Object Oriented Programming in C++

EG 1211 CT

Total: 6 hour /week
Year: I **Lecture: 3 hours/week**
Semester: II **Tutorial: hours/week**
Practical: hours/week
Lab: 3 hours/week

Course Description:

This course deals with the object oriented programming technique using the C++ programming language. This course introduces the concepts of object-oriented programming to students with a concept of object, class, inheritance, polymorphism, encapsulation and data abstraction. It also gives knowledge about virtual functions, input/output streams and files and templates in C++.

Course Objectives:

After completing this course the students will be able to:

- analyze the problem with object oriented approach
- design the problem using object oriented design methods
- implement the problem in C++ in object oriented way
- understand various object oriented concepts such as class/object, abstraction, inheritance, operator overloading, dynamic binding, templates in C++ programming language

Course Contents:

Unit1. Object Oriented Programming: [4]

1.1. Software Evolution
1.2. Basics of object oriented programming
  - Procedure oriented programming
  - Object oriented programming
  - Procedure oriented versus Object oriented programming

1.3. Elements of Object Oriented programming
  - Class & Object
  - Abstraction & Encapsulation
  - Inheritance
  - Polymorphism
  - Dynamic binding
  - Message passing

1.4. Object oriented languages
1.5. Advantage and Disadvantage of OOP

Unit2. Introduction to C++ [2]

2.1. History and Evolution of C++
2.2. Why C++
2.3. Features of C++

Unit3. C++ Language Basics: [5]

3.1. Character set, tokens (keywords, identifiers, operators)
3.2. Commenting
3.3. Variable declaration

3.4. Data type
3.5. Type Conversion and promotion rules
3.6. Input/ Output basics
3.7. Preprocessor directives
3.8. Control structures
3.9. Array, Pointer, String
3.10. Dynamic memory allocation
3.11. Functions
- Function overloading
- Default argument
- Inline function
- Pass by reference
- Return by reference

3.12. const construct
3.13. Structure and Unions

Unit4. Object and Class: [10]

4.1. Class syntax (similarities with structures)
4.2. Data Encapsulation (public, private modifiers)
4.3. Object and the member access
4.4. Defining member function (inside and outside of the class)
4.5. Constructor and Destructor
4.6. Objects as function arguments
4.7. Returning objects from functions (nameless object)
4.8. Array of objects
4.9. Pointer to objects
4.10. Dynamic memory allocation for objects and object array
4.11. this pointer (returning object using this pointer)
4.12. static data and function members
4.13. inline function
4.14. Constant data member of a class
4.15. Constant member functions and constant objects
4.16. friend function and friend class

Unit5. Overloading Operators: [5]

5.1. Over loadable operators
5.2. Syntax of operator overloading
5.3. Unary operator overloading
5.4. Binary operator overloading
5.5. Operator overloading using member operator functions (unary and binary)
5.6. Operator overloading using friend operator functions (unary and binary)
5.7. Index operator overloading
5.8. Data conversion
- Basic to basic (explicit and implicit)
- Basic to user defined and vice versa
- User defined to user defined

Unit6. Inheritance: [5]

6.1. Base and derived class (definition with diagrams)
6.2. protected access specifier (show whole class syntax including protected)
6.3. Syntax of derived class declaration (visibility modes)
6.4. Types of inheritance
- Single

- Multiple
- Multilevel

6.5. Scope of inherited member functions and variables
6.6. Constructors in derived and base class
6.7. Destructor in Derived and base class
6.8. Member function and data overriding
6.9. Ambiguity in member access in overriding members
6.10. Virtual base class

Unit7. Virtual functions: **[3]**

7.1. Pointer to derived class object
7.2. Array of pointers to derived class objects with function overriding
7.3. Need of virtual functions
7.4. Virtual functions definition
7.5. Pure Virtual functions and Abstract classes
7.6. Virtual Destructor

Unit8. Input/output Streams and Files: **[7]**

8.1. Input/output Stream class hierarchy
8.2. Unformatted Input/output
8.3. Formatted Input/output
- ios Stream class member functions and flags
- Standard manipulators
- User defined manipulators

8.4. File I/O with streams
8.5. File stream class hierarchy
8.6. Operations on files
8.7. ASCII and Binary files
8.8. Opening file, file modes and closing files
8.9. File read/write using stream and using read & write function
8.10. File pointers and their manipulators
8.11. Testing for errors during file operations

Unit9. Templates: **[4]**

9.1. Function Template
9.2. Overloading function template
- Overloading with functions
- Overloading with other template

9.3. Class Template
9.4. Function definition outside of the class template

Practical:

[45]

The lab exercise shall different aspects and features of the C++ programming language.

1. Programming with the structured components of the C++ language
2. Simple class and its implementation
3. Creating classes for data types such as complex no, date, time, distance etc and implement them in a program
4. Using constructors and destructors along with the objects
5. Using static and constant member functions and data
6. Using friends functions to act as bridge between the objects

7. Programs to overload different operators
8. Program to convert data from user defined to fundamental data and vice versa, and user defined to user defined type
9. Program to inherit the base class to add new functionality in the base class with example of single, multiple and multilevel inheritance.
10. Using virtual functions pointer to objects in program
11. Binary and ASCII file manipulation
12. Program to create and use function and class templates

Reference books:

1. Balagurusamy, E. (2013). Object oriented programming with C++. New Delhi: Tata McGraw-Hill.
2. Robert Lafore, "Object Oriented Programming in C++", Fourth Edition, Waite Group/Galgotia Publication, India
3. Deitel & Deitel, "C++ How to program", Second Edition, Pentice Hall India
4. Herbert Schildt, "C++: The Complete Reference", Fourth Edition, Tata McGraw Hill, India Schildt, H. (2003). C++: the complete reference (4th ed). New York: McGraw-Hill.