

Object Oriented Programming

Course Title: Object Oriented Programming
Course No: CSC161
Nature of Course: Theory + Lab
Semester: II

Full Marks: 60 + 20 + 20
Pass Marks: 24 + 8 + 8
Credit Hrs: 3

Course Description: The course covers the basic concepts of object oriented programming using C++ programming language.

Course Objectives: The main objective of this course is to understand object oriented programming and advanced C++ concepts such as composition of objects, operator overloads, inheritance and polymorphism, file I/O, exception handling and templates.

Detail Syllabus

Chapters / Units	Teaching Methodology	Teaching Hours
<p style="text-align: center;">Unit 1: Introduction to Object Oriented Programming</p> <ul style="list-style-type: none"> - Overview of structured programming approach; Problems with structured programming - Object oriented programming approach - Characteristics of object oriented languages: <u>objects, classes, data abstraction and encapsulation, polymorphism, and overloading, inheritance</u> 	Class Lecture	3 Hours
<p style="text-align: center;">Unit 2: Basics of C++ programming</p> <ul style="list-style-type: none"> - C++ Program Structure, - Character Set and Tokens (<u>keywords, identifiers, constants, operators, special symbols</u>), - Data Type (<u>basic data types, derived data types, and user defined data types</u>) - Type Conversion (<u>explicit conversion: type cast operator, implicit conversion</u>), - Preprocessor Directives (<u>include and define directives</u>), - Namespace, - Input/Output Streams (<u>cin and cout</u>) and Manipulators (<u>setw and endl</u>), - Dynamic Memory Allocation with new and delete, - Control Statements (<u>abstract review of decision making and branching, decision making and looping</u>) - Functions: Function Overloading(<u>different</u> 	Class Lecture + Lab Session	5 Hours

<p><i>number of arguments, different types of arguments</i>) , Inline Functions, Default Argument, Pass by Reference, Return by Reference,</p> <ul style="list-style-type: none"> - Scope/Visibility (<i>local, file, class</i>) and Storage Class. - Pointers: Pointer variables declaration and initialization, Operators in pointers (<i>address-of and value-at-the-address</i>), Pointers and Arrays, Pointer and Function (<i>passing simple variables, passing arrays</i>). 		
<p style="text-align: center;">Unit 3: Classes & Objects</p> <ul style="list-style-type: none"> - A Simple Class and Object, (<i>class definition: data members, member functions</i>), - Accessing members of class, - Initialization of class object (Using Constructor: Default Constructor, Parameterized Constructor, Copy Constructor, The Default Copy Constructor), - Destructor - Objects as Function Arguments, - Returning Objects from Functions, - Structures and Classes, - Memory allocation for Objects, - <i>Static data members</i>, - Member functions defined outside the class (<i>using scope resolution operator</i>). 	<p style="text-align: center;">Class Lecture + Lab Session</p>	<p style="text-align: center;">8 Hours</p>
<p style="text-align: center;">Unit 4: Operator Overloading</p> <ul style="list-style-type: none"> - Fundamental of operator overloading, Restriction on operator overloading, Operator functions as a class members, - Overview of unary and binary operator (<i>operator keyword, operator arguments, operator return values, nameless temporary objects</i>), - <i>Prefix and postfix unary operator overloading,</i> - <i>Overloading binary operator (arithmetic operators overloading, comparison operator overloading, assignment operator overloading,)</i> - Data Conversion (basic to basic, basic to user-defined, user-defined to basic, user-defined to user-defined: (<i>routine in source object, routine in destination object</i>)) 	<p style="text-align: center;">Class Lecture + Lab Session</p>	<p style="text-align: center;">7 hours</p>

<p style="text-align: center;">Unit 5: Inheritance</p> <ul style="list-style-type: none"> - Introduction to inheritance, - Derived Class and Base Class, - Access Specifiers (private, protected, and public), - <i>Overriding member functions</i> - Types of inheritance(<i>simple, multiple, hierarchical, multilevel, hybrid</i>) - <i>Abstract base class</i> - Public and Private Inheritance, - Constructor and Destructor in derived classes, - <i>Ambiguity in multiple inheritance,</i> - Aggregation (<i>class with in Class</i>) 	<p>Class Lecture + Lab Session</p>	<p>7 Hours</p>
<p style="text-align: center;">Unit 6: Virtual Function, Polymorphism, and miscellaneous C++ Features</p> <ul style="list-style-type: none"> - Concept of Virtual functions(<i>difference between normal member function accessed with pointers and virtual member function accessed with pointers</i>) - Late Binding, Abstract class and pure virtual functions, - Virtual Destructors, Virtual base class, - Friend function , <i>friend Class</i> - Static function, - Assignment and copy initialization, Copy constructor, - This pointer, - Concrete classes(<i>vs. abstract class</i>) - Polymorphism and its roles. 	<p>Class Lecture + Lab Session</p>	<p>5 Hours</p>
<p style="text-align: center;">Unit 7: Function Templates and Exception Handling</p> <ul style="list-style-type: none"> - Function templates, - Function templates with multiple arguments, - Class templates, - Templates and inheritance, - Exceptional Handling (Try, throw and catch), - <i>Multiple exceptions, exceptions with arguments</i> - Use of exceptional handling. 	<p>Class Lecture + Lab Session</p>	<p>4 Hours</p>
<p style="text-align: center;">Unit 8: File handling <i>and Streams</i></p>	<p>Class Lecture +</p>	<p>6 Hours</p>

