FACULTY OF ENGINEERING & TECHNOLOGY

Effective from Academic Batch: 2020-21

Programme: **Bachelor of Engineering (Information Technology)**

Semester: **II**

Course Code: **102000212**

Course Title: **Object Oriented Programming**

Course Group: **Engineering Science Courses**

Course Objectives: The object oriented approach for software development has become the de-facto standard for the industry to develop the product based or customized software based on customer demand. The software libraries developed for various fields also follows the phenomena of object oriented development. The subject covers the basic concepts of the object oriented paradigm and popular object oriented programming language C++. The subject covers the basics of C++, objects and classes, Inheritance, and Polymorphism. The subject introduces the Python Programming Language to harness its potential for modern computing requirements.

Teaching & Examination Scheme:

Contact hours per week			Course Credits	Examination Marks (Maximum / Passing)				
				Theory		J/V/P*		Total
Lecture	Tutorial	Practical		Internal	External	Internal	External	
3	0	2	4	40 / 14	60 / 21	20 / 07	30 / 10	150 / 52

* **J**: Jury; **V**: Viva; **P**: Practical

Detailed Syllabus:

Sr.	Contents	Hours
1	Pointers in C, Dynamic Memory Allocation and File management	02
2	**Concepts of OOP:** Introduction OOP, Procedural Vs. Object Oriented Programming, Principles of OOP, Benefits and applications of OOP	04
3	**C++ Basics:** Overview, Program structure, namespace, identifiers, variables, constants, enum, operators, typecasting, control structures	06
4	**C++ Functions:** Simple functions, Call and Return by reference, Inline functions, Macro Vs. Inline functions, Overloading of functions, default arguments, friend functions	04
5	**Objects and Classes:** Basics of object and class in C++, Private and public members, static data and function members, constructors and their types, destructors, operator overloading, type conversion	06

6	**Inheritance:** Concept of Inheritance, types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members, overriding, virtual base class	**06**
7	**Polymorphism:** Pointers in C++, Pointes and Objects, this pointer, virtual and pure virtual functions, Implementing polymorphism	**03**
8	**I/O, Files and Templates:** C++ stream classes, Unformatted and formatted I/O, manipulators, File management functions, File modes, Templates, Exception handling and Standard Template Library	**03**
9	**Introduction to Python Programming:** The basic elements of python, Branching Programs, Control Structures, Strings and Input, Iteration, Functions and scoping, Specifications, Recursion, Global variables, Strings, Tuples, List	**06**
	TOTAL	**40**

List of Practicals / Tutorials:

1	• Design a simple class with all arithmetic function. Use them in MAIN function. • Create a class Student with student name and age as data members. Define functions to read and display the data members. • Create a String class that includes all the string-related function like length, copy, compare, concatenation, sub string search (without using inbuilt string functions).
2	• Write a program to find the largest of three integers using a swap function. The function accepts integer arguments by reference. • Design classes named Triangle, Square, and Circle. Make the different function in each class to find area of particular shape. • Create a class with string pointer as data member and member functions.
3	• Write a program to create a Constructor to allocate memory dynamically and read value. Create a Display function to display the string. Create a Destructor function to free allocated memory.
4	• Write a function that creates an array of user given size using new operator. • Define a class to represent a bank account. Include the members like name of the depositor, account number, type of account, and balance amount in the account. Make functions; (1) To assign initial values (2) To deposit an amount (3) To withdraw an amount after checking the balance (4) To display name and balance Write a main program to test the program. • Create a program to convert temperature in Fahrenheit to Celsius and display, using class.

