

<b>Title</b>	<b>Object Oriented Programming Lab</b>		
<b>Code</b>	<b>CC-211-L</b>		
<b>Credit Hours</b>	1 (0,3)		
<b>Category</b>	Computing Core		
<b>Prerequisite</b>	CC-112 Programming Fundamentals		
<b>Co-Requisite</b>	None		
<b>Follow-up</b>	CC-213 Data Structures, CC-310 Artificial Intelligence, DI-328 Parallel and Distributed Computing, EI-333 Mobile Application Development		
<b>Course Introduction</b>	The course aims to focus on object-oriented concepts, analysis and software development. The basic concept of OOP is covered in this course.		
<b>Course Learning Outcomes (CLOs)</b>	At the end of the lab, the students will be able to:	<b>BT</b>	<b>PLO</b>
	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
<b>Syllabus</b>	<p><b>Implementation and Practice of the concepts studied in "CC-211 Object Oriented Programming"</b></p> <p>Introduction to Object-oriented Design, History and Advantages of Object-oriented Design. <b>Introduction to OOP and C++:</b> Brief description of C++ concepts, Introduction to OOP. <b>Introduction to Classes Objects and Member Functions:</b> Encapsulation and Abstraction, Class and Object, Getter/Setter Functions, Access Specifiers, Constructors, Overloaded Constructor, Default Constructor, Destructor. <b>Functions:</b> Inline Functions, Function Overloading. <b>Class Templates array:</b> Function Templates, Class Templates array, Vectors and Multidimensional Array, Reference to private Data Members, Default Member wise Assignment, const Objects, const Member Functions. <b>Composition and Aggregation:</b> Object Composition and Aggregation, Class Separation using header. <b>Friend Classes and Functions:</b> Friend Functions, Friend Classes. <b>static Members:</b> "this" pointer, static Data Members, static Member Functions. <b>Copy Constructor:</b> Default Copy Constructor. <b>Operator Overloading:</b> 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. <b>Stream I/O:</b> Introduction, Streams, Streams Input, Streams Output, Object Streams, data and object serialization using object streams. <b>Inheritance:</b> Introduction, Base and Derived Classes, Relationships between Base and Derived Classes, Constructors in Derived Classes, Destructor in Derived Classes, public protected and private Inheritance. <b>Polymorphism:</b> Relationship among Objects in Inheritance, Virtual Functions, Virtual Destructors, Pure Virtual Functions, Abstract and Concrete Classes. <b>File Processing:</b> 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, <b>Exception Handling:</b> Flow of Control, Rethrowing an Exception, Constructor Destructor and Exception handling. <b>Generic Programming Concepts:</b> Custom Templates, Class Templates, Function Templates, Arguments to Templates, Overloading Function Templates. <b>Standard Library:</b> Containers, Iterators, Adapters, Sequence Containers, Associative Containers, Container Adapters, Minimum Iterator Requirements, Lambda Expressions, Function Objects.</p>		
<b>Suggested Instructional/Reading Material</b>	<ol style="list-style-type: none"> <li>1. P. Deitel, H. Deitel, C++ How To Program, 10th Edition, Pearson.</li> <li>2. Robert Lafore, Object Oriented Programming in C++, 3rd Edition.</li> <li>3. Tony Gaddis, Starting Out with C++ from Control Structures to Objects, 9th Edition, Pearson, 2018.</li> <li>4. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly &amp; Elliot B. Koffman</li> </ol>		