Title	**Object Oriented Programming Lab**		
Code	**CC-211-L**		
Credit Hours	1 (0,3)		
Category	Computing Core		
Prerequisite	CC-112 Programming Fundamentals		
Co-Requisite	None		
Follow-up	CC-213 Data Structures, CC-310 Artificial Intelligence, DC-328 Parallel & Distributed Computing, EC-333 Mobile Application Development		
Course Introduction	The course aims to focus on object-oriented concepts, analysis and software development. The basic concept of OOP is covered in this course.		
Course Learning Outcomes (CLOs)	At the end of the lab, the students will be able to:	**BT**	**PLO**
	CLO1: Understand principles of object-oriented paradigm.	C2 (Understand)	1
	CLO2: Identify the objects & their relationships to build object-oriented solution	C4 (Identify)	2,3,4
	CLO3: Model a solution for a given problem using object-oriented principles	C3 (Apply)	4
	CLO4: Examine an object-oriented solution	C4 (Examine)	4
Course Description	***Implementation and Practice of the concepts studied in "CC-211 Object Oriented Programming"*** Introduction to Object-oriented Design, History and Advantages of Object-oriented Design. **Introduction to OOP and C++:** Brief description of C++ concepts, Introduction to OOP. **Introduction to Classes Objects and Member Functions:** Encapsulation and Abstraction, Class and Object, Getter/Setter Functions, Access Specifiers, Constructors, Overloaded Constructor, Default Constructor, Destructor. **Functions:** Inline Functions, Function Overloading. **Class Templates array:** Function Templates, Class Templates array, Vectors and Multidimensional Array, Reference to private Data Members, Default Member wise Assignment, const Objects, const Member Functions. **Composition and Aggregation:** Object Composition and Aggregation, Class Separation using header. **Friend Classes and Functions:** Friend Functions, Friend Classes. **static Members:** "this" pointer, static Data Members, static Member Functions. **Copy Constructor:** Default Copy Constructor. **Operator Overloading:** Overloaded Operators of Standard Library, Operator Overloading, Overloading Binary Operators, Overloading Unary Operators, Overloading ++ Operator, Overloading – Operator, Dynamic Memory Management, Operators as Members vs Non-Members, Conversion between Types, Explicit Constructor and Conversion Operators, Overloading the Function call Operator. **Stream I/O:** Introduction, Streams, Streams Input, Streams Output, Object Streams, data and object serialization using object streams. **Inheritance:** Introduction, Base and Derived Classes, Relationships between Base and Derived Classes, Constructors in Derived Classes, Destructor in Derived Classes, public protected and private Inheritance. **Polymorphism:** Relationship among Objects in Inheritance, Virtual Functions, Virtual Destructors, Pure Virtual Functions, Abstract and Concrete Classes. **File Processing:** Files and Streams, create a Sequential File, read a Sequential File, update a Sequential File, Random Access File, create a Random-Access File, read a Random-Access File, update a Random-Access File, **Exception Handling:** Flow of Control, Rethrowing an Exception, Constructor Destructor and Exception handling**. Generic Programming Concepts:** Custom Templates, Class Templates, Function Templates, Arguments to Templates, Overloading Function Templates. **Standard Library:** Containers, Iterators, Adapters, Sequence Containers, Associative Containers, Container Adapters, Minimum Iterator Requirements, Lambda Expressions, Function Objects.		
Text Book(s)	A. P. Deitel, H. Deitel, C++ How To Program, 10th Edition, Pearson.		
Reference Material	1. Robert Lafore, Object Oriented Programming in C++, 3rd Edition. 2. Tony Gaddis, Starting Out with C++ from Control Structures to Objects, 9th Edition, Pearson, 2018. 3. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman		