VTU Centre for Online Education (VTU-COE)

MCA in Cyber Security and Cloud Computing

Scheme and Syllabus

Program Outcomes		
Sl No	**Description**	**POs**
1	Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements..	**PO1**
2	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.	**PO2**
3	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	**PO3**
4	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	**PO4**
5	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.	**PO5**
6	Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices.	**PO6**
7	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.	**PO7**
8	Demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	**PO8**
9	Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.	**PO9**

10	Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant top professional computing practices.	**PO10**
11	Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.	**PO11**
12	Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.	**PO12**

Program Education Objectives (PEOs):

PEO 1: promote from current position to software architecture / administration.

PEO2: develop products using automation

PEO 3: demonstrate high moral professional ethics

PEO 4: exhibit lifelong adoption for change in technology.

SEMESTER-I

Sl. No	Code	Course Name	Type of the Course	Credits
1	OMCC101	Mathematical Foundation for Computer Application	CORE	4
2	OMCC102	Operating System	CORE	4
3	OMCC103	Database Management System	CORE / SKILL	4
4	OMCC104	Programming Using C	SKILL	4
6	OMCC105	C Programming Lab	SKILL	2
7	OMCC106	Database Management Lab	SKILL	2
Total Credits			**20**	

SEMESTER-II

Sl. No	Code	Course Name	Type of the Course	Credits
1	OMCC101	Data Structure and Algorithms	CORE	4
2	OMCC102	Object Oriented Programming Using Python	CORE / SKILL	4
3	OMCC103	Software Engineering with Agile Methodologies	CORE / SKILL	4
4	OMCC104	Fundamentals Of Artificial Intelligence and Machine Learning	SKILL	4
6	OMCC105	Data Structure Lab	SKILL	2
7	OMCC106	Python Lab	SKILL	2
Total Credits			**20**	

Mathematical Foundation for Computer Applications		Semester	I
Course Code	OMCC101	CIE Marks	30
		SEE Marks	70
Credits	4	Total Marks	100
		Exam Hours	3
Examination nature (SEE)	Theory		

Course objectives:

- The Curriculum supports the prerequisites to enhance their Mathematical knowledge towards their understanding mathematical Concepts in the concerned fields.

MODULE-1

MATRICES Definition, Types of Matrices, Addition, Subtraction, Scalar Multiplication and Multiplication of Matrices, Adjoint, Inverse, Eigen values and Eigen Vectors of a Matrix, Caley-Hamilton Theorem (Statement only) Rank of a matrix, Row reduced echelon form and normal form Solution of homogeneous and non homogeneous system of linear equations.

MODULE-2

SETS: Sets, Subsets, Types of Sets, Operation on Sets, Cartesian product, Cardinality of sets and applications.

MODULE-3

RELATIONS AND FUNCTIONS:

RELATIONS: Definition with illustrations, Representation of relations to Zero-one matrix and digraphs.

FUNCTIONS: Definition, Domain and Range of function, Types of functions with illustrations.

MODULE-4

Random variable and probability distribution:

Concept of random variable, discrete probability distributions, continuous probability distributions, Mean, variance and co-variance and co-variance of random variables. Binomial and normal distribution, Exponential and normal distribution with mean and variables and problems

MODULE-5
Graph Theory: Graphs and Graphs models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Colouring
Course Outcomes: **CO1**: Develop basic knowledge of matrices and to solve system of linear equations. CO2: Understand the basic concepts of sets, functions and relations.. **CO3**: Understand the concepts of representations of relations and functions.. **CO4**: Model the given problem by applying the concepts of graph theory.. **CO5:** Design strategy using gaming theory concepts for the given problem.
Suggested Learning Resources: **Text Books & Reference Books:** 1. Discrete Mathematics by Guru Raja Chaar. 2. B.S.Grewal: Higher Engineering Mathematics Khanna Publishers, 43rd Edition. 3. Richard A Johnson and C.B Gupta "Probability and statistics for engineers" Pearson Education 4. Kenneth H Rosen, "Discrete Mathematics and its Applications", McGraw Hill publications, 7th edition.

