REGULATIONS AND SYLLABUS

(University Department)

(For the candidates admitted from the academic year 2018-2019 onwards)

MASTER OF COMPUTER APPLICATIONS (M.C.A)

(Under Choice Based Credit System)

PERIYAR UNIVERSITY

(Reaccredited with "A" Grade by the NAAC)

SALEM-636 011

TAMILNADU, INDIA

1. OBJECTIVES OF THE PROGRAMME

- To prepare Post Graduates who will be successful professionals in industry, government, academia, research, entrepreneurial pursuit and consulting firms.
- To prepare Post Graduates who will contribute to the society as broadly educated, expressive, ethical and responsible citizens with proven expertise.
- To prepare Post Graduates who will achieve peer-recognition; as an individual or in a team; through demonstration of good analytical, design and implementation skills.
- To prepare Post Graduates who will thrive to pursue life-long learning to fulfill their goals.

2. ELIGIBILITY FOR ADMISSION TO THREE YEAR M.C.A. PROGRAMME

Candidates who have passed in any one of the following or equivalent are eligible to apply:

(i) Bachelor's Degree (under 10+2+3/4) in any subject with Mathematics at +2 level

OR

(ii) Bachelor's Degree (under10+2+3/4) in any subject with Mathematics / Statistics as one of the subjects.

3. DURATION OF THE PROGRAMME AND MEDIUM

The programme shall be of three years duration spread over six semesters under choice based credit system. The Maximum duration to complete the programme shall be three academic years after normal completion of the programme. The medium of instruction/study is English.

4. DISTRIBUTION OF CREDITS

The **minimum** credit requirement shall be 135 credits for full time three year Master of Computer Applications programme and 90 credits for lateral entry students. The break-up of credits is as follows:

Core Courses	:	115
Elective Courses	:	18
Supportive Courses	:	06

5. COURSE OF STUDY

The course of study for the degree Master of Computer Applications shall be with internal assessment according to the syllabus prescribed from time to time.

Total Marks	:	4600
For each theory course	:	100 Marks (I.A: 25 + ESE: 75)
For each practical course	:	100 Marks (I.A: 40 + ESE: 60)
Industrial training	:	100 Marks (IA: 40 + ESE: 60) (Joint viva-voice by internal and external examiners)
Dissertation and Viva-voce	:	200 Marks

(I.A: 50 + Evaluation of the report by external 50 +Joint viva by internal and external examiners 100)

6. PROGRAMME STRUCTURE / SCHEME OF EXAMINATIONS

Course	Number of Credits	Hours Per Week	Examination Duration (hrs)	Marks		
				I. A	ESE	Total
Semester-I						
Course - 18UPCSC1C01 Digital Principles and Computer Organization	4	4	3	25	75	100
Course - 18UPCSC1C02 OOPS with Python	4	4	3	25	75	100
Course - 18UPCSC1C03 Data Structures	4	4	3	25	75	100
Course - 18UPCSC1C04 Front End Tool	4	4	3	25	75	100
Course - 18UPCSC1C05 Problem Solving Techniques	4	4	3	25	75	100

Course - 18UPCSC1C06 Problem Solving - **Lab**	2	4	3	40	60	100
Course – 18UPCSC1C07 Data structures – **Lab**	2	4	3	40	60	100
Course – 18UPCSC1C08 Python – **Lab**	1	2	3	40	60	100
SWAYAM/MOOC-I						
Total	25					800
Semester-II						
Course – 18UPCSC1C09 Relational Data Base Management Systems	4	4	3	25	75	100
Course – 18UPCSC1C10 Operating Systems	4	4	3	25	75	100
Course – 18UPCSC1C11 Software Engineering	4	4	3	25	75	100
Course – 18UPCSC1C12 Computer Networks	4	4	3	25	75	100
Supportive – I	3	3	3	25	75	100
Course – 18UPCSC1C13 Operating Systems–**Lab**	2	4	3	40	60	100
Course – 18UPCSC1C14 RDBMS–**Lab**	2	4	3	40	60	100
Course – 18UPCSC1C15 Financial Computing – **Lab**	1	2	3	40	60	100
SWAYAM-MOOC-II						
Total	24					800
Semester-III						
Course – 18UPCSC1C16	4	4	3	25	75	100
Course – 18UPCSC1C17	4	4	3	25	75	100

Course – 18UPCSC1C18	4	4	3	25	75	100
Elective Course – I	3	3	3	25	75	100
Supportive - II	3	4	3	25	75	100
Course – 18UPCSC1C19	2	3	3	40	60	100
Course – 18UPCSC1C20	2	3	3	40	60	100
Course – 18UPCSC1C21	2	2	3	40	60	100
Course – 18UPCSC1C22	1	2	3	100	-	100
SWAYAM/MOOC-III						
Total	25					900
Semester-IV						
Course – 18UPCSC1C23	4	4	3	25	75	100
Course – 18UPCSC1C24	4	4	3	25	75	100
Course – 18UPCSC1C25	4	4	3	25	75	100
Elective Course – II	3	3	3	25	75	100
Elective Course – III	3	3	3	25	75	100
Course – 18UPCSC1C26	2	3	3	40	60	100
Course – 18UPCSC1C27	2	3	3	40	60	100
Course – 18UPCSC1C28	1	2	3	40	60	100
Course – 18UPCSC1C29	1	2	3	100	-	100

Course – 18UPCSC1C30	2	-	-	40	60	100
Human Rights		2	3	25	75	100
SWAYAM-MOOC-IV						
Total	26					1100
Semester-V						
Course-18UPCSC1C31	4	4	3	25	75	100
Course – 18UPCSC1C32	4	4	3	25	75	100
Elective Course –IV	3	3	3	25	75	100
Elective Course –V	3	3	3	25	75	100
Elective Course – VI	3	3	3	25	75	100
Course-18UPCSC1C33	2	4	3	40	60	100
Course-18UPCSC1C34	2	4	3	40	60	100
Course-18UPCSC1C35	2	2	3	40	60	100
SWAYAM/MOOC-V						
Total	23					800
Semester-VI						
Course-18UPCSC1C36 Dissertation and Viva-Voce	16	-	-	50	50 100*	200
SWAYAM-MOOC-VI						
Total	16					200

