-	-	-	-	H	-	-
CO2	L	M	H	-	H	-	-	-	-	-	-	-	H	-	-
CO3	L	M	H	-	H	-	-	-	-	-	-	-	L	-	H
CO4	L	M	-	H	M	-	-	-	-	-	-	-	-	H	-
CO5	-	L	-	M	M	H	-	-	-	-	-	-	H	-	-

20CSPM03: INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS LAB

Credits – 1	**Sessional Marks: 40**
L:T:P :: 0:0:2	**University Exam Marks: 60**

Course Objectives

To expose the students to the following:

1. Basics of database management system.
2. How a real-world problem can be mapped to schemas.
3. Solve different industry level problems.

Course Outcomes

After successful completion of course the student should be able to

CO1 Define a problem at the view level.
CO2 Understand the physical structure of the database to handle data.
CO3 Implement the logic by using software.
CO4 Apply the concepts of transaction management for real time applications.

LIST OF EXPERIMENTS

1. Queries on DDL commands (Create, Alter, Drop, Rename)

Task 1:

a. Create a table with the following schema
Student (sid, sname, saddress, sphone)
b. Write a query to display structure of the table as student (sid, sname, saddress, sphone, smail)
c. Write a query to display structure of the table as student (sid, sname, sphone, smail)
d. Write a query to change the name of the column smail to smailid
e. Drop the table student

Task 2:

a. Create a table with the following schema
faculty (Fid, Fname, Faddress, Fbranch)
b. Write a query to display structure of the table as faculty (Fid, Fname, Faddress, Fbranch, Fphone)
c. Write a query to display structure of the table as faculty (Fid, Fname, Fbranch)
d. Drop the table faculty

2. Queries on DML commands (Insert, update, Delete, select)

a. Create a table with the following schema
Storeinfo (storename, sales, txn_date, storeaddress)
b. Insert 10 rows in to the table
c. Insert 15 rows into the table using single insert statement
d. Write a query to change the sales of Levis store from 20% to 30%
e. Write a query to change the address of VanHeusen store to 40-32, Himayat Nagar, Hyderabad.
f. Write a query to delete the details of Levis store.
g. Write a query to delete the details of a store with address 20/35, Ameerpet

h. Write a query to display all the details of storeinfo.
i. Write a query to retrieve the details of Levis store.

3. Queries on DCL, TCL commands and computations on queries

DCL(Grant,Revoke)
a. Write queries to create a role called as testing and create permission to testing
b. Write queries to revoke a create table privilege from testing role and drop the testing role.

TCL(Commit,Rollback,Savepoint)
a. Perform commit operation to save the changes permanently
b. Create a schema student(sid,sname) with 3 rows initially. Later insert 3 rows with 3 save points.
c. perform rollback to savepoint B, savepoint A.

4. Commands on key-constraints (NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK)
a. Create a table persons (id, fname, lname, age) where the null values should not be allowed while inserting the rows in id column.
b. Create a table persons such that all the values in column id are different.
c. Create a table student such that the column student id should not allow null values and duplicate values
d. Create a table persons (pid, fname, lname, age) and orders (id, ordernumber, pid).
e. Write a query to retrieve the ordered, ordername of a person with id 3.
f. Write a query to retrieve age of a person whose order id is 3.
g. Create a table votedetails (voterid, name, age, address) which does not allow the details of people whose age is <18.

5. Pattern matching queries and SQL queries using oracle functions
a. Write a query (WAQ) to retrieve the names of students whose names start with “S”.
b. Write a query to retrieve the names of the students whose names end with “a”.
c. Write a query to retrieve the names of the students whose names start with “ca”.
d. Write a query to retrieve the names of the students whose names consists of “ee”.
f. WAQ to concatenate 2 strings
g. WAQ to set the first character in uppercase and rest in lower case
h. WAQ to display the location of DER in “Hyderabad”.
i. WAQ to return the length of a string engineering.
j. WAQ to convert all letters in a string “HYDERABAD” to lower case and upper case also
k. WAQ to add NEW to the word HYDERABAD and NAWABS to Hyderabad.
l. WAQ to extract base from Data base Management systems
m. Write query to count number of students in student table
n. Write queries to demonstrate CEIL, FLOOR, GREATEST, LEAST, MAX, MIN, SUM.
o. Write queries to demonstrate date functions like ADD_MONTHS, CURRENT_DATE, LAST_DAY, MONTHS_BETWEEN, NEXT_DAY, ROUND, SYSDATE, SYSTIMESTAMP.