OPERATING SYSTEMS		Semester	I
Course Code:	**OMCC102**	**CIE+SEE Marks**	**30 +70=100**
Credits	**03**	**Exam Hours**	**03**
Examination type (SEE)		**Theory**	

Course Objectives:

CLO 1. Explore the need for OS and different types of OS

CLO 2. Explain the different techniques for management of resources

CLO 3. Learn the Use of processor, memory, storage and file system commands

Module-1

Introduction to operating systems [OS]: What operating systems do; Computer System organization; Computer System architecture; Operating System operations; Resource Management; Security and Protection; Virtualization; Distributed system; Computing environments.

Operating System Structures: Operating Services; User and Operating System interface; System calls; System Services; System programs; Operating system design and implementation; Operating System structure; System Building and Booting; Why Applications Are Operating-System Specific?

Process Management: Process concept; Process scheduling; Operations on processes; Inter process communication [IPC]; IPC Systems

Textbook 1: Chapter – 1, 2 and 3

Module-2

Threads and Concurrency: Multicore Programming, Multithreading models; Thread Libraries; Implicit Threading; Threading issues; OS-Threading examples.

CPU Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling; Multi-Processor Scheduling, Real-Time CPU Scheduling; OS CPU scheduling examples and Algorithm Evaluation

Process Synchronization: Background; The critical section problem; Peterson's solution; Hardware Support for Synchronization; Mutex Locks; Semaphores; Monitors; Classical problems of synchronization.

Textbook 1: Chapter – 4, 5, 6 and 7

Module-3
Deadlocks: System model; Deadlock in Multithreaded Applications; Deadlock characterization; Methods for handling deadlocks; Deadlock Prevention; Deadlock Avoidance; Deadlock detection and Recovery from Deadlock. **Memory Management:** Background; Contiguous memory allocation; Paging; Structure of page table; swapping; Example: Intel 32- and 64-bit Architectures. **Virtual Memory Management:** Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **Textbook 1: Chapter – 7, 8, 9and 10**
Module-4
Mass-Storage Structure: Overview of Mass-Storage Structure; HDD Scheduling; NVM Scheduling; Storage Device Management; Swap-Space Management; Storage Attachment; RAID Structure. **File System:** File concept; Access methods; Directory structure; Protection; File system structure; File system operation; Directory implementation; Allocation methods; Free space management. File system mounting; File sharing. **Textbook 1: Chapter – 11, 12 and 13**
Module-5
Protection: Goals of protection, Principles of protection, Protection Rings; Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. **Case Study: The Linux Operating System:** Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output; Inter-process communication; Network Structure; Security **Textbook 1: Chapter – 17 and 20**

Course Outcomes (Course Skill Set)

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

CO 1. Identify the structure of an operating system and its scheduling mechanism.

CO 2. Demonstrate the allocation of resources for a process using scheduling algorithm.

CO 3. Interpret the root causes of deadlock and provide the solution for deadlock elimination

CO 4. Illustrate different memory management concepts and storage structures such as files , directories and functionalities provided in the Linux Operating system.

Suggested Learning Resources:

Textbooks & Reference Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 10th edition, Wiley-India, 2018

2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw-Hill, 2013.

3. William Stallings Operating Systems: Internals and Design Principles, 9th Edition, Pearson.

4. Andrew S.Tanenbaum, "Modern operating Systems", fourth Edition, PHI Learning Pvt.Ltd., 2008

Database Management System		Semester	I
Course Code	OMCC103	CIE +SEE Marks	30 +70 =100
Credits	05	Exam Hours	3
Examination type (SEE)		Theory	

Course Objectives:

CLO 1. Practice SQL programming through a variety of database problems.
CLO 2. Explore the use of concurrency and transactions in database.
CLO 3. Build database applications for real world problems.

MODULE 1

Introduction: Characteristics of Database approach, Actors on the Scene, Workers behind the scene, Advantages of using DBMS approach, Data models, schemas and instances, Three -schema architecture and data independence, Database languages and interfaces, the database system environment, Centralized and client -server architectures, Classification of Database Management systems, Entity-Relationship Model: Conceptual Database using high level conceptual data models for Database Design, A Sample Database Application, Entity types, Entity sets Attributes and Keys Relationship types, Relationship Sets, Roles and Structural Constraints Weak Entity Types.