Total no. of Credits **Core** **Elective** Supportive	 115 18 6					
Grand Total	139					
Total Marks						4600

***Joint via-voce**: **Internal Examiner : 50 Marks**

External Examiner : 50 Marks

External Evaluation : 50 Marks

6.1 List of Electives:

Elective Course - I

Course 18UPCSC1E01 -
Course 18UPCSC1E02 -
Course 18UPCSC1E03 -
Course 18UPCSC1E04 -

Elective Course - II

Course 18UPCSC1E05 -
Course 18UPCSC1E06 -
Course 18UPCSC1E07 -
Course 18UPCSC1E08 -

Elective Course - III

Course 18UPCSC1E09 -
Course 18UPCSC1E10 -
Course 18UPCSC1E11 -
Course 18UPCSC1E12 -

Elective Course – IV

Course 18UPCSC1E13 -
Course 18UPCSC1E14 -
Course 18UPCSC1E15 -
Course 18UPCSC1E16 -

Elective Course – V

Course 18UPCSC1E17 -

Course 18UPCSC1E18 -
Course 18UPCSC1E19
Course 18UPCSC1E20 -

Elective Course –VI

Course 18UPCSC1E21 -
Course 18UPCSC1E22 -
Course 18UPCSC1E23 -
Course 18UPCSC1E24 -

IA – Internal Assessments
ESE– End Semester Examinations

Core Course Code : 18UPCSC1C -
Elective Course Code : 18UPCSC1E -

7. EXAMINATIONS

7.1 THEORY EXAMINATION

7.1.1 Evaluation of Internal Assessment

Test	:	5 Marks (Best one out of Two Tests) :
Model Examination	:	5 Marks
Seminar	:	5 Marks
Assignment	:	5 Marks
Attendance	:	5 Marks

Total	:	25 Marks

There is no Minimum for Internal Assessment

7.1.2 Evaluation of End Semester Examinations

QUESTION PAPER PATTERN

Time Duration: 3 Hours Max. Marks: 75

PART- A: 20 x 1 = 20

4 questions from each unit

Answer all the questions

Objective type four questions from each unit

PART- B: 3 x 5 = 15

Answer three questions

One question from each unit

All questions must be of analytical

PART- C: 5 x 8 = 40

Answer all the questions

Either or type from each unit

7.2 PRACTICAL \ MINI PROJECT EXAMINATION

7.2.1 Evaluation of Internal Assessment

Test 1	:	20 Marks
Test 2	:	20 Marks (Best one out of Two Tests)
Test 3	:	20 Marks

Total	:	40 Marks

No Internal Minimum

7.2.2 Evaluation of End Semester Examinations

QUESTION PAPER PATTERN

Time duration: 3 Hours Max. Marks: 60

Questions may be taken from the list of practical problems: 60 Marks

Distribution of the Marks

(i) Practical

➢ Record Note Book	10
➢ Problem Understanding	10
➢ Implementation	20

➢ Debugging and Modification	10
➢ For correct output and viva	10

(ii) Industrial Training/Mini project

➢ Internal Assessment	40
➢ Joint Viva-voce	60

(Internal Examiner 30 and External Examiner 30)

(iii) Dissertation

➢ Internal Assessment	50
➢ Report Evaluation by External Examiner	50
➢ Joint Viva-voce	100

(Internal Examiner 50 and External Examiner 50)

8. REGULATIONS FOR DISSERTATION WORK

- Students should attach themselves with well reputed Industry/Company/Institutions to do their five months dissertation work.
- The students should submit the filled in format of the project proposal to the department for approval during the First week of December.
- The review of the dissertation will be carried out periodically.
- The student should submit three copies of their dissertation work.
- The students may use OHP/Power Point presentation during their dissertation Viva-Voce examinations.

9. PASSING MINIMUM

The candidate shall be declared to have passed in the theory/practical/dissertation examination if the candidate secures:

(i) 50% marks in the ESE and
(ii) 50% in ESE and IA put together

10. CLASSIFICATION OF SUCCESSFUL CANDIDATES

- ✓ Candidates who obtain 75% and above in the aggregate shall be deemed to have passed the examination in First Class with

Distinction provided they pass all the examinations prescribed for the programme at the first appearance.

- ✓ Candidates who secure not less than 60% of the aggregate marks in the examination shall be declared to have passed the examination in First Class.
- ✓ All other successful candidates shall be declared to have passed in Second Class.
- ✓ Candidates who pass all the examinations prescribed for the programme in first instance and within a period of three academic years from the year of admission are only eligible for University Ranking.

11. COMMENCEMENT OF THIS REGULATION

These regulations shall take effect from the academic year 2018-19, i.e., for students who are admitted to the first year of the programme during the academic year 2015-16 and thereafter.

12. TRANSITORY PROVISION

Candidates who were admitted to the PG programme of study before 2015-16 shall be permitted to appear for the examinations under those regulations for a period of three years after the completion of the programme i.e., up to and inclusive of the examination of April/May 2018. Thereafter, they will be permitted to appear for the examination only under the regulations then in force.

SEMESTER I

Course-18UPCSC1C01 Credits 4

DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION

Course objective

- To understand the role of basic number systems, Logic Gates, Flip Flops and data processing circuits.
- To prepare number system conversion and simplification problems
- This course enables the students to learn with the advanced concept of Central Processing Unit, Input/output Unit and Memory Unit.

Unit-I

Number Systems: Binary Number System, Binary-to-decimal Conversion, Decimal-to-binary Conversion, Octal Numbers, Hexadecimal Numbers, The ASCII Code, The Excess-3 Code, The Gray Code, -Binary Addition, Subtraction, Multiplication and Division Codes – BCD Weighted - Excess-Gray -Error Detection Codes. Basic Logic Gates – Boolean laws and theorems - Sum of products – product of sums - Karnaugh map simplification methods - don't care conditions. *Case study:* Add two gray coded numbers 0100 and 0111 and express the result in gray code.

Unit-II

Data processing circuits – Multiplexers – Demultiplexers – Decoders – Encoders –Arithmetic Building Blocks: Half and Full Adder: Subtractor, adder - TTL circuits – CMOS circuit. Flip-Flops: RS, Clocked RS, D-Edge – Triggered D, JK, Master/slave flip-flop-clocks and timers-counters, Asynchronous counters, Synchronous counters-MOD3, MOD5, Shift counters. *Case study* : Show how data processing circuits can be used to compare two 2-bit numbers, A1 A0 and B1 B0 to generate two outputs, A>Band A= B.