5. Implementing Group By, Having, Order by clause

Task 1:
a. Create a table Northzone (custid, custname, address, city, country)
WAQ to display the number of customers in each country.

b. WAQ to display the number of customers in each country in descending order
c. Create a table Northzone (custid, custname, age, address, city, country)
WAQ to display the number of customers in each country whose age is greater than 30.

Task 2:
a. For the schema student
WAQ to display number of students in each section
b. WAQ to display names of students in descending order
c. WAQ to display the names of students in each section whose percentage is>65.

6. Queries on joins (INNER, LEFT OUTER, RIGHT OUTER, FULL JOINS)

a. Create a table orders(orderid,custid,orderdate)
b. Create a table customers(custid,cname,country)
perform all join operations on the given two tables based on conditions.

Sub-Queries
a. WAQ to display list of children taller than 'myke' from height table.
b. WAQ to display the names of children who are taller and older than 'Jim'
c. WAQ to get the names of the employees who work in department with the highest budget.
d. WAQ to display the names of students whose percentage is greater than 65
e. WAQ to display the names of the employees whose salary is greater than the average of all salaries.
f. WAQ to demonstrate ALL, ANY.

7. Operations on views (Insert, Update, Delete), sequences
a. Create a students above 65 with the details sid,sname,sphone,smailid
b. WAQ to change the phone number of a student whose name is 'XYZ' in students above 65 view.
c. WAQ to drop students above 65 view.
d. WAQ to delete the students details whose sid is 123 from students above 65 view.

SEQUENCES
a. Create a sequence seq1 for a table class which starts with 1 increament by1 and max value is 999 with cycle and without cycle.
b. Insert the values into class using nextval.

8. SYNONYMS, CLUSTER and INDEX
a. Create a synonym for the product table in Adventure works2012
b. Create a cluster named personnel with a cluster key column department and a size of 512 bytes.
c. Create an index on the cluster key of personnel.

Introduction to basics of pl/sql programming
a. Write a program to print 'Hello'.
b. Write a program to add 2 numbers
c. Write a program to print greatest of 3 numbers.

9. Sample Programs in PL/SQL
a. Write a program to demonstrate basic loop, for loop, while loop
b. Write a program to demonstrate insert, update, delete and select using pl/sql.

Week 10,11,12: Mini Project

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	L	-	H	-	-	-	-	-	-	-	H	M	-
CO2	L	M	-	-	H	-	-	-	-	-	-	-	H	-	-
CO3	-	L	M	L	H	-	-	-	-	-	-	-	H	L	-
CO4	-	L	M	-	H	L	L	-	-	-	-	-	H	-	-

20CSTM04: DESIGN AND ANALYSIS OF ALGORITHMS

Credits – 3	**Sessional Marks: 30**
L:T:P :: 3:0:0	**University Exam Marks: 70**

Course Objectives

To expose the students to the following:

1. Knowledge in analyzing the efficiency and performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate major algorithms and data structures.
4. Concepts in algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.

Course Outcomes

After successful completion of course the students should be able to

CO1 Understand the fundamental concepts of various algorithms.
CO2 Analyse the performance of algorithms.
CO3 Apply appropriate algorithm design techniques for solving real time problems.
CO4 Choose the algorithmic design methods to test the impact on performance of algorithms.
CO5 Evaluate tractable and Intractable Problems.

UNIT I

Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behaviour; Performance measurements of Algorithm, randomized algorithms.

Divide and Conquer: General method, Binary Search, Merge sort, Quick Sort, Strassen’s matrix multiplication.

UNIT II

Greedy Method: General method, Minimum cost Spanning Trees, Knapsack problem

Dynamic Programming: General Method, Optimal binary search trees, 0/1 knapsack, The travelling sales person problem.

UNIT III

Graph and Tree Algorithms: Techniques for binary trees, Techniques for Graphs, connected components and Spanning trees, Bi-connected components and DFS

Back tracking: General Method, 8 – queens’ problem, Sum of subsets problem, Graph coloring and Hamiltonian cycles

UNIT IV

Branch and-Bound: The method, Travelling salesperson, 0/1 Knapsack problem, Efficiency considerations.

UNIT V

Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook’s theorem, Standard NP-complete problems and Reducibility.

Textbooks

1. Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, “Introduction to Algorithms”, 4thEdition, MIT Press/McGraw-Hill, 2014.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Second Edition, Universities Press, 2008.

Reference Books

1. Jon Kleinberg and Eva Tardos, “Algorithm Design”, 1stEdition, Pearson, 2013.
2. Michael T Goodrich and Roberto Tamassia, “Algorithm Design: Foundations, Analysis, and Internet Examples”, Second Edition, Wiley, 2006.
3. Udi Manber, “Algorithms—A Creative Approach”, 3rdEdition, Addison-Wesley, 2000.

