

## Course: 304: Object Oriented Programming and Data Structures (OOPP & D.S.)

<b>Course Code:</b>	<b>304</b>
<b>Course Title:</b>	<b>Object Oriented Programming and Data Structures (OOP &amp; D.S.)</b>
<b>Total Credits :</b>	4 Credits
<b>Nature of Subject :</b>	Theory and Practical application
<b>Teaching per Week:</b>	4 Hours per week per Semester
<b>Minimum weeks per Semester:</b>	15 weeks (Including class work, examination, preparation etc.)
<b>Review/Revision Year:</b>	June, 2021
<b>Purpose of Course :</b>	<ul style="list-style-type: none"> <li>- Understand Object Oriented Programming Concepts and skills necessary for developing programs using C++. And it is important for a computer programmer to understand the storage representation and implementation of various data structures used in a computer program. This helps a programmer to use various data structures efficiently which in turn makes the program efficient. This course introduces various data structures, their storage representation &amp; implementation.</li> <li>- Data Structure concepts are important concepts to understand and implement. Purpose of the Data structure is to get basic ideas about how user defined data structures can be implemented. Implementation of Data Structure concept is not language specific.</li> </ul>
<b>Objective :</b>	<ol style="list-style-type: none"> <li>1. This course has been designed for the beginners to help them understand basic to advanced concepts related to C++ Programming language.</li> <li>2. To make students understand the importance of OOP methodology and techniques.</li> <li>3. Basic concepts of data structures, role and importance of data structures in computer programming.</li> <li>4. Distinguish the key difference between storage &amp; implementation of various data structures.</li> <li>5. Recognize the problem properties and determine the use of appropriate data structures in different scenarios.</li> </ol>
<b>Pre-requisite:</b>	Knowledge of C programming Language
<b>Course Outcome :</b>	<ul style="list-style-type: none"> <li>- Students will be able to formulate a computing problem to executable computer program using C++ language.</li> <li>- Understand concepts of class, objects, polymorphism, Inheritance and other important Object oriented concepts.</li> <li>- Understanding about user defined data structures and their importance.</li> <li>- Basic implementations of Stack and Queue.</li> <li>- Concepts of variables, literals, data types, conversions of data types, input and output data and processing of data, inbuilt functions, arrays, header files, conditional and iterative statements.</li> </ul>
<b>Course Content:</b>	<p><b>Unit 1. Concepts of OOPS: ( Max. 20% of Weightage)</b></p> <ol style="list-style-type: none"> <li>1.1 Difference between procedural programming and OOPS</li> <li>1.2 Various library(header) files require for C++</li> <li>1.3 Data types in C++</li> <li>1.4 Concepts of String:               <ol style="list-style-type: none"> <li>1.4.1 character Array</li> <li>1.4.2 pointer to character array</li> <li>1.4.3 Use of String.h and its important functions: (strcmp, strcat, strcpy, strlen, strcmp)</li> </ol> </li> <li>1.5 Concepts of Class and Objects.</li> </ol> <p><b>Unit 2. Data Encapsulation and inheritance: (Max. 20% Weightage)</b></p> <ol style="list-style-type: none"> <li>2.1 Access controls concepts (Public, Private, Protected) and difference among</li> </ol>

	<p>them.</p> <p>2.2 Declaring simple class, member variables and member functions.</p> <p>2.3 Concepts and use of enum.</p> <p>2.4 Concepts of Data hiding, abstraction and encapsulation with examples</p> <p>2.5 Concepts of Inheritance and Types of Inheritance</p> <p>2.6 Constructors and Destructors</p> <p><b>Unit 3.Polymorphism</b> (Max. 20% Weightage)</p> <p>3.1 Concepts of Polymorphism</p> <p>3.2 Compile time and Run time Polymorphism</p> <p>3.3 Overloading and Overriding: Concepts, difference and application</p> <p>3.4 Concepts of friend function</p> <p>3.5 Concepts of virtual function and pure virtual function</p> <p><b>Unit 4.Data Structure</b> (Max. 20% Weightage)</p> <p>4.1 Introduction of Data Structure and application areas.</p> <p>4.2 Recursion concepts</p> <p>4.3 Difference among Linear and Non-Linear Data Structure</p> <p>4.4 Stack</p> <ul style="list-style-type: none"> <li>- Concepts of Stack(LIFO)</li> <li>- Pop, Push and Display(Peep)</li> <li>- Application areas of Stack ( Infix to postfix, Infix to prefix )</li> </ul> <p><b>Unit 5.Queue</b> (Max. 20% Weightage)</p> <p>5.1 Concepts of Queue(FIFO)</p> <p>5.1.1 Concepts of Queues and its basic operations</p> <p>5.2 Implementation of Queue:</p> <p>5.2.1 Simple Queue: insert, delete and display</p> <p>5.2.2 Double ended Queue: insert, delete and display</p> <p>5.2.3 Circular queue: Insert, delete and display.</p>
<b>Reference Books:</b>	<ol style="list-style-type: none"> <li>1. Let us C++, Yaswant Kanitkar - TMH Publication</li> <li>2. Programming with C++, E Balaguruswamy - BPB Publication</li> <li>3. C++ and Object-Oriented Programming Paradigm, Jana - PHI</li> <li>4. The Complete Reference C++, Herbert Schildt - TMH</li> <li>5. The C++ Programming Language, Stroustrup – Addison Wesley</li> <li>6. OOP in Turbo C++, Robert Lafore - Galgotia Publication</li> <li>7. C++ Primer, Lippman – Addison Wesley</li> <li>8. Object Oriented Programming Fundamentals &amp; Applications, Probal Sengupta – PHI</li> <li>9. An Introduction to Data Structures with applications, Trembley – Tata McGraw Hill.</li> <li>10. Algorithms – Data structure programs, Wirth Niclaus - PHI.</li> <li>11. Data structures – A Programming Approach with C, Dharmender Singh Kushwaha and Arun Kumar Misra – PHI.</li> <li>12. Fundamentals of Data structures, Horwitz E. and Sahni – Computer Science Press</li> <li>13. Schaum's outline of Data Structure with C++, John R. H. - Tata McGraw Hill.</li> <li>14. Expert Data Structure with C, R. B. Patel - Khanna Publication</li> <li>15. Data structures - a Pseudocode approach with C++, Richard F. Gilberg and Behrouz A. Forouzan - Thomson books</li> </ol>
<b>Teaching Methodology:</b>	Class Work, Discussion, Self-Study, Seminars and/or Assignments
<b>Evaluation Method:</b>	30% internal assessment. 70% External assessment