Unit-III

Central Processing Unit: Introduction – General Register Organization – Stack Organization – Instruction Formats – Addressing Modes – Data Transfer and Manipulation – Program Control Reduced Instruction Set Computer (RISC). *Case study*: Convert the following numerical arithmetic expression into reverse Polish notation and show the stack operations for evaluating the numerical result. (3+4)[10(2+6) + 8]

Unit-IV

Input - Output Organization: Peripheral Devices – Input-Output Interface Asynchronous Data Transfer – Modes of Transfer – Priority Interrupt – Direct Memory Access (DMA) – Input- Output Processor (IOP) – Serial Communication. *Case study:* Write your full name in ASCII using eight bits per character with the leftmost bit always 0. Include a space between names and a period after middle initial.

Unit-V

Memory Organization: Memory Hierarchy – Main Memory – Auxiliary Memory – Associative Memory – Cache Memory – Virtual Memory – Memory Management Hardware. Control Unit: Control Memory – Address Sequencing, Conditional Branching Mapping of Instruction Subroutines – Design of Control Unit, Micro program Sequencer, Problems.

Case study: A magnetic disk system has the following parameters:

T_s = average time position the magnetic head over a track
R = rotation speed of disk in revolutions per second
N = number of bits per track
N_s = number of bits per sector

Calculate the average time T_a that it will take to read one sector.

Course outcome

After successful completion of the course students will,

- Familiarize the number system concepts thoroughly.
- Understand the design and functioning of a machines central processing unit (CPU).
- Comprehend basic input/output functioning including program controlled I/O and interrupt I/O.
- Be through with organization of memory hierarchies including Cache and Virtual Memory.

Text Book

1. Albert Paul Malvino, Donals P. Leach, Goutam Saha, Digital Principles and Applications, McGrawHill, SeventhEdition,2011.Chapters: 4(1,2,3,6), 5(1,2,3,4,5,6,7,8), 6(1,2,11)
2. M. Morris Mano, Computer System Architecture, Prentice-Hall India Third Edition,
   2005. Chapters 7(1, 2, 4), 8, 11, 12.

Reference

1. Thomas C. Bartee, -Digital Computer Fundamentals, Tata McGraw Hill, 1996.
2. M. Morris Mano, -Digital Logic and Computer Design Prentice –Hall of India, 1979.
3. Albert Paul Malvino, Donals P. Leach, Goutam Saha, —Digital Principles and Applications, McGraw Hill, Seventh Edition, 2010.Chapters: 6(1,2,9), 4 (1to 6), 6(7,8),7(1,2,4), 8(1,2,4,5), 10(1,3,7), 13(10).

Course-18UPCSC1C02 **Credits 4**

OBJECT ORIENTED PROGRAMMING WITH PYTHON

Course objective

- Core programming basics and design in Python programming language.
- Fundamental principles of Object-Oriented Programming

Unit – I

Rapid Introduction to Procedural Programming: Creating and Running Python Programs - Data Types - Object References - Collection Data Types - Logical Operations - Control Flow Statements - Arithmetic Operators - Input/Output - Creating and Calling Functions – Data Types: Identifiers and Keywords - Integral Types - Integers - Booleans - Floating-Point Types - Floating-Point Numbers - Complex Numbers - Decimal - Strings - Comparing - Slicing and Striding Strings - String Operators and Methods - Formatting with the str.format() Method - Character Encodings.

Unit – II

Collection Data: Sequence Types - Tuples - Named Tuples - Lists - Set Types - Sets - Frozen Sets - Mapping Types - Dictionaries - Default Dictionaries- Ordered Dictionaries - Iterating and Copying Collections - Iterators and Iterable Operations and Functions - Copying Collections - Control Structures and Functions : Control Structures - Conditional Branching - Looping - Exception Handling - Catching and Raising Exceptions - Custom Exceptions - Custom Functions - Names and Docstrings - Argument and Parameter Unpacking - Accessing Variables in the Global Scope - Lambda Functions – Assertions

Unit – III

Modules : Modules and Packages - Packages - Custom Modules - Overview of Python's Standard Library - String Handling - Command-Line Programming - Mathematics and Numbers - Times and Dates - Algorithms and Collection Data Types - File Formats, Encodings, and Data Persistence - File, Directory, and Process Handling - Networking and Internet Programming - XML - Other Modules - Object-oriented Design : Introducing object-oriented - Objects and classes - Specifying attributes and behaviors - Data describes objects - Behaviors are actions - Hiding details and creating the public interface - Composition - Inheritance - Inheritance provides abstraction - Multiple inheritance

Unit – IV

Objects in Python - Creating Python classes - Adding attributes - Making it do something - Initializing the object - Explaining yourself - Data

Accessing - Third-party libraries - Basic inheritance - Extending built-ins - Overriding and super - Multiple inheritance - The diamond problem - Different sets of arguments - Polymorphism - Abstract base classes - Using an abstract base class - Creating an abstract base class -Demystifying the magic

Unit – V

Expecting the Unexpected: Raising an exception - The effects of an exception - Handling exceptions - The exception hierarchy - Defining our own exceptions - When to Use Object-oriented Programming - Treat objects as objects - Adding behavior to class data with properties - Properties in detail - Decorators – another way to create properties - Deciding when to use properties - Manager objects - Removing duplicate code - An alternative to method overloading - Default arguments - Variable argument lists - Unpacking arguments - Functions are objects too - Using functions as attributes - Callable objects

Course outcome

At the end of the course, Students will able

- To design and develop simple Python programs.
- To understand and implement the OOPs concept in Python Programming language effectively.