MODULE 2

Relational Model Relational Model and Relational Algebra: Relational Model Concepts, Relational Model Constraints and Relational Database Schema Update Operations, Transactions and Dealing with Constraint violations, Unary Relational operations, Relational Algebra Operations from Set Theory, Binary Relational Operations, JOIN and DIVISION, Additional Relational Operations, Examples of Queries in Relational Algebra Relational Database Design Using ER-to Relational Mapping .

MODULE 3

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic structure of SQL Queries, Additional Basic Operations, Null values, Aggregate Functions, nested Sub queries, Modification of the Database, Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Authorization.

MODULE 4
Database Design: Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms based on Primary Keys, General Definitions of 2nd and 3rd Normal Forms, Boyce Codd Normal Forms, Stored Procedures and functions, Triggers.
MODULE 5
Transaction Management: Transaction Concept, A Simple Transaction Model, Transaction Atomicity and Durability, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels. Concurrency Control: Lock Based Protocols, Deadlock Handling. Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm.
Course outcome (Course Skill Set): At the end of the course the student will be able to : **CO1:** Figure out the concepts of database objects, enforce integrity constraints on a database using RDBMS. **CO2**: Demonstrate Structured Query Language (SQL) for database manipulation and also the basic of query evaluation. **CO3**: Develop application to interact with databases, relational algebra expression, **CO4**: Construct an application using tuple and domain relation expression from queries. **Recommended Text and Reference Books:** **Text Books:** 1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson. 2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill. 3. Abraham Silberschatz, Henry F. Korth and S. Sudarshan‟s Database System Concepts 9th EditionTata Mcgraw Hill Education Private Limited-2013 4. Introduction to Database Management System ,Satinder bal Gupta,Aditiya Mittal,2nd Edition,An imprint of Laxmi publications Private Limited-2017

Programming Using C		Semester	I
Course Code	OMCC104	CIE + SEE Marks	30 + 70 = 100
Credits	3	Exam Hours	03
Examination type (SEE)	Theory		

Course objectives:

CLO1 : Explain user-defined data structures like arrays, structures,/unions and pointers in implementing solutions to problems
CLO2: Design and Develop Solutions to problems using modular programming constructs such as functions and procedures.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Lecturer method (L) need not be only a traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
2. Use of Video/Animation to explain the functioning of various concepts.
3. Encourage collaborative (Group Learning) Learning in the class.
4. Ask at least three HOT (Higher-order Thinking) questions in the class, which promotes critical thinking.
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyse information rather than simply recall it.
6. Introduce Topics in manifold representations.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.

Module-1

Introduction to C: Structure of C Program, Compiling and executing C programs, Variables, Constants, IO statements in C, Operators in C

Module-2

Decision control and Looping statements: Introduction to decision control, Conditional branching statements, iterative statements, nested loops, break and continue statements, goto statement.

Module-3
Arrays: Declaration of arrays, accessing and storing of values in array, Operations on arrays, 2-D arrays, operations on two-dimensional arrays, multidimensional arrays, applications of arrays **Functions**: Introduction using functions, Function definition & declaration, function call, return statement, passing parameters to functions, Passing arrays to functions, scope of variables, storage classes, recursive functions.
Module-4
Strings: Introduction to strings, operations on strings, arrays of strings. **Pointers**: Introduction to pointers, declaring pointer variables, Types of pointers, Passing arguments to functions using pointers.
Module-5
Structure and Union: Introduction, structures and functions, Unions, unions inside structures. **Files**: Introduction to files, Operation of Files.
Course outcome (Course Skill Set) At the end of the course the student will be able to: **CO1**: Illustrate the fundamental programming constructs of C programming language to solve problem. **CO2**: Interpret the Use of functions and arrays in implementing solutions. **CO3**: Demonstrate the use of structures, unions and pointers to solve problems.
Suggested Learning Resources: Books 1. Computer Fundamentals and Programming in C – Reema Thareja, 2nd Edition, Oxford University, 2017. 2. E. Balaguruswamy, Programming in ANSI C, 7th Edition, Tata McGraw-Hill 3. Brian W. Kernighan and Dennis M. Ritchie, The 'C' Programming Language, Prentice Hall of IndiaYashavanth Kanetkar, Let us C, Authentic Guide to C Programming Langauge, bpb publisher, 17th Edition, 2020 4. Yashavanth Kanetkar, Let us C, Authentic Guide to C Programming Langauge, bpb publisher, 17th Edition, 2020
Web links and Video Lectures (e-Resources):
• elearning.vtu.ac.in/econtent/courses/video/BS/15PCD23.html • https://nptel.ac.in/courses/106/105/106105171/ MOOC courses can be adopted for more clarity in und

