

<b>Subject Code : 1CS2010202</b>	<b>Subject Title: OBJECT ORIENTED PROGRAMMING</b>
<b>Pre-requisite :</b>	Fundamentals of Procedural Programming

**Course Objective:**

The objectives of the course are to:

- Understand the concepts of object oriented programming using C++.
- Learn the concepts starting from the basics like class and objects and extending the knowledge to high-level concepts like file IO which are widely required when developing an entire application system.
- Enable the student to visualize as well as synthesize a real world object oriented scenario and make the student ready for other similar object oriented programming languages.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
4	-	2	6	60	40	30	20	150

Subject Contents			
Sr. No.	Topic	Total Hours	Weight (%)
1	<b>Fundamentals of Object Oriented Programming:</b> Object-Oriented Programming Paradigm, Basic concepts of OOP, Benefits of OOP, Applications of OOP, Concept of input and output operator ,structure of C++, Keywords, Identifiers and Constants, Data Types, variable declaration and initialization, Operators, Control Structure. <b>Functions:</b> Introduction to Function, Function Prototyping, Call by Reference and Return by Reference, Inline Function, Default Argument, Function Overloading.	9	20
2	<b>Classes and Objects:</b> The Difference between structure and class, specify class and member functions, Private and Public member function, Static data member and Static Function, Array of Objects, Object as Function Argument, Friend Function, Returning Object, pointer to members of class. <b>Constructors and Destructors:</b> Introduction to constructors, Parameterized constructors, Constructor Overloading, Constructors with default arguments, Dynamic Initialization of objects, Copy constructors, Dynamic constructor, Destructors	9	20
3	<b>Operator Overloading and Data Conversions:</b> Introduction to Operator Overloading, Unary and Binary Operators, Overloading Binary operators using friends, Data Conversions. <b>Inheritance:</b> Introduction, defining derived class, Derivation using public, private and protected access modifiers, Types of Inheritance, Virtual Base Class, Abstract classes, Constructor in Derived Class, Initialization list in Constructor, Containership.	10	20
4	<b>Virtual Functions, Run time polymorphism:</b> Compile Time and Runtime Polymorphism, Pointers to Objects, This pointer, Compatibility of Derived and base class pointers, Virtual functions, Pure virtual functions. <b>Exception Handling:</b> Introduction, Exception handling mechanism, re-throwing an exception, Specifying exception.	10	20

<b>5</b>	<p><b>IO Streams:</b> Introduction, The C++ Predefined streams, stream classes hierarchy, Formatting IO, Formatting using ios members, Manipulators, User defined manipulators.</p> <p><b>File Management:</b> File I/O programming, Text and binary streams, Opening and closing files, Dealing with text files, Dealing with binary files, Random Access, IO Modes, and Handling Errors.</p>	10	20
----------	--	----	----

**Course Outcome:**

**At the end of the course, the student would be able to:**

- Understand the basic object oriented programming concepts like object, class and Encapsulation.
- Develop the simple C++ programs using the variables, operators, control structures, functions and I/O objects by objects and classes.
- Use OOP concepts like type conversion, inheritance, polymorphism, I/O streams, exceptions and files to develop programs for real life problems.
- Develop the applications using object oriented programming with C++.

**List of References:**

1. Object Oriented Programming with C++ by E. Balagurusamy, TMH
2. Programming with ANSI C++ by Bhushan Trivedi, Oxford University Press
3. Complete Reference C++ by Herbert Schildt, McGraw Hill Publications
4. C++ Programming Language – By Bjarne Stroustrup, Pearson Education
5. Object Oriented programming in C++ by Robert Lafore, Pearson Education

**List of Experiments:**

**Note:** The experiment list provided beneath is for reference only. The course teacher may Change/formulate it as per his/her methodology and requirement.

- 1 Write a program which illustrates the use of scope resolution operator.
- 2 **Practical based on implementing classes.**
  - 2.1 Define a class Address to store the house number, apartment-name, city, state and postal code. Create few objects and set and print the address
  - 2.2 Implement a class Account. An Account class has account holder's name, a balance, functions to add and withdraw money and a function to query current balance. Charge a penalty of Rs. 100 if an attempt is made to withdraw more money than available in the account. Also provide the functionality to withdraw and deposit money in the account.
- 3 Demonstrate the use of static variables in a class by using it to count the number of objects created in the program.
- 4 **Practical based on function & function overloading.**
- 5 Write a program to find out sum of 1 to n number using constructors
- 6 **Practical based on Operator overloading.**
  - 6.1 Write a program to overload the unary minus operator using friend function and member function.
  - 6.2 Write a program to overload the binary plus operator using friend function and member function.
  - 6.3 Write a program for concept of overloaded constructors.
  - 6.4 Write a menu driven program that can perform the following functions on strings. (Use overloaded operators where possible). (Do not use predefined string class )
    1. Compare two strings for equality (== operator)
    2. Check whether first string is smaller than the second (<= operator)
    3. Copy the string to another
    4. Reverse the string

5. Concatenate two strings (+ operator)
- 7 Construct a invent1 class having member variables code, item and prize and invent2 class having member variables code and value. Find out following conversion.
  1. To convert invent1 class into the basic data type int.
  2. To convert a basic data type int to invent1 class.
  3. To convert invent1 to invent2 class to find out total value. (value= item \* prize)
- 8 Practical based on inheritance.
  - 8.1 Create a class Student with the member variables like rollno and name. Create a class Sports which include member variable score. Create another class Test that inherits Student class and includes the member variable marks1, marks2. Each class should include the functions get and put to set and retrieve the member variables. Finally create Result class that inherits test and sports class and contain its member variable total\_marks and function display to show the result of student.  
Write a program to demonstrate the concept of virtual base class.
  - 8.2 Create a class cricketer that holds the name, date-of-birth, no-matches etc., of each cricketer. Include the virtual functions to set and retrieve values. Inherit the class cricketer to create a class batsman that stores the number of 100's, 50's of the batsman and a class bowler that holds the total-wickets, no-maiden-overs of the bowler. Include function to set and retrieve values in both classes.
- 9 Write a program to show the implementation of pointer to objects.
- 10 Write a program to implement multiple catch statement.
- 11 Practical based on console I/O.
  - 11.1 Design a manipulator to provide the following output specifications for printing float values
    1. 15 column width
    2. Right justified
    3. 2 digits precision
    4. Filling unused spaces with +.
    5. Showing trailing zeros.Write a C++ program to demonstrate creation of user defined manipulator
- 12 Practicals based on file handling.
  - 12.1 Write a program of working with single file. (Creates a file with constructor function).
  - 12.2 Write a program of reading from two files simultaneously.
  - 12.3 Write a program that stores and displays the records of the Inventory from a file The information for inventory is name, code and cost to be stored. Create a binary file name "STOCK.DAT" that store the stock information. Perform following operation on file.
    1. Adds a new item to the file.
    2. Modifies the details of an item.
    3. Displays the contents of the file.
  - 12.4 Create two files called ODD and EVEN using the command line arguments. Set of numbers stored in an array are written to these files. The odd numbers are written to the file ODD and the even numbers are written to the file EVEN. The program then displays the contents of the files.
  - 12.5 Write a program that displays a student object read from a file backwards.
  - 12.6 Write a program that reads a text file and creates another file that is identical except that sequence of consecutive blank spaces is replaced by single space.
  - 12.7 Write a program that swaps each character pair in a text file. For example, if the file contains "1234", then after the program in run, the file will contain "2143".