Text book

1. Dusty Phillips, "Python 3 Object-oriented Programming", Second Edition, PACKT Publishing, 2015. Chapters – 1-5,7 (http://file.allitebooks.com/20160830/Python%203%20Object-Oriented%20Programming, % 20 Second%20Edition.pdf)
2. Mark Summerfield, "Programming in Python 3 A Complete Introduction to the Python Language", Second Edition, Pearson Education, 2010. Chapters – 1-5 (http://file.allitebooks.com/20150514/Programming%20in%20Python%203,%202nd%20Edition.pdf)

Reference

1. Kenneth A. Lambert, "Fundamentals of Python: First Programs", CENGAGE Learning,2012.
2. Paul Gries, Jennifer Campbell and Jason Montojo, "Practical Programming: An Introductionto Computer Science using Python 3", Second edition, Pragmatic Programmers, LLC,2013
3. Robert Sedgewick, Kevin Wayne, Robert Dondero, "Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.
4. Timothy A. Budd, "Exploring Python", Mc-Graw Hill Education (India) Private Ltd.,, 2015.

Course-18UPCSC1C03 **Credits: 4**

DATA STRUCTURES

Course objective

The main objective of this course is to

- Teach efficient storage mechanisms of data for an easy access.
- Design and implementation of various basic and advanced data structures in C++.
- Introduce various techniques for representation of the data in the real world.

Unit –I

Introduction to Data Structures and Algorithms –Linked Lists: Introduction - Singly linked lists - Circular linked lists - Doubly linked lists – Circular Doubly linked lists – Applications of linked lists- Polynomial Representation - Stacks: Introduction to stacks – Array Representations of Stacks – operations on stack – Linked representations of stack – Operation on a linked stack – Applications of Stacks. Queues: Introduction to Queues – Array and linked representations of Queues – Types of Queues – Applications of Queues.

Unit – II

Trees: Introduction – Types of Trees: General Trees – Forests – Binary Trees – Binary Search Trees. Traversing a Binary Tree: Pre-order Traversal – In-order Traversal – Post-order Traversal – Level-order Traversal - Applications of Trees. Efficient Binary Trees: Binary Search Trees – Operations on Binary Search Trees: Searching for a node in a Binary Search Tree – Inserting a new node in a Binary Search Tree – Deleting a node from a Binary Search Tree – Determining Height and number of nodes in a Binary Search Tree.AVL Trees: Operations on AVL Trees – Searching for a node in an AVL Tree.

Unit – III

Heaps: Binary Heaps – Inserting a new element in a Binary Heap – Deleting a element from a Binary Heap – Applications of Binary Heaps. Graphs: Introduction – Graph Terminology – Directed Graph –Bi-connected Components – Representation of Graphs - Graph Traversal Algorithms: Breadth first search algorithm – Depth first search algorithm. Shortest Path Algorithms: Minimum Spanning Tree – Prim's algorithm – Kruskal's algorithm – Dijkstra's algorithm – Applications of Graphs.

Unit-IV

Sorting: Introduction to Sorting – sorting on multiple keys - Bubble Sort: Technique – Complexity of Bubble sort - Insertion Sort: Technique – Complexity and advantage of insertion sort. Selection Sort: Technique – Complexity and advantage of Selection sort - Merge Sort – Quick Sort: Technique – Complexity and advantage of Quick sort – Radix Sort – Heap Sort – Shell Sort – Tree Sort – Comparison of Sorting Algorithms – External Sorting.

Unit-V

Hashing and Collision: Introduction – Hash Tables – Hash Functions – Different Hash Function: Division method - Multiplication method – Mid Square method – Folding Method – Collision: Collision Resolution by open addressing – Collision Resolution by Chaining – Pros and cons of Hashing – Applications of Hashing.

Course outcome

At the end of the course, Students will able

- To design and implement simple and advanced data structure concepts in C++.
- To develop the real time application using data structure.

Text Book

1. ReemaThareja, **"Data Structures using C",** Oxford University Press, Second Edition, 2014.
   Unit I: Chapters (2, 6 (6.1-6.5,6.8), 7 (7.1-7.5, 7.7),8)
   Unit II: Chapters (9 (9.1, 9.2 (9.2.1-9.2.4), 9.4, 9.6), 10 (10.1,10.2 (10.2.1 – 10.2.5), 10.4))
   Unit III: Chapters (12 (12.1, 12.5),13 (13.1 – 13.6, 13.8 (13.8.1-13.8.4)))
   Unit IV: Chapter (14 (14.6 - 14.17))
   Unit V: Chapter (15 (15.1 – 15.7))

Reference

1. Horowitz, Sahni and Anderson-Freed, "Fundamentals of Data Structures in C", University Press, Second Edition, 2008.
2. YashavantKanetkar, "Data Structures Through C", BPB Publications, Second Edition, 2009.
3. Srivastava, "Data Structures Through C in Depth", BPB Publications, Second Edition, 2011.

Course-18UPCSC1C04 **Credits 4**

FRONT END TOOL

Course objective

- To introduce computer programming using the Visual BASIC programming language with object-oriented programming principles.
- To emphasis on event-driven programming methods, including creating and manipulating objects, classes, and implementation by using Microsoft Visual Studio 2008 and the Microsoft .NET Framework 3.5.

Unit–I

Introduction: Visual Studio Environment- Visual Basic Project –Visual Basic Code Statements – Run the Project- Finding and Fixing Errors – Controls- Working with Multiple Controls- Designing your Applications for User Convenience.

Unit-II

Data - Variables and Constants –Formatting Data for Displaying – Handling Exceptions – Displaying Messages in Message Boxes- List Boxes and Combo Boxes -Decision and Conditions: If Statements – Boolean Expressions- Nested If Statements – Input Validation – Case Structure- Do-Loops- Sharing an Event Procedure- Calling an Event Procedure – Debugging Visual Basic Projects.

Unit–III

Menus – Common Dialog Boxes – Creating Context Menus – Writing General Procedure – Using Multiple Forms – Variable and constants in Multiple Forms- Arrays: Single Dimensional Arrays- For Each/Next Statements – Structures – Multi Dimensional Array.

Unit–IV

Database Applications: Databases – Using ADO. Net and Visual Basics-Creating a Database Application – Binding Individual Data Fields – Selecting Records from the List – Selecting Records using Web Forms

Unit-V

Web Applications: Visual Basic and Web Programming – Creating a Website – Laying out Web Forms- Navigating a Web Forms – Using the Validating Controls- AJAX – Graphics in Windows and Web- Simple Animation- Timer Component- Playing Sounds – Drag and Drop Programming.

Course outcome

- Able to build various Visual BASIC applications

Text Book

1. Julia Case Bradley, Anita C. Millspaugh, “PROGRAMMING IN VISUAL BASIC 2008”.—7th ed., McGraw-Hill, 2009.
   Chapters: 1 – 10, 13.