5	• Create a class 'DISTANCE' with feet and inches as data members. Create member function to input distance, member function to output distance and member function to add two distance objects. Write a main function to create objects of DISTANCE class. Input two distances and output the sum. • Write a function that creates a vector of user given size M using new operator. Demonstrate the use of the function. • Write a program to swap two numbers by both call by value and call by reference mechanism, using two functions swap_value and swap_reference respectively, by getting the choice from the user and executing the user's choice by switch-case.
6	• Write a program to implement function overloading in order to compute power(m,n) where, (1) m is double and n is int (2) m and n are int. • Create a function called reverse that takes two parameters. The first parameter, called str is a pointer to a string that will be reversed upon return from the function. The second parameter is called count, and it specifies how many characters of str to reverse. Give count a default value that, when present, tells function reverse to reverse the entire string. • Write a program to demonstrate the use of arrays within a class. Create and manage an inventory system.
7	• Create a program to understand and use of static members and static member functions. • Create a class Employee with suitable members and functions. Create an array of objects and demonstrate the use of the class using the main function. • Create a class Time with members hours and minutes. Write a member function 'add' which takes 2 arguments of type class Time and demonstrate the use with a main program.
8	• Create a class Sample with members a and b of type integer. Write a friend function that takes an object as argument and calculates the mean of the two members. • Create a class Complex that has two members of type float. Write a friend function that calculate the sum of the two complex objects and returns the result as an object. Demonstrate the working using a main function. For the Complex class, demonstrate the use of multiple constructors. • Write a program to demonstrate the use of copy constructor.
9	• Construct a two-dimensional array using dynamic constructors. • Write a program to overload the + and – operators for the complex class. • Write a program to overload the unary – operator for a suitable class.
10	• Write a program to overload the + and == operators for the string class. • Write a program to overload the [] operator. • Write a program to overload the << and >> operators. • Write a program to convert a basic type to a class type and vice versa.
11	• Write a program to convert an object of one class to another class. • Design a class Polar which describes a point in the plane using polar coordinates radius and angle. Use overloaded + operator to add two polar objects. • Define two classes Polar and Rectangle to represent points in the polar and rectangular systems. Use conversion routines to convert from one system to the other.

12	• Write a program to implement single inheritance. Show the consequences of deriving a class in public, protected and private manner with a simple example. Consider a simple example. Class Student stores the roll-number, class Test stores the marks in two subjects and class Result contains the total marks obtained in the test. The class Result inherits the details of the marks obtained and roll number of students through multilevel inheritance. Write a program to demonstrate the above. • Extend the above program in to add a Sports class. The Result class inherits the details of marks obtained from class Test and the performance in sports from the Sports class (hybrid inheritance).
13	• Write a program to demonstrate how parameters are passed to the base class constructor via the derived class constructor. • Write a program to use the following functions: Put(), Get(), Getline(), Write(). • Write a program to produce formatted output using the following functions: Width(), Precision(), Fill(), Setf(), Unsetf().
14	• Use of various flags and bit fields to produce formatted output. ○ Write a program to use manipulators setw, setiosflags and setprecision for formatted output. ○ Write a program to read a list containing item name, item code, and cost interactively and produce a three column output as shown below. NAME CODE COST _____ Turbo C++ 1001 250.95 C Primer 905 95.70 ... ... _____
15	• Write a program to create files with constructor function, open function, and using various file mode parameters. • Write a program to use the following functions: Seekg(), Tellg(), Seekp(), Tellp(), Put(), Get(), Write(), Read(). • Basic Python programs.

Reference Books:

1	Object Oriented Programming with C++, by E Balagurusamy, TMH
2	Object Oriented Programming in Turbo C++, by Robert Lafore, Galgotia
3	The Compete Reference C++, by Herbert Schlitz, TMH
4	C++ : How to Program, by Deitel and Deitel, PHI
5	C++ Programming, by Steven Holzner, Dreamtech
6	Introduction to Computation and Programming Using Python by John V Guttag , PHI
7	Core Python Programming by R. Nageswara Rao, dreamtech
8	Core Python Programming - Second Edition by Wesley J. Chun., PHI
9	Fundamentals of Python – First Programs, Kenneth A. Lambert, CENGAGE Publication

Supplementary learning Material:

1	NPTEL courses

Pedagogy:

- Direct classroom teaching
- Assignments/Quiz
- Continuous assessment
- Seminar/Poster Presentation
- Course Projects

Suggested Specification table with Marks (Theory) (Revised Bloom's Taxonomy):

Distribution of Theory Marks in %						**R**: Remembering; **U**: Understanding; **A**: Applying; **N**: Analyzing; **E**: Evaluating; **C**: Creating
R	**U**	**A**	**N**	**E**	**C**	
20%	30%	30%	20%	-	-	

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes (CO):

Sr.	Course Outcome Statements	%weightage
CO-1	Describe concepts of OOP.	**25**
CO-2	Write basic programs in C++ using class, objects, inheritance etc.	**25**
CO-3	Write programs for real life problems using polymorphism, templates etc.	**25**
CO-4	Develop an application using Object Oriented Programming.	**15**
CO-5	Write basic programs in Python.	**10**

Curriculum Revision:	
Version:	1.0
Drafted on (Month-Year):	June-2020
Last Reviewed on (Month-Year):	-
Next Review on (Month-Year):	June-2025