

<b>Department/Institute Name</b>	Department of Computer Science and Engineering / Chitkara University Institute of Engineering & Technology		
<b>Program Name</b>	BE -Computer Science and Engineering		
<b>Course Code</b>	CSL3201		
<b>Course Name</b>	Object Oriented Programming		
<b>Lecture / Tutorial (per week)</b>	2-0-0	<b>Course Credits</b>	2
<b>Course Coordinator Name</b>	Er. Tanya Gera		

### 1. Course Objectives

This course intends to inculcate logic building among students and aims to provide the knowledge of object oriented programming approach. At the completion of the course, the students will be able to achieve the following objectives:

1. To Formulate new solutions of programming problems using object oriented approach of programming.
2. To demonstrate the importance of major features of OOP such as encapsulation, inheritance, code extensibility, reusability, and polymorphism.
3. To define their own Templates and implement the generic programming.
4. To evaluate the improved existing programs using Standard Template Library.
5. To compute the efficiency and performance of object oriented programming paradigms.
6. To implement the complex programming problems with integration of file management systems.

### 2. Recommended Books

**RB1:** 'Object Oriented Programming with C++' by E Balagurusamy, 6<sup>th</sup> Edition, Tata McGraw Hill.

**RB2:** Object Oriented Programming in C++ by Robert Lafore, 4<sup>th</sup> Edition, Galgotia.

**RB3:** The Complete Reference C++ by Herbert Schildt , 4<sup>th</sup> Edition ,Tata McGraw Hill.

**RB4:** Stroustrup, Bjarne, The C++ Programming Language, Pearson Education .

**RB5:** Lippman, S.B. and Lajoie, J., C++Primer, Pearson Education .

### 3. Other readings and relevant websites

S.No.	Link of Journals, Magazines, websites and Research Papers
1.	<a href="http://www.cprogramming.com/tutorial/c++-tutorial.html">http://www.cprogramming.com/tutorial/c++-tutorial.html</a>
2.	<a href="http://www.cplusplus.com/doc/tutorial/">http://www.cplusplus.com/doc/tutorial/</a>
3.	<a href="http://www.tenouk.com/cncplusplus-tutorials.html">http://www.tenouk.com/cncplusplus-tutorials.html</a>
4.	<a href="http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-088-introduction-to-c-memory-management-and-c-object-oriented-programming-january-iap-2010/">http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-088-introduction-to-c-memory-management-and-c-object-oriented-programming-january-iap-2010/</a>

**4. Recommended Tools and Platforms:**

GCC compilers, Hackerank and Hackerearth.

**5. Course Plan**

Lecture Number	Topics
1	<b>Introduction:</b> Introduction to basic concepts of object-oriented programming, Comparison between procedural programming paradigm and object-oriented programming paradigm.
2	<b>Functions in C++:</b> inline functions, default arguments, function prototyping, function overloading, call by reference, call by value & call by pointer, return by reference.
3-4	<b>Classes and Objects:</b> Specifying a class, Creating class objects, Accessing class members, Access specifiers – public, private, and protected, Objects and memory, Static members, Static objects, constant member function, constant objects, friend functions, friend class. Passing Object as an argument (by value, by reference, by address), Returning object from a function.
5-6	<b>Constructors and Destructors:</b> Need for constructors and destructors, Copy constructor, Dynamic constructors, Destructors, Constructors and destructors with static members.
<b>Activity-1 (*Code Project)</b>	
7-10	<b>Operator Overloading and Type Conversion:</b> Defining operator overloading, Rules for overloading operators, Overloading of unary operators, binary operators(+,-,/), binary operators using friend functions, manipulation of strings using operators Overloading(>,<>=) <b>Type conversion:</b> Basic type to class type, Class type to basic type, class to class type.
11-12	<b>Dynamic Memory Management &amp; pointers:</b> Understanding pointers, Accessing address of a variable, Declaring & initializing pointers, Accessing a variable through its pointer, Pointer arithmetic, Pointer to a pointer, Pointer to a function, Dynamic memory management - new and delete Operators, Pointers and classes, Pointer to an object, Pointer to a member, this Pointer, Possible problems with the use of pointers - Dangling/wild pointers, Null pointer assignment, Memory leak and allocation failures.
<b>ST-1</b>	
13-16	<b>Inheritance:</b> Introduction, Defining derived classes, Forms of inheritance (single, multilevel, multiple, hybrid & hierarchical), Ambiguity in multiple and multipath inheritance, inheritance with constructor.
<b>Activity-2 (*Code Project)</b>	