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	-	-	-	-	-	-	-	-	-	-	-	H	-	-
CO2	L	M	-	H	-	-	-	-	-	-	-	-	H	-	-
CO3	-	L	-	M	-	H	-	-	-	-	-	-	L	-	H
CO4	L	M	-	H	-	-	-	-	-	-	-	-	L	-	H
CO5	L	M	H	-	-	-	-	-	-	-	-	-	H	-	-

20CSPM04: DESIGN AND ANALYSIS OF ALGORITHMS LAB

Credits – 1	**Sessional Marks: 40**
L:T:P :: 0:0:2	**University Exam Marks: 60**

Course Objectives

To expose the students to the following

1. Mathematical approach for Analysis of Algorithms.
2. Methods and techniques for analyzing the correctness and resource requirements of algorithms.
3. Different paradigms of algorithm design including recursive algorithms, divide-and-conquer algorithms, dynamic programming, greedy algorithms, Backtracking, Branch and Bound and graph algorithms.

Course Outcomes

After successful completion of course students should be able to

CO1 Apply divide and conquer algorithms for solving sorting, searching problems.

CO2 Correlate different tree traversal techniques for solving graph problems.

CO3 Compare the efficiencies of same problem using different algorithms (Ex. searching, sorting and graph traversal)

CO4 Apply greedy algorithms for finding solutions.

CO5 Analyse dynamic programing algorithms for calculating optimised solution of a problem.

LIST OF EXPERIMENTS

1. Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2. Using OpenMP, implement a parallelized Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
3. a. Obtain the Topological ordering of vertices in a given digraph
   b. Compute the transitive closure of a given directed graph using Warshall's algorithm.
4. Implement 0/1 Knapsack problem using Dynamic Programming.
5. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
6. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
7. a. Print all the nodes reachable from a given starting node in a digraph using BFS method.
   b. Check whether a given graph is connected or not using DFS method.
8. Find a subset of a given set S = {s1, s2, ..., sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1, 2,6} and {1,8}. A suitable message is to be displayed if the given problem instance doesn't have a solution.
9. Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.
10. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H		M	-	M	-	-	L	-	-	-	-	H	-	L
CO2	H	L	-	H	M	-	-	-	-	-	-	-	H	M	-
CO3	-	M	-	L	H	-	-	-	L	-	-	-	H	-	L
CO4	L	H	-	-	M	-	-	-	-	-	-	-	H	M	-
CO5	H	M	L	-	H	-	-	-	-	-	-	-	H	-	L

20CSTM05: WEB TECHNOLOGIES

Credits - 3	**Internal Marks:30**
L:T:P :: 3:0:0	**University Exam Marks:70**

Course Objectives

To expose the students to the following:

1. The fundamental concepts for developing web application using PHP language for server-side scripting
2. How data can be transported using XML
3. Develop a web application with server-side programming using java servlets & JSP Servlets
4. Develop a web application with client-side scripting with java script and AJAX

Course Outcomes

After successful completion of course the student should be able to

CO1 Understand the concepts of client-side scripting, validating of forms and AJAX programming.

CO2 Identify the role of server-side scripting using PHP programming

CO3 Create web pages using XML and explore how to parse and use XML Data with java.

CO4 Design dynamic web application using server-side programming with java servlets and JSP

CO5 Contrast on how to connect and retrieve data through a web page from database using JDBC

UNIT I

Introduction To Web Technologies: Introduction to html, fundamentals of HTML elements, document body, text, hyperlink, lists, tables, Color and Images, frames, cascading style Sheets: Introduction, defining your own styles, properties and values in styles, style sheets, formatting blocks, and layers; JavaScript: JavaScript basics, variables, string manipulation, mathematical functions, statements, operators, arrays and functions.

UNIT II

Objects in Javascript and Xml Objects in Javascript: Data and objects in JavaScript, regular expressions, exception handling, built-in objects, events; Dynamic HTML with JavaScript: Data validation, opening a new window, Rollover buttons, moving images, multiple pages in a single download, floating logos; XML: Basics XML, document type definition, xml schemas, Document Object Model, presenting XML.

UNIT III

SERVLETS AND JSP Servlet: Lifecycle of a Servlet, a simple Servlet, the servlet API, the Javax. servlet package, reading Servlet parameters, the javax.servlet. HTTP package, Handling HTTP requests and responses, using cookies and sessions. JSP: The anatomy of a JSP page, JSP processing, declarations, directives, expressions, code snippets, implicit objects, using beans in JSP pages, connecting to database in JSP.

UNIT IV

Introduction To Php: Introduction to PHP: Basics of PHP, downloading, installing, configuring PHP, programming in a web environment and the anatomy of a PHP page; Overview of PHP data types and concepts: Variables and data types, operators, expressions and statements, strings, arrays and functions.