Reference Book:

1. Tim Patrick, “Programming Visual Basic 2008”, 1st ed. O’Reilly Media, 2008.

Course-18UPCSC1C05 **Credits 4**

PROBLEM SOLVING TECHNIQUES

Course Objective:

- To identify methods appropriate for solving problems
- To apply problem solving skills when working with real time projects

Unit-I

Introduction - The Problem-solving Aspect - Top-down Design-implementation of Algorithms- Program Verification - The Efficiency of Algorithms. Fundamental Algorithms - Exchanging the values of Two Variables - Counting - Summation of a set of Numbers - Factorial Computation-Sine function computation - Generation of the Fibonacci sequence - Reversing the Digits of an Integer - Base Conversion Character to Number Conversion.

Unit-II

Finding the square Root of a number - The Smallest Divisor of an Integer - The Greatest Common Divisor of Two Integers - Generating Prime Numbers - Computing the Prime Factors of an Integer - Generation of Pseudo - random Numbers - Raising a Number to a Large Power - Computing the nth Fibonacci Number.

Unit-III

Array Order Reversal-Array Counting or Histogramming - Finding the Maximum Number in a Set - Removal of Duplicates from an Ordered Array - Partitioning an Array – Finding the kth Smallest Element - Longest Monotone Subsequence.

Unit-IV

Overview of C – Constants, Variables, Data types – Operators – Expressions – The Decision Control Structure – The Loop control Structure – The Case Control Structure – Functions and Pointers – Arrays – Pointers and Arrays – Two Dimensional Arrays – Array of Pointers – Three Dimensional Array – Puppetting on Strings – Structures.

Unit-V

File Input / Output – Data Organization – File Operations – File Modes – String (Line) or Record I/O in Files – Text Files and Binary Files – Database Management - Miscellaneous Features – Enumerated Data Type – Typedef – Typecasting – Bit Fields – Pointers to Functions – Functions Returning Pointers – Union of Structures – C Under Windows – DOS and Windows Programming Model.

Course outcome

- Employ good software engineering practices such as incremental development, data integrity checking and adherence to style guidelines.
- Select and model data using primitive and structured type

Text Book

1. R.G.Dromey, " How to Solve it by Computer ", Pearson Education, India, 2007.
   (Chapters: 1, 2, 3, and 4)
2. Yashavant P. Kanetkar, "Let Us C", Fifth Edition, Sridhara Publication, India, 2008.
   (Chapters: 1, 2, 3, 4, 5, 8, 9, 10, 12, 15, 16, 19)

Reference

1. SeymourLipschutz, "Essentials Computer Mathematics", Schaums" outlines series, Tata McGrawHill Edition, 2004.
2. Kernigan Brian W., and Dennis M. Ritchie, "The C Programming Language", Second Edition, Prentice Hall, 1988.
3. Balagurusamy E, "Programming in ANSI C", Third Edition, Tata McGraw Hill, 2004.

Course-18UPCSC1C06 **Credits: 2**

PROBLEM SOLVING –LAB

Course objective

- The students will be able to enhance their analyzing and problem solving skills and use the same for writing programs in C

List of Programs

1. Write a C program to compute the factorial of an integer
2. Write a C program to generate the Fibonacci sequence using recursive function
3. Write a C program to check whether the word is palindromeor not
4. Write a C program to find the square root of a given number
5. Write a C program to generate prime numbers upto an integer
6. Write a C program to find the GCD of two integers
7. Write a C program to remove the duplicates from an ordered array
8. Write a C program to find smallest and largest number present in an array
9. Write a C program to sort the elements of an increasing order
10. Create a C Program with structure to specify data on students given below

    Roll no, Name, Department, Course, Year of joining. Assume that there are 100 students in the department

    a) Write a function to print names of all students who joined in a particular year.
    b) Write a function to print the data of students whose roll number is given.

Course Outcome

- Know concepts in problem solving ·
- To do programming in C language ·
- To write diversified solutions using C language

Course-18UPCSC1C07 **Credits: 2**

DATA STRUCTURE LAB USING C

Course objective

- To develop skills to design and analyze simple linear and nonlinear data structures using C
- To Strengthen the ability to identify and apply the suitable data structure for the given real world problem.
- To Gain knowledge in practical applications of data structures using C

To implement the following using C

1. Stack
2. Queue
3. Singly Linked List
4. Circular Linked List
5. Doubly Linked List
6. Polynomial Addition using Singly Linked List
7. Tree traversal
8. Quick Sort
9. Merge Sort
10. Heap Sort

Course outcome

- Able to design and analyze the time and space efficiency of the data structure.
- Able to identify the appropriate data structure for given problem.
- Have practical knowledge on the application of data structures.

Course-18UPCSC1C06 **Credits: 1**

PYTHON LAB

Course objective

- To Learn Syntax and Semantics and create Functions in Python.
- To Implement Object Oriented Programming concepts in Python

Implement the following in Python 3 version:

1. Arithmetic and Boolean Operations
2. String Operations
3. Control Structures : Conditional and Looping
4. Creation of User-defined Functions
5. Exceptional Handling
6. Create and Import Built-in and Custom Modules
7. Working of classes and objects
8. Class method and static method
9. Constructors
10. Inheritance

Course outcome

- Learn and implement various data structures in python
- Develop python applications using oops.

SEMESTER II

Course-18UPCSC1C09 Credits: 4

RELATIONAL DATABASE MANAGEMENT SYSTEMS

Course objective

- To understand the different issues involved in the design and implementation of a database system.
- To study the physical and logical database designs, database modeling, relational, hierarchical, and network models.
- To understand and use data manipulation language to query, update, and manage a database
- To develop an understanding of essential DBMS concepts such as: database security, integrity, and concurrency.
- Analyze and design a real database application.

Unit-I

Introduction to Database Systems: Overview – Data Models – Database System Architecture– History of Database Systems. Diagram – Weak Entity Sets – Extended E-R Features –Design of an E-R Database Schema – Reduction of E-R Schema to Tables. Entity Relationship Model: Basic Concepts – Constraints – Keys – Design Issues – Entity Relationship.

