

Subject Code: 1ET4000302	Subject Title: Object Oriented Programming with C++
Pre-Requisite	--

COURSE OBJECTIVE:

This course intends to teach the students about basic concepts of Object-Oriented Programming (OOP) and C++. Large programs are probably the most complicated entities ever created by humans. Because of this complexity, programs are prone to error and software errors can be expensive and even life-threatening. Object-oriented programming offers a new and powerful way to cope with this complexity and act as the backbone to all other courses that are based on Object Oriented concept. Therefore by learning this course sincerely the students will be able to develop programs in 'C++' using Object Oriented Programming Concepts.

COMPETENCY (Programme Outcome (POs) according to NBA Terminology):

The course content should be taught and implemented with the aim to develop various types of related skills leading to the achievement of the following competency.

Develop programs in 'C++' using Object Oriented Programming Concepts.

TEACHING AND EVALUATION SCHEME:

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				Total (Marks)
Lecture	Tutorial	Practical	Credit	Theory		Practical		
				University Assessment	Continuous Assessment	University Assessment	Continuous assessment	
3	0	4	5	60	40	40	60	200

DETAILS OF COURSE CONTENTS:

SR. NO.	Topic and Sub-Topics	Total Hrs.	WEIGHTAGE %
Unit – I	Principles of Object Oriented Programming 1.1 Procedure oriented Programming 1.2 Object oriented programming paradigm 1.3 Basic concepts of Object Oriented Programming 1.4 Advantages of Object Oriented Programming 1.5 Object Oriented Languages 1.6 C++ Concepts 1.7 Structure of C++ program 1.8 Applications of C++ 1.9 Basic Data types in C++ 1.10 Operators in C++ 1.11 Scope Resolution Operators 1.12 Member dereferencing Operators 1.13 Memory Management Operators and Manipulators	04	10
Unit - II	Functions in C++ and Working with objects 2.1 The Main Function 2.2 Function prototyping 2.3 Inline functions 2.4 Default Arguments 2.5 Function Overloading	14	30

	2.6 Classes and Objects 2.7 Defining Class and Creating Objects 2.8 Memory Allocation for Objects 2.9 Defining Member function 2.10 Nesting of Member functions 2.11 Private Member functions 2.12 Static Data member and Static Member functions 2.13 Array of Objects 2.14 Friend function		
Unit - III	Constructor and Destructor 3.1 Constructor Concepts 3.2 Destructor 3.3 Parameterized Constructor 3.4 Multiple Constructors in a Class 3.5 Constructor with Default Arguments 3.6 Copy Constructor 3.7 Dynamic Constructor	08	20
Unit - IV	Inheritance 4.1 Define Inheritance 4.2 Concepts of Inheritance 4.3 Defining Derived Classes 4.4 Single Inheritance 4.5 Making a Private Member Inherited 4.6 Multiple Inheritance 4.7 Multilevel Inheritance 4.8 Hybrid Inheritance 4.9 Virtual Base Class 4.10 Abstract Classes	08	20
Unit - V	Pointers, Virtual functions and polymorphism 5.1 Pointers to objects 5.2 'this' Pointer 5.3 Pointer to Derived Classes 5.4 Virtual Functions	06	15
Unit - VI	Managing Console I/O Operations 6.1 Input and Output Streams 6.2 C++ Stream Classes	02	05

COURSE OUTCOME:

The practical/exercises should be properly designed and implemented with an attempt to develop different types of practical skills (Course Outcomes in psychomotor and affective domain) so that students are able to acquire the competencies (Programme Outcomes). Following is the list of practical exercises for guidance.

Course Outcomes in psychomotor domain are listed as practical/exercises. However, if these practical/exercises are completed appropriately, they would also lead to development of Programme Outcomes/Course Outcomes in affective domain as given in a common list at the beginning of curriculum document for this programme. Faculty should refer to that common list and should ensure that students also acquire those Programme Outcomes/Course Outcomes related to affective domain.

SUGGESTED LIST OF EXERCISES/PRACTICAL/EXPERIMENTS:

S. No.	Unit No.	Exercises (Course Outcomes in Psychomotor Domain according to NBA Terminology)	Approx. Hrs. Required
1	I	Develop minimum 5 programs using control structures	2
2	I	Develop minimum 2 programs using arrays	2
3	I	Develop programs using reference variable, scope resolution operator, simple manipulators, and number data type.	4
4	II	Develop programs using call by reference and return by reference, default arguments, constant arguments, and function overloading	4
5	II	Define minimum 5 different classes such as student, distance, shape, employee, feet, time, data etc. with data member & member functions. Also Develop programs to test those classes functionality	4
6	II	Develop Programs using array of objects and static member functions.	4
7	II	Develop programs to pass object as an argument and returning object.	4
8	III	Develop programs using various types of constructors and destructor.	4
9	III	Apply the concepts of constructors and destructors in the programs developed in unit-2 and test those programs.	4
10	IV	Develop programs using single, multilevel, multiple inheritance	6
11	IV	Develop programs using inheritance and constructors	4
12	V	Develop programs using pointer to derived classes	5
13	VI	Develop programs using unformatted i/o functions	5
14	VI	Develop programs using formatted i/o functions	4
Total			56

LIST OF REFERENCES:

Text Books

Object Oriented Programming with C++ by E Balagurusamy (McGraw Hill)

Reference Books

Object Oriented Programming in C++ by Lafore, Robert (SAMS)

Programming in c++ by Kamthane, Ashok (Pearson)

E- Learning References

- C++ Fundamentals : <http://www.oupinheonline.com>
- C++ Tutorials : www.tutorialspoint.com/cplusplus/cpp_overview.htm
- Learn C++ Programming : <http://www.learncpp.com>
- Complete C++ : <http://www.cplusplus.com>
- <https://www.geeksforgeeks.org>
- Video tutorials of C++ :
<http://nptel.iitm.ac.in/syllabus/syllabus.php?subjectId=106101006>