17-18	<b>Virtual base class</b> , Overriding member functions, Order of execution of constructors and destructors.
19-20	<b>Virtual Functions and Polymorphism</b> : Concept of Binding - Early binding and late binding, Virtual functions, Pure virtual functions, Abstract classes, Virtual destructors & polymorphism.
<b>ST-2</b>	
21	<b>Exception Handling</b> : Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions.
22	<b>Templates and Generic Programming</b> : Function templates, Class templates, overloading of template functions.
<b>Activity-3 (*Code Project)</b>	
23-24	<p><b>Introduction to the Standard Template Library</b>: STL Components(Container, Algorithms and Iterators)</p> <p><b>CONTAINERS</b>:</p> <p>1.<b>Sequence Container</b>:</p> <p><b>vector</b>(push_back(),pop_back(),back(),size(),empty())  <b>list</b>(push_front(),pop_front(),front(),size(),empty())  <b>de-queue</b>(push_back(),pop_back(),push_front(),pop_front(),size(),empty())</p> <p>2.<b>Associative Container</b>:</p> <p><b>set</b>(Insert(),erase(),Size(),Empty(),Count(),Clear())  <b>multiset</b>(Insert(),erase(),Size(),Empty(),Count(),Clear())  <b>map</b>(Insert(),erase(),Size(),Empty(),Count(),Clear())  <b>multimap</b>(Insert(),erase(),Size(),Empty(),Count(),Clear())</p> <p>3.<b>Derived Container</b>: stack, queue, priority_queue</p> <p><b>ALGORITHMS</b>:</p> <p>count(),count_if(),find(),find_if(),copy(),fill(),remove(),remove_copy(),replace(),replace_copy(),reverse(),reverse_copy(),unique(),unique_copy(),max(),max_element(),min(),min_element()</p> <p><b>ITERATORS</b>: input, output, forward</p> <p><b>VECTORS</b>:back(),begin(),clear(),empty(),end(),erase(),pop_back(),push_back()</p>
25	<b>Console I/O</b> : Concept of streams, Input/ Output using Overloaded operators >> and << and Member functions of I/O stream classes.
26	<b>Data Files management</b> : File streams, Hierarchy of file stream classes, Error handling during file operations, Reading/Writing of files, Accessing records randomly.
<b>ST-3</b>	

\*Code Project- Inter and Intra Section Coding contest

**6. Evaluation Scheme (Theory):**

Component	Type of Evaluation	Test Marks	Paper Format			
			1 Mark MCQ	2 Mark MCQ	5 Mark Code	10 Mark Code
Component 2*	Sessional Tests (STs)	40	10	05	02	01
Component 3**	End Term Examination	60	10	10	02	02

	<b>Total</b>	<b>100</b>	-	-	-	-
--	--------------	------------	---	---	---	---

\*There will be three sessional tests out of which best two will be considered for evaluation. Each test will be of 40 marks.

\*\*The End Term Comprehensive examination will be held at the end of semester. The mandatory requirement of 75% attendance in all theory classes is to be met for being eligible to appear in this component.