Unit-II

Relational Model: Structure of Relational Databases – Relational Algebra – Extended Relational Algebra Operations – Modification of Database – Views – Tuple Relational Calculus – Domain Relational Calculus. SQL: Basic Structure – Set Operations – Aggregate Functions – Null Values – Nested Sub queries –Modification of the database – Joined Relations– Embedded SQL – Dynamic SQL –Query-by-Example, Quel.

Unit-III

Integrity and Security: Domain Constraints – Referential Integrity – Assertions – Triggers –Security and Authorization – Authorization in SQL – Encryption and authentication. Relational Database Design: First Normal Form – Second Normal Form – Boyce-Codd Normal Form – Third Normal Form – Fourth Normal Form.

Unit-IV

Storage and File Structures: Overview of Physical Storage Media – Magnetic Disks – RAID – Tertiary Storage – Storage Access – File

Organization – Organization of Records in Files –Data-Dictionary Storage. Indexing and Hashing: Basic Concepts – Ordered Indices – B+-Tree Index Files – B-Tree Index Files – Static Hashing – Dynamic Hashing Index Definition in SQL – Multiple-Key Access. Query Processing: Measures of Query Cost – Selection Operation – Join Operation. Query Optimization: Overview - Estimating Statistics of Expression Results.

Unit-V

Transactions: Transaction concept – Transaction State – Implementation of Atomicity and Durability – Concurrent Executions – Serializability – Recoverability – Implementation of Isolation – Transaction Definition in SQL – Testing for Serializability Concurrency Control: Lock-Based Protocols – Timestamp-Based protocols – Validation-Based Protocols – Multiple Granularity – Deadlock Handling- Weak Levels of Consistency: Degree Two Consistency – Cursor Stability. Recovery System: Failure Classification – Storage Structure – Recovery and Atomicity – Log-Based Recovery – Shadow Paging – Recovery with concurrent Transactions – Buffer Management – Failure with Loss of Non - volatile Storage.

Course outcome

- Define program-data independence, data models for database systems, database schema and database instances.
- Identify Structure Query Language statements used in creation and manipulation of Database.
- Identify the methodology of conceptual modeling through Entity Relationship model.

Text Book

1. A.Silberschatz, N.F. Korth, S. Sudarshan, “Database System Concepts”, 6th Edition – McGraw Hill Higher Education, International Edition 2011.
   (Chapters: 1 to 7, 11, 12, 13, 14, 15 to 17)

Reference

1. Fred R McFadden, Jeffery A Hoffer, Mary B. Prescott, “Modern Database Management”, Fifth Edition, Addison Wesley, 2000.
2. R. Elmasri and S.B. Navathe Benjamin Cummings, “Fundaments of Database Systems”, Redwood City, 1994.
3. B.C. Desai, “An Introduction to Database Systems”, Galgotia Publication, New Delhi, 1995.

Course-18UPCSC1C10 **Credits: 4**

OPERATING SYSTEM

Course objectives:

On successful completion of the course the students should have:

- Understood the Concept operating system principles and Evolution of OS
- Understood the Principles of Process and Threads, Deadlock, processor scheduling and Virtual memory management.
- Learnt case studies in different OS

Unit I:

Introduction to Operating Systems: What is an Operating Systems – **Early History:** The 1940s to 1990s, and 2000 and beyond. **Operating System Components and Goals:** Core Operating System Components – Operating System Goals - Operating System Architectures. **Process:** Introduction – Process States – Process Management – Interrupts – Interprocess Communication. **Case Study:** UNIX Processes.

Unit II:

Asynchronous Concurrent Execution: Introduction - Mutual Exclusion – Implementing Mutual Exclusion Primitives – **Software solutions to the Mutual Exclusion Problem:** Dekker's Algorithm – Peterson's Algorithm. **Hardware solutions to the Mutual Exclusion Problem:** Disabling Interrupts – Test and Set Instruction – Swap Instruction. **Semaphores:** Mutual Exclusion with Semaphore – Thread Synchronization with Semaphore – Counting Semaphore – Implementing Semaphores. **Concurrent Programming:** Introduction – **Monitors:** Condition Variables – Simple Resource Allocation with Monitors – **Monitor Examples:** Circular Buffer - Readers and Writers - Java Monitors.

Unit III:

Deadlock and indefinite Postponement: Introduction – **Examples of Deadlock:** Traffic Deadlock – Simple Resource Deadlock – Deadlock in Spooling Systems – Dinning Philosophers - Resource Concepts – Four Necessary Conditions for Deadlock – Deadlock Solutions – Deadlock Prevention. **Deadlock Avoidance with Dijkstra's Banker's Algorithm:** Example of a Safe and unsafe State - Example of a State Safe to unsafe State Transition – Banker's Algorithm Resource Allocation – Weakness in the Banker's Algorithm - Deadlock Detection – Deadlock Recovery. **Process Scheduling:** Introduction – Scheduling levels – Preemptive V_s Non preemptive Scheduling – Priorities – Scheduling Objectives – Scheduling Criteria – Scheduling Algorithms.

Unit IV:

Physical and Virtual Memory: Introduction – Memory Organization – Memory Management – Memory Hierarchy – Memory Management Strategies – Contiguous V$_s$ Noncontiguous Memory Allocation. **Single-User Contiguous Memory Allocation:** Protection in a Single User System – Single Stream Batch Processing - Fixed Partition Multiprogramming – Variable Partition Multiprogramming – Multiprogramming with Memory Swapping. **Virtual Memory Organization:** Introduction – Virtual Memory Basics – Block Mapping - Paging – Segmentation – Segmentation / Paging Systems.

Unit V:

Virtual Memory Management: Introduction – Demand Paging - Anticipatory Paging – Page Replacement – Page Replacement Strategies – Page Fault Frequency (PFF) Page Replacement – Page Release – Page Size – Program Bahavior under Paging – Global V$_s$ Local Page Replacement. **Disk Performance Optimization:** Introduction – Evolution of Secondary Storage – Characteristics of Moving Head Disk Storage – Why Disk Disk Scheduling Is Necessary – Disk Scheduling Strategies – Rotational Optimization.