C Programming Lab		Semester	I
Course Code	**OMCC105**	**CIE + SEE Marks**	**30 +70 =100**
Credits	**02**	**Exam Hours**	**03**
Examination type (SEE)	**Practical**		

Course objectives:

CLO1 : Exploring an programs using constructs of C programming language
CLO2: Demonstrate the use of IDE, C Compiler, and identify and rectify the syntax and syntactic errors during programming.
CLO3: Learn to Reporting the observations and debug the program.

Laboratory Experiments:

Implement the following programs with WINDOWS / LINUX platform using appropriate C compiler

1. Implement a C program that takes three coefficients (a, b, and c) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots, output the roots with appropriate messages.
2. Write a C program to simulate a Simple Calculator using Switch case construct.
3. Develop a C Program to check whether a given number is PALINDROME or NOT. Ex: Num: 1221, Reverse: 1221, It is a Palindrome
4. Design and develop a C program to read a year as an input and find whether it is leap year or not.
5. Develop a C Program to search a Name in a list of names using Binary searching Technique (Use strcmp built-in function).
6. Write a C program that reads N integer numbers and arrange them in ascending order using Bubble Sort.
7. Develop, implement and execute a C program that reads two matrices A (m x n) and B (p x q) and Compute product of matrices A and B. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.
8. Design and develop a C function isprime(num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given range.
9. write a recursive C function to find the factorial of a number, n!, defined by fact(n)=1, if n=0. Otherwise fact(n)=n*fact(n-1). Using this function, write a C program to compute the binomial coefficient nCr . Tabulate the results for different values of n and r with suitable messages.

10. Write a C program to copy the contents of one file to another.

11. Write a C program that uses functions and structures to perform the following operations:

- **a.** Reading a complex number
- **b.** Displaying a complex number
- **c.** Addition of two complex numbers
- **d.** Multiplication of two complex numbers

Display the appropriate output.

12. Write a Program in c to swap two number using pointer.

Course outcome (Course Skill Set)

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

CO1: Make use of IDE for programming, identify and correct the syntax and syntactic errors using various programming constructs.

CO2: Demonstrate use of functions, recursive functions, arrays, strings, structures and pointers in problem solving.

CO3: Design and development of C programs to implement different searching and sorting techniques.

Suggested Learning Resources:

Books

1. Computer Fundamentals and Programming in C – Reema Thareja, 2nd Edition, Oxford Univeristy, 2017
2. E. Balaguruswamy, Programming in ANSI C, 7th Edition, Tata McGraw-Hill
3. Brian W. Kernighan and Dennis M. Ritchie, The 'C' Programming Language, Prentice Hall of India
4. Yashavanth Kanetkar, Let us C, Authentic Guide to C Programming Langauge, bpb publisher, 17th Edition, 2020

Web links and Video Lectures (e-Resources):

- elearning.vtu.ac.in/econtent/courses/video/BS/14CPL16.html
- https://nptel.ac.in/courses/106/105/106105171/

Database Management Lab		Semester	I
Course Code	OMCC106	CIE + SIE Marks	30 +70 = 100
Credits	02	Exam Hours	03
Examination type (SEE)	Practical		

Course objectives:

CLO1: Create SQL queries for the small projects.

CLO2: Create database objects that include tables, constraints, indexes, and sequences.

1. Students should be allowed to choose appropriate DBMS software, install it, configure it and start working on it. Create sample tables, execute some queries, use SQLPLUS features, Use PL/SQL features like cursors on sample database. Students should be permitted to practice appropriate User interface creation tool and Report generation tool.