## 7. SYLLABUS

Topics	No of lectures	Weightage
<b>Introduction:</b> Introduction to basic concepts of object-oriented programming, Comparison between procedural programming paradigm and object-oriented programming paradigm.	1	5%
<b>Functions in C++:</b> inline functions, default arguments, function prototyping, function overloading, call by reference (call by value & call by pointer), return by reference.	1	5%
<b>Classes and Objects:</b> Specifying a class, Creating class objects, Accessing class members, Access specifiers – public, private, and protected, Classes, Objects and memory, Static members, Static objects, constant member function, constant objects, friend functions , friend class. Passing Object as an argument (by value, by reference, by address), Returning object from a function.	2	5%
<b>Constructors and Destructors:</b> Need for constructors and destructors, Copy constructor, Dynamic constructors, Destructors, Constructors and destructors with static members.	2	10%
<b>Operator Overloading and Type Conversion:</b> Defining operator overloading, Rules for overloading operators, Overloading of unary operators, binary operators(+,-,/), binary operators using friend functions, manipulation of strings using operators Overloading(>,<,<=,=) <b>Type conversion</b> :Basic type to class type, Class type to basic type, class to class type.	4	10%
<b>Dynamic Memory Management &amp; pointers:</b> Understanding pointers, Accessing address of a variable, Declaring & initializing pointers, Accessing a variable through its pointer, Pointer arithmetic, Pointer to a pointer, Pointer to a function, Dynamic memory management - new and delete Operators, Pointers and classes, Pointer to an object, Pointer to a member, this Pointer, Possible problems with the use of pointers - Dangling/wild pointers, Null pointer assignment, Memory leak and allocation failures.	2	5%
<b>Inheritance:</b> Introduction, Defining derived classes, Forms of inheritance, Ambiguity in multiple and multipath inheritance, Virtual base class, Overriding member functions, Order of execution of constructors and destructors, inheritance with constructor.	3	12%

Virtual base class, Overriding member functions, Order of execution of constructors and destructors, inheritance with constructor.	3	7%
<b>Virtual Functions and Polymorphism:</b> Concept of Binding - Early binding and late binding, Virtual functions, Pure virtual functions, Abstract classes, Virtual destructors & polymorphism.	2	7%
<b>Exception Handling:</b> Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions.	1	7%
<b>Templates and Generic Programming:</b> Function templates, Class templates, overloading of template functions.	1	7%
<b>Introduction to the Standard Template Library: STL Components(Container,Algorithms and Iterators)</b> <b>CONTAINERS:</b> <b>1.Sequence Container:</b> <b>vector</b> (push_back(),pop_back(),back(),size(),empty()) <b>list</b> (push_front(),pop_front(),front(),size(),empty()) <b>de-queue</b> (push_back(),pop_back(),push_front(),pop_front(),size(),empty()) <b>2.Associative Container:</b> <b>set</b> (Insert(),erase(),Size(),Empty(),Count(),Clear()) <b>multiset</b> (Insert(),erase(),Size(),Empty(),Count(),Clear()) <b>map</b> (Insert(),erase(),Size(),Empty(),Count(),Clear()) <b>multimap</b> (Insert(),erase(),Size(),Empty(),Count(),Clear()) <b>3.Derived Container:</b> stack, queue, priority_queue  <b>ALGORITHMS:</b> count(),count_if(),find(),find_if(),copy(),fill(),remove(),remove_copy(),replace(),replace_copy(),reverse(),reverse_copy(),unique(),unique_copy(),max(),max_element(),min(),min_element() <b>ITERATORS:</b> input,output,forward <b>VECTORS:</b> back(),begin(),clear(),empty(),end(),erase(),pop_back(),push_back()	2	14%
<b>Console I/O:</b> Concept of streams, Hierarchy of console stream classes, Input/ Output using Overloaded operators >> and << and Member functions of I/O stream classes.	1	3%
<b>Data Files management:</b> File streams, Hierarchy of file stream classes, Error handling during file operations, Reading/Writing of files, Accessing records randomly	1	3%

This Document is approved by:

Designation	Name	Signature
-------------	------	-----------

Course Coordinator	Er. Tanya Gera	
Associate Dean	Er. Sudha Goyal	
Deputy Dean	Er. Meenu Khurana	
Date	Dec 30, 2017	