Course outcome

- Master understanding of design issues associated with operating systems.
- Master various process management concepts including scheduling, virtual memory, deadlocks
- Be familiar with Threading.
- Be familiar with various types of operating systems including Unix

Text Book:

1. Harvey M. Deitel, Paul J.Deitel and David R. Choffness, **“Operating System”** , Pearson, 2013. (Third Edition)

   Unit 1: Part 1(1) (1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.8, 1.9, 1.12, 1.13)
   Part 2(3) (3.1 – 3.6)
   Unit 2: Part 2(5): (5.1, 5.2, 5.3, 5.4 (5.4.1, 5.4.2), 5.5, 5.6) Part 2(6): (6.1-6.3)
   Unit 3: Part 2(7):(7.1, 7.2, 7.4-7.10) Part 2(8): (8.1)
   Unit 4: Part 3(9): (9.1 – 9.10) Part 3(10): (10.1-10.6)
   Unit 5: Part 3(11): (11.1, 11.3-11.6, 11.8-11.13) Part 4 (12): (12.1-12.6)

Reference Books

1. H. M. Deitel, **"Operating Systems",** Pearson, 1990. (Second Edition)
2. Andrew S. Tanenbaum, **"Modern Operating Systems"**, PHI Learning Private Limited, 2013. (Third Edition)
3. Prdeep K. Sinha, **"Distributed Operating Systems Concepts and Design",** PHI Learning Private Limited, 2014.

Course-18UPCSC1C11 **Credits 4**

SOFTWARE ENGINEERING

Course objective:

- To learn about steps in software development lifecycle and various models.
- To know about the process models and their types.
- To learn about the Design engineering concepts, testing strategies and project management.

Unit-I

Software and Software Engineering:–The Nature of Software–Software–The Unique Nature of Web Apps- Software Engineering –The Software Process- Software Engineering Practice- Software Myths- PROCESS MODELS:– A Generic Process model- Process- Process Assessment And Improvement- Prescriptive Process Models- Specialized Process Models-the Unified Process- Personal And Team Process Models- Process Technology-AGILE DEVELOPMENT- What Is Agility?- Agility And The Cost Of Change- What Is An Agile Process?- Extreme Programming (XP)- Other Agile Process Models. *Case Study*: Add two additional myths to the list. Also state the reality that accompanies the myth.

Unit-II

Principles That Guide Practice:- Software Engineering Knowledge- Core Principles- Principles That Guide Each Framework Activity- UNDERSTANDING REQUIREMENTS:- Requirements Engineering- Establishing the Groundwork- Eliciting Requirements- Developing Use Cases- Building the Requirements Model- Negotiating Requirements- Validating Requirements-Requirements Modeling: Scenarios, Information, and Analysis Classes:- Requirements Analysis- Scenario-Based Modeling- UML Models That Supplement the Use Case- Data Modeling Concepts- Class-Based Modeling-Requirements Modeling: Flow, Behavior, Patterns, And Web apps:- Requirements Modeling Strategies- Flow-Oriented Modeling- Creating a Behavioral Model- Patterns for Requirements Modeling- Requirements Modeling for Web Apps. *Case Study:* Discuss some of the problems that occur when requirements must be elicited from three or four different customers.

Unit-III

Design Concepts:– Design within the Context of Software Engineering- The Design Process- Design Concepts- The Design Model- ARCHITECTURAL DESIGN:- Software Architecture- Architectural Genres- Architectural Styles- Architectural Design- Assessing Alternative Architectural Designs- Architectural Mapping Using Data Flow-

Component-Level Design:- What Is a Component?- Designing Class-Based Components- Conducting Component-Level Design- Component-Level Design for Web Apps- Designing Traditional Components- Component-Based Development-User Interface Design:- The Golden Rules- Interface Design Steps- Web App Interface Design- Pattern-Based Design:- Design Patterns- Pattern-Based Software Design- Webapp Design:- Web App Design Quality- Design Goals- A Design Pyramid for Web Apps- Web App Interface Design.

Case Study: Consider the content object **Order,** generated once a user of **SafeHomeAssured.com** has completed the selection of all components and is ready to finalize his purchase. Develop aUML description for **Order** along with all appropriate design representations.

Unit-IV

Quality Management:– Quality Concepts:- Software Quality- The Software Quality Dilemma- Achieving Software Quality- Review Techniques:- Cost Impact of Software Defects- Defect Amplification and Removal- Review Metrics and Their Use- Informal Reviews- Formal Technical Reviews-Software Quality Assurance:- Elements of Software Quality Assurance-SQA Tasks, Goals, and Metrics- Formal Approaches to SQA- Statistical Software Quality Assurance- Software Reliability- Software Testing Strategies- A Strategic Approach to Software Testing- Strategic Issues-Test Strategies for Conventional Software- Test Strategies for Object-Oriented Software- Test Strategies for Web Apps- Validation Testing-System Testing- The Art of Debugging. *Case Study*: Add two additional questions to each of Garvin's quality dimensions.

Unit-V

Product Metrics: – A Framework for Product Metrics- Metrics for the Requirements Model- Metrics for the Design Model- Design Metrics for Web Apps- Metrics for Source Code- Metrics for Testing- Metrics for Maintenance- Project Scheduling: - Scheduling- Risk Management: - Software Risks- Risk Identification- Risk Projection- EMERGING Trends In Software Engineering:- Technology Evolution- Observing Software Engineering Trends- Identifying "Soft Trends"- Technology Directions-Tools-Related Trends. *Case Study*: Select an open-source development effort (other than Linux), and present a brief history of its evolution and relative success.

Course Outcome

On successfully completing the syllabus students will,

- Understand common software engineering models.

- Have an ability to identify, formulate and solve software engineering problems.
- Develop an ability to analyze, design, verify, validate, implement, apply and maintain software systems.
- Familiarize with Software Design concepts.
- Understand the importance of Software Quality Management and Product metrics.

Text Book:

1. Roger S.Pressman, -Software Engineering a Practitioner‘s Approach, Seventh Edition, McGraw - Hill Higher Education, 2014. Chapters: 1-17, 23, 27, 28, 31. Case Study: - Unit I: Problem 1.11, Unit II: Problem 5.3, Unit III: Problem 13.6, Unit IV: Problem 14.4 Unit V: Problem 31.7.