<ul style="list-style-type: none"> - Stream Class Hierarchy for Console Input /Output (<i>fstream base, ifstream, ofstream and fstream</i>) (<i>binary vs. character files</i>) - Unformatted Input /Output, - Formatted Input /Output with ios Member functions, - Formatting with Manipulators, - File Input/output with Streams, - Opening and Closing files (<i>open() and close() member functions</i>), - Read/Write from File (<i>put(), and get(), read () and write() member functions</i>) - File Access Pointers and their Manipulators (<i>seekg(), seekp(), tellg(), tellp(), offset, ios::beg, ios::cur, ios::end</i>) - Sequential, Random Access to File - Testing Errors during File Operations (<i>eof(), fail(), bad(), good()</i>) - Stream Operator Overloading(<i>overloading extraction and insertion operators</i>) 	Lab Session	
---	--------------------	--

Text Book:

1. Robert Lafore, Object Oriented Programming in C++, Fourth Edition, SAMS publications.
2. Herbert Schildt, C++ The Complete Reference, Fourth Edition, Tata McGraw Hill Publication.

Reference Books:

1. Deitel and Deitel, C++ How to Program, Third Edition, Pearson Publication.
2. Joyce Farrell, Object-oriented programming using C++, Fourth Edition, Cengage Learning.

Laboratory work manual

This course requires a lot of programming practices. Each topic must be followed by a practical session. Practical sessions for each unit should be conducted and should include writing the programs in C++. The instructors have to prepare lab sheets for individual units covering the each concepts of the units as per the requirement. The sample lab sessions can be as following descriptions;

For Unit 2:

- Write programs for illustrating the concepts of
 - o input/output streams and manipulators,
 - o dynamic memory operators with new and delete operators.
 - o function overloading, inline functions, default arguments, pass by reference, return by reference

For Unit 3:

- Write programs for illustrating the concepts of
 - class and object
 - constructor (default, parameterized, copy)
 - destructor
 - objects as function arguments
 - returning objects from functions

For Unit 4:

- Write programs for illustrating the concepts of
 - unary operator overloading (prefix and postfix)
 - binary operator overloading (Arithmetic , comparison and assignment)
 - data conversion (basic to basic, basic to user-defined, user-defined to basic, user-defined to user-defined)

For Unit 5:

- Write programs for illustrating the concepts of
 - base class and derived class
 - protected access specifier
 - overriding member function
 - public and private inheritance
 - constructor in derived class

For Unit 6:

- Write programs for illustrating the concepts of
 - abstract class and pure virtual function
 - friend function and friend class
 - static function
 - this pointer

For Unit 7:

- Write programs for illustrating the concepts of
 - Function templates and class templates
 - Templates and Inheritance
 - Exceptional handling using try, throw and catch
 - Multiple exceptions , Exceptions with arguments

For Unit 8:

- Write programs for illustrating the concepts of
 - ifstream, ofstream and fstream
 - Opening and Closing files using open () and close () member functions
 - Read/Write from File using put(), and get(), read () and write() member functions
 - File Access Pointers and their Manipulators using seekg(), seekp(), tellg(), tellp(), offset, ios::beg, ios::cur, ios::end
 - Testing Errors during File Operations using eof(), fail(), bad(), good()

Model Question
Tribhuvan University
Institute of Science and Technology

Course Title: Object Oriented Programming
Course No: CSC161
Level: B. Sc CSIT First Year/ Second Semester

Full Marks: 60
Pass Marks: 24
Time: 3 Hrs

Section A
Long Answer Questions

Attempt any two questions.

*[2*10=20]*

1. What is object oriented programming? Explain objects, class, encapsulation, data hiding, inheritance, and polymorphism.
2. Explain operator overloading. Write a program that overloads insertion and extraction operators.
3. What is inheritance? Explain the ambiguities associated with multiple inheritance with suitable example programs.

Section B
Short Answer Questions

Attempt any eight questions.

*[8*5=40]*

4. Explain the purpose of a namespace with suitable example.
5. What is the principle reason for passing arguments by reference? Explain with suitable code.
6. Why constructor is needed? Explain different types of constructors with example.
7. Write a program that illustrates the conversions between objects of different classes having conversion function in source object.
8. Explain the difference between private and public inheritance with suitable diagram.
9. Why friend function is required? Discuss with example.
10. How late binding is different from early binding. Write a program that explains late binding using virtual function.
11. Why do we need exceptions? Explain “exceptions with arguments” with suitable program.
12. What are the advantages of using the stream classes for I/O? Write a program that writes object to a file.