2. A college consists of number of employees working in different departments. In this context, create two tables' employee and department. Employee consists of columns empno, empname, basic, hra, da, deductions, gross, net, date-of-birth. The calculation of hra,da are as per the rules of the college. 1. Create tables department and employee with required constraints. 2. Initially only the few columns (essential) are to be added. Add the remaining columns separately by using appropriate SQL command 3. Basic column should not be null 4. Add constraint that basic should not be less than 5000. 5. Calculate hra,da,gross and net by using PL/SQL program.

3. Students may be divided into batches and the following experiments may be given to them to better understand the DBMS concepts. Students should gather the required information, draw ER diagrams, map them to tables, normalize, create tables, triggers, procedures, execute queries, create user interfaces, and generate reports.

- Student information system
- KSRTC reservation system
- Hostel management
- Library management
- Indian Railways reservation

Course outcomes (Course Skill Set):

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

CO1: Design entity-relationship diagrams to solve given database applications.

CO2: Implement a database schema for a given problem.

CO3: Formulate SQL queries in Oracle for the given problem.

CO4: Design and Develop suitable database and verify for its appropriate normalization for any given problem.

Data Structures and Algorithms		Semester	II
Course Code	OMCC201	CIE + SIE Marks	30 + 70 =100
Credits	04	Exam Hours	03
Examination type (SEE)	Theory		

Course Learning objectives:

CLO1: Explore step by step and develop algorithms to solve real world problems.

CLO2: Evaluate the Expressions like postfix, prefix conversions.

CLO3: Implementing various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs.

CLO4: Define various searching & sorting techniques.

CLO5: Compare functions using asymptotic analysis and describe the relative merits of worst-, average-, and best-case analysis.

Module-1	
Classification of Data Structures: Primitive and Non- Primitive, Linear and Nonlinear; Data structure Operations, Stack: Definition, Representation, Operations and Applications: Polish and reverse polish expressions, Infix to postfix conversion, evaluation of postfix expression, infix to prefix, postfix to infix conversion.	
Teaching-Learning Process	Chalk and talk method / PowerPoint Presentation
Module-2	
Recursion - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi. Queue: Definition, Representation, Queue Variants: Circular Queue, Priority Queue, Double Ended Queue; Applications of Queues. Programming Examples.	
Teaching-Learning Process	Chalk and talk method / PowerPoint Presentation
Module-3	
Linked List: Limitations of array implementation, Memory Management: Static (Stack) and Dynamic (Heap) Memory Allocation, Memory management functions. Definition, Representation, Operations: getnode() and Freenode() operations, Types: Singly Linked List. Linked list as a data Structure, Inserting and removing nodes from a list, Linked implementations of stacks, Header nodes, Array implementation of lists.	
Teaching-Learning Process	Chalk and talk method / PowerPoint Presentation

Module-4	
Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees- Evaluation of Expression, Programming Examples.	
Teaching-Learning Process	Chalk and talk method / PowerPoint Presentation
Module-5	
Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search. Insertion Sort,. Hash Table organizations, Hashing Functions, Static and Dynamic Hashing.	
Teaching Learning Process	Chalk and talk method / PowerPoint Presentation

Course outcome (Course Skill Set):

At the end of the course the student will be able to:
CO1: Illustrate the different data structures and operations.
CO2: Demonstrate the concept of stack and Queue data structures use
CO3: Infer the concept of Linked list, Trees and Graphs in problem solving
CO 4: Employ various data structures for solving various problems.

Suggested Learning Resources: Reference books:

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014.
3. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012.
4. Introduction to Algorithms ,Thomas h.Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein,4th Edition,2022.MIT Press

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=BBpAmxU_NQo
- https://www.youtube.com/watch?v=8hly31xKli0
- https://archive.nptel.ac.in/courses/106/106/106106127/

Object Oriented Programming Using Python		Semester	II
Course Code	OMCC202	CIE + SIE Marks	30 + 70 =100
Credits	4	Exam Hours	3
Examination type (SEE)	Theory		

Course objectives:

CLO1: Study the importance of Object Oriented Programming
CLO2: Explore the Object Oriented Programming concepts
CLO3: Explain the concept of Polymorphism, Inheritance
CLO4: Understand the creation of modules ,packages and organization of modules and packages

Teaching-Learning Process (General Instructions)

Programming Exercises and mini project works.