Reference

1. IanSomerville, -Software Engineering, Seventh Edition, Pearson Education, 2005.
2. Richard Fairly,-Software Engineering Concepts‖, TMGH, 2004.
3. Roger S.Pressman,-Software Engineering a Practitioner‘s Approach, Sixth Edition, McGraw - Hill Higher Education, 2006.
4. Rajib Mall, -Fundamentals of Software Engineering, PHI, Second Edition, 2000.
5. Carlo Ghezzi, Mehdi Jazayeri, Dino Mndrioli,-Fundamentals of Software Engineering, Second Edition, PHI/Pearson Education Asia, 2000

Course18UPCSC1C12 Credits 4

COMPUTER NETWORKS

Objective

- Understand the fundamental concepts of Computer network
- Enable to understand the modern network architecture, protocols and applications
- Exposed to learn the emerging technologies and their potential impact.

Unit- I

Introduction – Network Hardware – Network Software – Reference models: OSI Reference model – TCP/IP Reference model – Network Standardization Physical Layer: Basics of Data Communication – Transmission media – Wireless transmission – Communication Satellites Digital Modulation and Multiplexing – PSTN – Switching.

Unit- II

Data Link layer: Design issues – Error Detection and Correction – Elementary Protocol – Sliding Window Protocol – MAC sub layer: Channel Allocation problem – Multiple Access Protocols – Ethernet – Wireless LAN – Bluetooth – Case Study: Calculation of CRC.

**Unit-
III**

Network Layer: Design Issues – Routing algorithms – Congestion control algorithm - Quality of Service – Internetworking – Network Layer in the Internet – Case Study: Router forwarding.

Unit-IV

Transport Layer: Transport Service – Elements of Transport Protocol – Congestion control - Internet Transport Protocols: UDP – TCP – Delay Tolerant Networking (DTN) – Case Study: Simple Protocol Program for FSM.

Unit- V

Application Layer: DNS – Electronic mail – WWW – Streaming Audio and Video – Content delivery – Cryptography and Network Security: Introduction – Confidentiality – Case Study: Program for RSA Cryptosystem.

Course Outcome

On successful completion of the syllabus students will learn to,

- Describe and analyze the hardware, software components of a network and the interrelations.
- Master the terminology and concepts of the OSI reference model and the TCP-IP reference model.
- Develop solutions for networking and security problems.
- Be familiar with WWW and Cryptography concepts.

Text Book

1. Andrew S Tanenbaum, and David J Wetherall, -Computer Networks, Fifth Edition,
   Pearson Education, 2014. Unit – I (Chapter: 1, 2), Unit – II (Chapter: 3, 4(4.1-4.6)), Unit – III (Chapter: 5), Unit – IV (Chapter: 6), Unit – V (Chapter: 7, 8)
2. B.A. Forouzan, -Data Communication and Networking, Fifth Edition, McGraw Hill, 2013. Unit – V (Chapter: 31),
   Case Study: (Page. No: 292 - Prg.10-1), (Page. No: 560 - Prg.18-4), (Page. No: 733 - Prg.23-1), (Page. No: 1122 - Prg.31-3).

Reference

1. B.A. Forouzan and Firouz Mosharraf -Computer Networks – A Top – down approach, McGraw Hill, 2012.

Course-18UPCSC1C13 **Credits: 2**

OPERATING SYSTEMS LAB

Course objective

- To understand the design aspects of operating system.
- To study the process management concepts.
- To know storage management concepts

Operating System Concepts:

1. Implementation of Process Creation
2. Implementation of Message Communication
3. Implementation of Round Robin Scheduling algorithm
4. Implementation of FCFS scheduling algorithm
5. Implementation of Shared Memory Client/ Server techniques
6. Implementation of Priority Scheduling algorithm
7. Implementation of Paging algorithm
8. Implementation of Semaphore (Producer _Consumer Process)

SHELL SCRIPT

1. Implementation of File status test command.
2. Implementation of Student Grading Process
3. Implementation of Menu driven Program
4. Implement the Menu driven shell program to perform the following :
   i. Enter the sentence in file.
   ii. Search a whole word in an existing file.
   iii. Quit.
5. Develop shell program using 3 arguments to take the pattern as well as input and output file names. If the pattern is found display “Pattern found”, else display “Error message”, also check if right number of arguments are entered.

Course outcome

- Understand and implement basic services and functionalities of the operating system using system calls.
- Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin.
- Implement memory management schemes
- Understand the differences between segmented and paged memories, and be able to describe the advantages and disadvantages of each

Course-18UPCSC1C14 **Credits: 2**

RDBMS LAB

Course objectives

The major objective of this lab is to provide a strong formal foundation in database concepts, technology and practice to the Students to groom them into well-informed database application developers.

- To give a good formal foundation on the relational model of data
- to practice SQL and procedural interfaces to SQL comprehensively
- to understand techniques relating to query processing by SQL engines

List of Exercises

RDBMS– LAB

1. Design a calculator.
2. Simple program using menu Design.
3. Simple program using Timer Control.
4. Simple programs with classes and objects
5. Preparation of student mark list
6. Railway Reservation system
7. Handling Events
8. Creation of Active X controls
9. Simple Animations

Oracle

1. Simple Queries using DDL, DML, base Tables.
2. Simple Queries using DDL, DML base Views.
3. Column Reports Creation.
4. PL/SQL Procedures Creation.
5. Database Triggers Creation.
6. Reports Creation.
7. Built-in Functions.

Course outcome

- Populate and query a database using SQL DML/DDL commands.
- Declare and enforce integrity constraints on a database using RDBMS.
- Programming PL/SQL including stored procedures and cursors

Course-18UPCSC1C15 **Credits: 1**

FINANCIAL COMPUTING-LAB

Course objective

- To prepare and maintain the ledger
- To prepare the estimate for various business activities like credit & debit notes, purchase, sales, receipt and delivery notes.
- To create balance sheet

List of Programs

1. Create a Single Ledger using display and alter commends
2. Create a Multiple Ledger using display and alter commends
3. Create Single group using display and alter commends
4. Create a multiple groups using display and alter commends
5. Create a voucher types like contra, credit & debit notes, purchase, sales, receipt and delivery notes.
6. Create a single stock group.
7. Create a multiple stock groups
8. Create balance sheet for various items
9. Create profit and loss account for various aspects
10. Prepare various types of vouchers like accounting, inventory, and import transaction

Course outcome

- Understand the balance sheet preparation and do analysis