

## 10. CMP 104.3 Object Oriented Programming in C++ (3-1-3)

### Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

### Course Objectives:

- To familiarize with Object Oriented Concept.
- To introduce the fundamentals of C++
- To enable the students to solve the problems in Object Oriented technique
- To cope with features of Object Oriented Programming

### Course Contents:

Chapter	Content	Hrs.
1	<b>Thinking Object Oriented</b>  Object oriented programming a new paradigm, a way of viewing world agent, types of classes, computation as simulation, coping with complexity, nonlinear behavior of complexity, abstraction mechanism	4
2	<b>Classes and Methods:</b>  Review of structures, classes and inheritance, state, behavior, method, responsibility, encapsulation, data hiding, Functions: friend function, inline function, static function, reference variable, default argument	7
3	<b>Message, Instance and Initialization</b>  Message, message passing formalization, message passing syntax in C++, mechanism for creation and initialization (constructor and its types), Issues in creation and initialization: memory map, memory allocation methods and memory recovery	6
4	<b>Object Inheritance and Reusability</b>  Introduction to inheritance, Subclass, Subtype, Principle of Substitutability; Forms of polymorphism and their implementation in C++, inheritance merits and demerits, composition and its implementation in c++, The <i>is-a</i> rule and <i>has-a</i> rule, Composition and Inheritance contrasted, Software reusability	9
5	<b>Polymorphism</b>  Polymorphism in programming language, Varieties of polymorphism, compile time polymorphism, function overloading, operator overloading, type conversion, polymorphic variable, run time polymorphism, object pointer, this pointer, virtual function, overriding, deferred method, pure polymorphism.	8
6	<b>Template and generic programming</b>  Generic and template functions and classes, cases study: container class and the	4

standard template library, Exception handling

## 7 Object oriented Design

7

Reusability implies non- interference, Programming in small and programming in large, components and behaviors, role of behaviors in OOP, CRC, sequence diagram, Software components, formalizing the interface, interface and implementation, Design and representation of components, coming up with names, implementation components, integration of components

### Laboratory Work

There shall be 20 exercises in minimum, as decided by the faculty. The exercises shall encompass a broad spectrum of real-life and scientific problems, development of small program to the development of fairly complex subroutines, programs for engineering applications and problem solving situations. Laboratory assignments will be offered in groups of two to four for evaluation purpose. In general, the Laboratory Work must cover assignments and exercises from the following areas:

1. Data types – control structures, functions and scoping rules.
2. Composite data types, C++ strings, use of " Constant " keyword, pointers and references
3. Classes and data abstraction
4. Inheritance, abstract classes and multiple inheritance
5. Friend functions, friend classes and operator overloading.
6. Static class members
7. Polymorphism, early binding and late binding
8. C++ type conversion
9. Exception handling
10. Function templates, class templates and container classes.

### Textbooks:

1. Budd, T., *An Introduction to Object Oriented Programming*, Second Edition, Addison-Wesley, Pearson Education Asia, ISBN: 81-7808-228-4.
2. R. Lafore, *Object Oriented Programming in Turbo C++*, Galgotia Publications Ltd. India, 1999

### Reference Books:

1. E Balaguruswamy, *Object Oriented Programming with C++*, Third Edition
2. Tata McGraw-Hill ISBN:0-07-059362-0, Parson David, *Object Oriented Programming with C++*, BPB Publication\ISBN817029-447-9