Module-1

Python Basic Concepts and Programming

Parts of Python Programming Language, Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, The type( ) Function and Is Operator, Control Flow Statements, The if Decision Control Flow Statement, The if…else Decision Control Flow Statement, The if…elif…else Decision Control Statement, Nested if Statement, The while Loop, The for Loop, The continue and break Statements, Built-In Functions, Commonly Used Modules, Function Definition and Calling the Function, The return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments.

Module-2

Python Collection Objects, Strings- Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings, Lists-Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods. Sets, Tuples and Dictionaries. Files: reading and writing files

Module-3
Object-oriented Design :Introducing object-oriented ,Objects and classes, Specifying attributes and behaviours : Data describes objects , Behaviours are actions ,Hiding details and creating the public interface: Composition, Inheritance: Inheritance provides abstraction, Multiple inheritance Objects in Python : Creating Python classes , Adding attributes, Making a function work: passing arguments, Initializing the object, self argument
Module-4
Modules and packages: Organizing the modules, Absolute imports, Relative imports, Organizing module contents: Access control, Third-party libraries, Basic inheritance, Extending built-ins, Overriding and super, Multiple inheritance, The diamond problem, Different sets of arguments,
Module-5
Polymorphism , Abstract base classes, Using an abstract base class , Creating an abstract base class Exceptions: Raising exceptions, The effects of an exception , Handling exceptions, The exception hierarchy, Defining our own exceptions
Course outcome (Course Skill Set) At the end of the course the student will be able to: **CO1**: Demonstrate proficiency in handling loops and creation of functions **CO2**: Illustrate the methods to create and manipulate lists, tuples and dictionaries . CO3: Design and Develop programs for string processing and file organization. **CO4**: Interpret the concept of OOP as used in Python
Suggested Learning Resources: **Books** 1. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist'', 2nd edition,Updated for Python 3, Shroff/O'Reilly Publishers, 2016 2. Python 3 Object Oriented Programming, 2nd Edition, Unleash the power of Python 3 Objects by Dusty Phillips , PACKT Publishing. 3. Python Object–Oriented Programming :Build robust and maintainable Object-oriented python applications and libraries, Steven F. Lott, Dusty Philips,4th Edition, Packt Publishing Limited; 2021 4. Python the complete reference ,Martin C. Brown,4th Edition, McGraw Hill Education ,2018
Web links and Video Lectures (e-Resources):
• http://greenteapress.com/wp/thinkpython/

Software Engineering with Agile Methodologies		Semester	II
Course Code	OMCC203	CIE + SIE Marks	30 + 70 = 100
Credits	04	Exam Hours	03
Examination type (SEE)		Theory	

Course Learning objectives:

CLO1: Outline software engineering principles and activities involved in building large software programs.
CLO2: Identify ethical and professional issues and explain why they are of concern to software engineers.
CLO3: Explain the fundamentals of object oriented concepts.
CLO4: Describe the process of requirements gathering, requirements classification, requirements specification and requirements validation.
CLO5: Differentiate system models, use UML diagrams and apply design patterns.
CLO6: Discuss the distinctions between validation testing and defect testing.

MODULE 1

Introduction: Software Products and Software process, Process models: Waterfall modal, Evolutionary Development, Bohemia's Spiral model, Overview of risk management, Process Visibility, Professional responsibility. Computer based System Engineering: Systems and their environment, System Procurement, System Engineering Process, System architecture modelling. Human Factors, System reliability Engineering.

MODULE 2

Requirements and Specification: The requirement Engineering Process, The Software requirement document, Validation of Evolution of requirements, Viewpoint – oriented & method based analysis, system contexts, Social 7 organizational factors . Data flow, Semantic, Objects, models, Requirement Specification, Non functional requirement.

MODULE 3

Software Prototyping: Prototyping in software process, Prototyping techniques, User interface prototyping. Software Design: Design Process, Design Strategies, Design Quality, System Structuring control models, Modular decomposition, Domain Specific architecture.