UNIT V

PHP AND DATABASE ACCESS PHP and database access: Basic database concepts, connecting to a MySQL database, retrieving and displaying results, modifying, updating and deleting data; MVC architecture: PHP and other web technologies: PHP and XML, PHP and AJAX.

Textbooks

1. Chris Bates, “Web Programming: Building Internet Applications”, Wiley Dream Tech, 2nd Edition, 2002.
2. Jeffrey C K Jackson, “Web Technologies”, Pearson Education, 3rd Edition, 2010.
3. Steven Holzner, “The Complete Reference”, PHP, Tata McGraw-Hill, 1st Edition, 2007.

Reference Books

1. Hans Bergsten, “Java Server Pages”, O Reilly, 3rd Edition, 2003.
2. D. Flanagan, “Java Script”, O ‘Reilly, 6th Edition, 2011.
3. Jon Duckett, “Beginning Web Programming”, WROX, 2nd Edition, 2008.
4. Herbert Schildt, “Java the Complete Reference”, Tata McGraw-Hill - Osborne, 8th Edition, 2011.

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	L	M	-	-	-	-	-	-	-	-	-	M	H	L
CO2	L	M	-	H	-	-	-	-	-	-	-	-	M	H	L
CO3	L	M	H	-	-	-	-	-	-	-	-	-	M	H	L
CO4	L	-	H	M	-	-	-	-	-	-	-	-	M	H	L
CO5	L	-	M	H	-	-	-	-	-	-	-	-	M	H	L

20CSPM05: WEB TECHNOLOGIES LAB

Credits – 1 **Sessional Marks: 40**
L: T:P: 0:0:2 **University Exam Marks: 60**

Course Objectives

To expose the students to the following:

1. The web technologies to create adaptive web pages for web application.
2. How to develop a web application using java technologies
3. Create fully functional website/web application with MVC architecture.
4. What is XML and how to parse and use XML Data with Java.

Course Outcomes

After successful completion of course students should be able to

CO1 Write a well-formed / valid XML document.

CO2 Facilitate interface between frontend and backend of a web application.

CO3 Develop java program to a DBMS and perform insert, update and delete operations on DBMS table.

CO4 Write a server-side java application called Servlet to catch data sent from client, process it and store it on database.

LIST OF EXPERIMENTS

1. Design the following static web pages required for online book store. **1. Home page: -**the static home page must contain three pages **2. Top frame: -**logo and college name and links to homepage, login page, registration Page, catalogue page and cart page **3. Left frame: -**at least four links for navigation which will display the catalogue of Respective links **4. Right frame: -**the pages to links in the left frame must be loaded here initially it Contains the description of the website.
2. Design the following static web pages required for online book store. **1. Home page: -** the static home page must contain three pages **2. Top frame: -** logo and college name and links to homepage, login page, registration Page, catalogue page and cart page **3. Left frame: -** at least four links for navigation which will display the catalogue of Respective links **4. Right frame: -** the pages to links in the left frame must be loaded here initially it Contains the description of the website **5. Registration page** and **6. Cart page.**
3. Write a java script to validate the following fields in a registration page 1. Name (should contains alphabets and the length should not be less than 6 characters) 2. Password (should not be less than 6 characters) 3. E-mail (should not contain invalid addresses).
4. Write an XML file which displays the book details that includes the following: 1) Title of book 2) Author name 3) Edition 4) Price Write a DTD to validate the above XML file and display the details in a table (to do this use XSL).
5. Write an XML file which displays the book details that includes the following: 1) Title of book 2) Author name 3) Edition 4) Price Write a DTD to validate the above XML file and display the details in a table (to do this use XSL).
6. **User Authentication** Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a PHP for doing the following. 1. Create a Cookie and add these four user ID's and passwords to this Cookie. 2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies.

If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display “You are not an authenticated user’’.

7. **User Authentication** Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a PHP for doing the following. 1. Create a Cookie and add these four user ID’s and passwords to this Cookie. 2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies. If he is a valid user (i.e., user-name and password match) you should welcome him by name (user-name) else you should display “You are not an authenticated user.
8. Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number (these should hold the data from the registration form). Write a PHP program to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page

Course Outcomes – Program Outcomes – Program Specific Outcomes (CO-PO-PSO) Mapping

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	-	-	M	L	H	-	-	-	-	-	-	-	H	-	M
CO2	-	L	H	-	M	-	-	-	-	-	-	-	M	H	L
CO3	-	L	H	-	M	-	-	-	-	-	-	-	H	-	L
CO4	-	-	H	L	M	-	-	-	-	-	-	-	H	M	-