

UTA018: Object Oriented Programming				
		L	T	P
		3	0	2
Cr 4.0				
<p>Course Objective: To become familiar with object oriented programming concepts and be able to apply these concepts in solving diverse range of applications.</p>				
<p>Objects and Classes: Structure in C and C++, Class specification, Objects, Namespaces, Overview of pillars of OOPS (Data Encapsulation, Data Abstraction, Inheritance, Polymorphism), Inline functions, Passing objects as arguments, Returning object from a function, Array of objects, Static keyword with data member, member function and object, Friend function, and Friend classes, Pointer to objects, this pointer, Dynamic Initialization, Dynamic memory allocation.</p>				
<p>Constructor and Destructor: Constructors and its types, Constructor Overloading, Constructors in array of objects, Constructors with default arguments, Dynamic Constructor, Destructor, <code>_const</code> keyword with data member, member function and object. Case Study on Optimizing Power Consumption in Smart City Energy Grids.</p>				
<p>Inheritance: Introduction to Inheritance, Forms of Inheritance (Single, Multiple, Multilevel, Hierarchical and Hybrid) with various modes (Public, Private and Protected), Inheritance with Constructor and Destructor, Benefits and Limitations of Inheritance.</p>				
<p>Polymorphism: Classification of Polymorphism (Compile-time and Run-time), Compile Time-Function Overloading, Operator Overloading (Unary operator and Binary operator with member function and friend function), Data Conversion (Basic to user-defined, user-defined to basic, one user-defined to another user-defined). Run-time- Pointers to derived class object, Overriding member function, Virtual functions, pure virtual functions, Abstract class. Case Study on Developing an Eco-Friendly Smart Transportation System.</p>				
<p>Exception Handling, Templates and Standard Template Library: Exception handling mechanism, Usage of template, Function templates, Overloading of Function templates, Class templates, Introduction to Standard Template Library and its components. Algorithms, Containers (Array, Vector, Stack, List and Queue) and Iterators.</p>				
<p>Laboratory Work To implement object oriented constructs using C++ programming language.</p>				
<p>Course Learning Objectives (CLO) The students will be able to:</p> <ol style="list-style-type: none"> 1. To recall the knowledge of structure and its variables to comprehend the concept of classes, objects, constructors and destructors for implementing the object oriented paradigms. 2. To apply and analyze the inheritance on real life case studies via coding competences. 				

3. To design and develop code snippets for polymorphism to proclaim coding potential; and management of run-time exceptions.
4. To assess and interpret the knowledge of templates to appraise the standard template libraries.

Text Books

1. C++:The Complete Reference , Schildt H., Tata McGraw Hill, 4thed, 2003
2. C++Primer, Lippman B.S., Lajoie J., and MooE.B., , Addison-Wesley Professional, 5th ed, 2013

Reference Books

1. Object-Oriented Programming in C++, Lafore R., Pearson Education, 4thed, 2002
2. Object Oriented Programming with C++, E Balagurusamy, 8thed,2017
3. The C++programming language, Stroustrup B., Pearson Education India, 4thed, 2013