MODULE 4
Agile Methodology : Theories for Agile Management – Agile Software Development – Traditional Model vs. Agile Model - Classification of Agile Methods – Agile Manifesto and Principles – Agile Project Management – Agile Team Interactions – Ethics in Agile Teams - Agility in Design, Testing – Agile Documentations – Agile Drivers, Capabilities and Values
MODULE 5
Agile Process: Lean Production - SCRUM, Crystal, Feature Driven Development- Adaptive Software Development - Extreme Programming: Method Overview – Lifecycle – Work Products, Roles and Practices.
Course outcome (Course Skill Set) **At the end of the course the student will be able to :** **CO1:** Illustrate a software development process model, components or processes to meet desired needs within realistic constraints. **CO2**: Demonstrate the usage of techniques, skills, and modern engineering tools necessary for software engineering practice. CO3: Showcase the suitable prototyping concept, software design strategies and modular approaches in software design process. **CO5**: Explore the Agile Software Development, project management, Team Interactions etc., **Suggested Learning Resources:** **Reference books:** 1. Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012. 2. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML,2nd Edition, Pearson Education,2005. 3. Roger S. Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill. 4. Stephan R. Schach, "Object oriented software engineering", Tata McGrawHill,2008

Fundamentals of Artificial Intelligence and Machine Learning		Semester	I
Course Code	**OMCC204**	**CIE + SIE Marks**	**30 + 70 = 100**
Credits	**4**	**Exam Hours**	**03**
Examination type (SEE)	**Theory**		

Course Learning objectives:

CLO1. Explain the basic principles of Artificial Intelligence.
CLO2. Learn and design intelligent agents.
CLO3. Explore the basic areas of artificial intelligence including problem solving, knowledge representation, reasoning, decision making, planning, perception and action.
CLO4: Interpret the fundamental concepts of machine learning and its various algorithms.
CLO5: Implement various strategies of generating models from data and evaluating them

Module 1: Introduction to AI and Machine Learning (10 hours)

Introduction to AI and ML (2 hour): Definition of Artificial Intelligence (AI) and Machine Learning (ML), Historical context and key milestones, Applications of AI and ML in real-world scenarios, Types of Machine Learning (4 hour): Supervised learning, unsupervised learning, and reinforcement learning, Examples of applications for each type, Overview of supervised and unsupervised algorithms, Basics of Data and Feature Engineering (2 hours): Importance of data quality and pre-processing, Data representation and feature extraction, Handling missing data and categorical variables
Evaluation Metrics (2 hour): Accuracy, precision, recall, F1-score, ROC curve, AUC

Selecting appropriate metrics for different tasks, Trade-offs between different metrics

Module 2: Supervised Learning (10 hours)

Linear Regression (2 hours): Introduction to linear regression, Simple and multiple linear regressions, Model training, evaluation, and interpretation. Classification Algorithms (5 hours): Logistic regression, Decision trees and random forests, Naive Bayes classifier, Support Vector Machines (2 hours): Introduction to SVM, Linear SVM and kernel trick Hyperparameter tuning and model evaluation

Module 3: Unsupervised Learning	**(10 hours)**
Clustering (4 hours): K-means clustering, Hierarchical clustering, Evaluating cluster quality, Dimensionality Reduction (3 hours): Principal Component Analysis (PCA) t-Distributed Stochastic Neighbor Embedding (t-SNE), Applications and benefits of dimensionality reduction. Anomaly Detection (3 hours): Identifying anomalies in data, Approaches to anomaly detection, Real-world use cases.	
Module 4: Neural Networks and Deep Learning	**(10 hours)**
Introduction to Neural Networks (3 hours): Basics of artificial neurons, Activation functions and network architectures, Feed forward and back propagation, Convolutional Neural Networks (4 hours): Basics of CNNs, Image classification and object detection, Transfer learning and pre-trained models, Recurrent Neural Networks (3 hours) , Understanding RNNs, Applications in sequential data processing Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU).	
Module 5: Practical Applications and Future Trends	**(10 hours)**
Natural Language Processing (4 hours), Introduction to NLP and its challenges Text pre-processing and tokenization, Basics of sentiment analysis and text generation AI Ethics and Bias (3 hour), Ethical considerations in AI and ML, Addressing bias and fairness in algorithms, Responsible AI development, Future Trends in AI and ML (3 hour), Reinforcement learning advancements, Generative Adversarial Networks (GANs), Explainable AI and interpretable models.	
Course outcome (Course Skill Set): **At the end of the course the student will be able to :** **CO1**. Illustrate the foundational principles, mathematical tools and program paradigms of AI and fundamental principles of machine learning. **CO2**. Demonstrate the formal methods of knowledge representation and Formulation of a Machine Learning problem. **CO3**. Showcase the usage of intelligent agents for Artificial Intelligence programming techniques and Develop a model using supervised/unsupervised machine learning algorithms for classification/prediction/clustering.	

