

Module Title:	Computer Programming -2
Module ID:	CSC 113
Prerequisite:	CSC 112
Level:	2
Credit Hours:	4 (3+2+1)

Module Description:

Object-Oriented Concepts and on how the three key concepts, encapsulation, inheritance, and polymorphism, are applied. stream classes, constructors and destructors, data hiding, application classes, generics, containers and the graphical user interface. Introduction to GUI components and API package, Recursion, Class: Operator overloading, Dynamic structure, generics, Inheritance, polymorphism, More on GUI components

Module Aims:

- State the basic concepts of Object Oriented Programming.
- List the benefits of OOP over traditional structured programming
- Help student to master the C++ implementation of object-oriented concepts
- Implement object-oriented programs in C++.programs.

Learning Outcomes:

- Understand the basic OO programming concepts.
- Compare the OO programming approach against the traditional approach.
- Identify the main objects/classes, methods, attributes from given problem specifications.

- Design and code small to medium sized problems from the start using the appropriate OO concepts and other concepts introduced (class, inheritance, polymorphism, generic programming etc.)
- Create and manipulate Files using the available I/O file streams classes.
- Contribute to a group effort to realize an OOP based solution

List of Topics	No. of Weeks	Contact Hours
Chapter 1: Classes and Objects <ul style="list-style-type: none"> • Structure Definition • Classes vs objects • Interface and Implementation • Constructors and Destructors • Set and get Functions • const Objects and const Member 	3	12
Chapter 2: Functions <ul style="list-style-type: none"> • const parameters, const return types • friend Functions and friend Classes • static Class Members • Composition • Dynamic Memory Management (creating object at run-time) • Arrays of objects 	2	8
Chapter 3: Inheritance <ul style="list-style-type: none"> • Base Classes and Derived Classes • protected Member • public, protected and private Inheritance • Inheritance Hierarchy • Software reusability using Inheritance • Multiple inheritance 	2	8
Chapter 4: Polymorphism <ul style="list-style-type: none"> • Relationships Among Objects in an Inheritance Hierarchy • Invoking Base-class Functions from Derived Class Objects • Aiming Derived-Class Pointers at Base Class Objects • Derived-Class Member-Function Calls via Base-Class Pointers • Virtual Functions • Abstract classes and pure virtual functions 	2	8
Chapter 5: Operator Overloading	2	8

<ul style="list-style-type: none"> • Fundamentals of operator overloading • Restrictions of operator overloading • Global and member operators • Overloading Stream-Insertion and Stream-Extraction Operators • Overloading Unary Operators (++,--,! Etc..) • Overloading Binary Operators(+,-,* etc..) 		
<p>Chapter 6: File processing</p> <ul style="list-style-type: none"> • Files and Streams classes in C++ • Creating a Sequential File • Reading Data from a Sequential File • Updating Sequential Files • Random-Access Files • Creating a Random-Access File • Writing and updating Random-Access File 	2	8
<p>Chapter 7: Templates</p> <ul style="list-style-type: none"> • Function Templates • Class Templates • Containers and templates 	2	8

Textbook:

C++: How To Program, Deitel and Deitel, 5th edition, Prentice Hall, 2005.