

<b>Branch: B.Sc(IT)</b>	<b>Semester-II</b>
<b>Subject Code: 2102</b>	<b>Lecture: 04</b> <b>Credit: 04</b>
<b>Course Opted</b>	<b>Core Course -4</b>
<b>Subject Title</b>	<b>PROGRAMMING METHODOLOGY AND C++</b>

**Course Objectives:**

- To understand how C++ improves C with object-oriented features.
- To learn how to design C++ classes for code reuse.
- To learn how to implement copy constructors and class member functions.
- To understand the concept of data abstraction and encapsulation.
- To learn how to overload functions and operators in C++.
- To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
- To learn how to design and implement generic classes with C++ templates.

**Course Outcomes:**

- Students will be able to
- Describe the object-oriented programming approach in connection with C++
- Apply the concepts of object oriented programming
- Analyze a problem and construct a C++ program that solves it
- Discover errors in a C++ program and describe how to fix them
- Illustrate the process of data file manipulations using C++

Modules	Sr. No.	Topic and Details	No. of Lectures Assigned	Marks Weightage %
---------	---------	-------------------	--------------------------	-------------------

Unit- I	1	<p><b>INTRODUCTION:</b> MODELING CONCEPTS, CLASS MODELING: What is Object Orientation? What is OO development? OO themes; Evidence for usefulness of OO development</p> <p><b>Evolution of OOP:</b> Advantages and disadvantages of OOP over its predecessor paradigms. Characteristics of Object Oriented Programming: Abstraction, Encapsulation, Data hiding, Inheritance, Polymorphism, Code Extensibility and Reusability, User defined Data Types.</p> <ul style="list-style-type: none"> <li>• C++ProgramStructure</li> <li>• Simple Input/OutputProgram</li> <li>• Program Comments</li> <li>• Identifiers</li> <li>• Literals</li> <li>• String, Character, Integer,Floating Point, Constants</li> <li>• Keywords</li> <li>• DataTypes</li> </ul> <p><b>Operators in C++</b> <b>Control Structures in C++</b></p>	5	10
	2	<p><b>Advanced Language Constructs</b></p> <ul style="list-style-type: none"> <li>• Arrays</li> <li>• Multi dimensional arrays</li> <li>• Pointers</li> <li>• Structures</li> </ul>	3	6
Unit- II	3	<p><b>Object and Classes :</b></p> <ul style="list-style-type: none"> <li>• Core object concepts</li> <li>• Encapsulation, Abstraction, Polymorphism</li> <li>• Classes, Messages Association, Interfaces</li> <li>• Implementation of class in C++,</li> <li>• C++ Objects as physical object,</li> <li>• C++ object as data types constructor.</li> <li>• Object as function arguments.</li> </ul> <p><b>Functions and Variables</b></p> <ul style="list-style-type: none"> <li>• Functions: Declaration and Definition</li> <li>• Variables: Definition, Declaration, and Scope</li> <li>• Dynamic Creation and Derived Data</li> <li>• Arrays and Strings in C++</li> </ul>	4	8

	4	<b>Inheritance</b> <ul style="list-style-type: none"> <li>• Concept of inheritance</li> <li>• Derived class and based class</li> <li>• Types of inheritance</li> <li>• Classes within classes</li> <li>• Functions and Friend Functions</li> </ul> <b>Constructors</b> <ul style="list-style-type: none"> <li>• Multiple Constructors and Initialization</li> <li>• Using Destructors to Destroy Instances</li> </ul>	6	12
Unit- III	5	<b>Polymorphism</b> <ul style="list-style-type: none"> <li>• Syntax for Operator overloading</li> <li>• Overloading unary operations.</li> <li>• Overloading binary operators</li> <li>• Data conversion</li> <li>• Pitfalls of operators overloading and conversion keywords.</li> </ul>	8	16
	6	<b>Memory management</b> <ul style="list-style-type: none"> <li>• New and Delete</li> <li>• Pointers to objects</li> <li>• Debugging pointers.</li> </ul>	8	16
Unit- IV	7	<b>Files and streams</b> <ul style="list-style-type: none"> <li>• iostream hierarchy</li> <li>• Standard Input/output Stream Library</li> <li>• Programming using Streams, Basic Stream Concepts.</li> </ul> <b>File input and output:</b> <ul style="list-style-type: none"> <li>• Reading a File</li> <li>• Managing I/O Streams</li> <li>• Opening a File – Different Methods</li> <li>• Checking for Failure with File Commands</li> <li>• Checking the I/O Status Flags</li> <li>• Dealing with Binary Files</li> </ul>	8	16

	8	<p><b>Class templates:</b></p> <ul style="list-style-type: none"> <li>• Implementing a class template</li> <li>• Implementing class template member functions</li> <li>• Using a class template</li> <li>• Function templates</li> <li>• Class template specialization</li> <li>• Template parameters, Static members and variables</li> </ul> <p><b>Exception Handling:</b></p> <ul style="list-style-type: none"> <li>• try</li> <li>• throw and</li> <li>• catch constructs</li> <li>• rethrowing an exception</li> <li>• Catch all Handlers.</li> </ul>	8	16
<b>TOTAL</b>			50	100

**Text Books:**

1. E. Balguruswamy, 'Object Oriented Programming with C++', Tata McGraw – Hill Education, 2008
2. K.R Venugopal 'Mastering C++', Tata McGraw-Hill Education, 1997

**References:**

1. B.Stroustrup 'C++ Programming Language' (3rd Edition). Addison Wesley, 1997
2. B.chandraNarosa 'A Treatise On Object Oriented programming using C++'- Publications, 1998
3. Herbert Schildt, "The Complete Reference CN", Tata McGraw-Hili, 2001

<b>Branch: B.Sc(IT)</b>	<b>Semester-II</b>
<b>Subject Code: 2201</b>	<b>Lecture: 02</b> <b>Credit: 02</b>
<b>Course Opted</b>	<b>Core Course -4 Practical</b>
<b>Subject Title</b>	<b>PROGRAMMING METHODOLOGY AND C++ LAB</b>

**Course Objectives:**

- Will enable students to
- Identify and practice the object-oriented programming concepts and techniques
- Practice the use of C++ classes and class libraries, arrays, vectors, inheritance and file I/O stream concepts.
- **Course Outcomes:**

Students will be able to:

- Create simple programs using classes and objects in C++.
- Implement Object Oriented Programming Concepts in C++.
- Develop applications using stream I/O and file I/O.
- Implement simple graphical user interfaces.
- Implement Object Oriented Programs using templates and exceptional handling

Modules	Sr. No.	Topic and Details	No. of Lectures	Marks Weightage
---------	---------	-------------------	-----------------	-----------------