CO4. Interpret logic reasoning and problem solving techniques for AI applications

Reference Book

1. AurolienGeron, "Hands-On Machine Learning with Scikit-Learn and TensorFlow, Shroff/O'Reilly",2017
2. Andreas Muller and Sarah Guido, "Introduction to Machine Learning with Python: A Guidefor Data Scientists", Shroff/O'Reilly, 2016
3. Andrew Ng, Machine learning yearning, https://www.deeplearning.ai/machine-learningyearning/
4. Russell, Norvig, Artifificial Intelligence: A Modern Approach, Third edition, Prentice Hall,2010

Data Structures Laboratory		Semester	II
Course Code	OMCC205	CIE + SIE Marks	30 + 70 = 100
Credits	2	Exam Hours	03
Examination type (SEE)	Practical		

Course objectives:
CLO1: Explain the Evaluation of Expressions like postfix, prefix conversions.
CLO2: Implementing various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs.

Sl. No	Experiments
1	Implement a Program in C for converting an Infix Expression to Postfix Expression.
2	Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).
3	Design, develop, and execute a program in C to simulate the working of a queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display
4	Write a C program to simulate the working of a singly linked list providing the following operations: a. Display & Insert b. Delete from the beginning/end c. Delete a given element
5	Write a C program to Implement the following searching techniques a. Linear Search b. Binary Search.
6	Write a C program to implement the following sorting algorithms using user defined functions: a. Bubble sort (Ascending order) b. Selection sort (Descending order).
7	Write a C program to implement the Binary Search Tree operations.
8	Write a C program to demonstrate the Binary Tree Traversals - Inorder, postorder, preorder

Course outcomes (Course Skill Set):

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

CO1: Design and Develop Data structure techniques for evaluating the given expression.
CO2: Demonstrate various sorting / searching techniques and validate input/output for the given problem.
CO3: Design data structures to show the operations on Stacks, Queues, Circular Queues, Linked Lists, and Trees.
CO4: Implement the suitable algorithm to find whether the given graph is connected or not and illustrate the performance of the technique implemented.

Python Lab		Semester	II
Course Code	OMCC206	CIE + SIE Marks	30 + 70 =100
Credits	02	Exam Hours	03
Examination type (SEE)		Practical	

Course Learning Objectives:

CLO1: To be able to introduce core programming basics and program design with functions using Python programming language.

CLO2: To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.

CLO3: To understand the high-performance programs designed to strengthen the practical expertise.

1. Write a program to sum all the elements from n1 to n2 where n1 and n2 are positive integers
2. Input an array of n numbers and find separately the sum of positive numbers and negative numbers.
3. Write a program to search an element using linear search
4. Write a program to search an element using binary search.
5. Write a program to simulate stack.
6. Using a stack evaluate an arithmetic expression.
7. Write a program to multiply two matrices
8. Write a program to find the roots of a quadratic equation
9. Write a program to Insert a number in a sorted array.
10. Write a Python Program to check whether the given string is palindrome or not using built in string manipulation methods.
11. Write a Python Program to read a word and prints the number of letters, vowels and percentage of vowels in the word using dictionary
12. Write a Python Program to check a given sentence is a pangram or not using function/Module.

Course Outcomes: At the end of the course the student will be able to:

CO1: Demonstrate proficiency in handling loops and creation of functions.

CO2: Illustrate the methods to create and manipulate lists, tuples and dictionaries in Python Programme.

CO3: Design and Develop programs for string processing and file organization and use the concept of OOP as used in Python.