# CS-104 Object Oriented Programming

**Credit Hours 4 (3-1)**

|  |  |  |
| --- | --- | --- |
| **Program:** BSCS | | **Semester:** II |
| **Pre Requisite:** Programming Fundamentals | | **Follow Up:** Data Structures and Algorithms, Introduction to Software Development |
| **Course Description**  The course is designed to teach the concepts of Object Oriented paradigm. Object Oriented Design is concerned with developing an object-oriented model of a software system to implement the identified requirements Object Oriented Design. It offers a powerful way to cope with program complexity. Object-oriented programming languages (OOPLs) are the natural choice for implementation of an Object-Oriented Design because they directly support the object notions of classes, inheritance, information hiding, and dynamic binding. Because they support these object notions, Object Oriented Programming Languages make an object-oriented design easier to implement. | | |
| **Course Objectives**  The basic goal is to develop such programs that are clear, reliable and easily maintainable. The course includes concepts like Data Encapsulation, Classes, Objects, Constructors & Destructors, Operator and Function overloading, Virtual functions & Polymorphism, Inheritance, I/O & File handling, Templates & Exception handling. | | |
| **Learning Outcomes**  Upon completion of this course the student should be able to:   * Explain object-oriented programming concepts like- classes, instance variables, instance methods, constructors * Explain advanced object-oriented programming concepts - inheritance, polymorphism * Demonstrate creativity and problem-solving skills. * Demonstrate proper use of the object-oriented principle of inheritance. * Given a problem description, the student should be able to decide on appropriate classes in a class hierarchy tree. * Demonstrate proper use of the object-oriented principle of composition. * Given a description that involves two classes, the student should be able to decide on whether composition or inheritance is more appropriate. | | |
| **Textbooks**  How to Programe C++ by Deitel & Deitel 5/e, Pearson | | |
| **Reference Books / Material**   * *An Introduction to Object-Oriented Programming with Java*, C. Thomas Wu (2010). 5th Edition. McGraw-Hill. ISBN: 9780073523309 * *Java: How to Programme*, 5/e, Deitel and Deitel, Prentice Hall, 0131016210/ 0131202367 International Edition * *Ivor Horton’s Beginning Java*, 7/e, Ivor Horton * *C++: How to Programme*, Deitel and Deitel, 5/e, Pearson * *Object Oriented Programming in C++*, 3rd Edition, Robert Lafore | | |
| **Course Distribution** | Theory: 45%  Problem Analysis: 20%  Solution Design: 30%  Social and Ethical Issues: 5% | |
| **Marks Distribution** | Test1: 10 Marks,  Test 2: 10 Marks  Assignment: 05 Marks,  Presentation: 05 Marks,  Final: 55 Marks  Practical: 10 Marks  Attendance: 05 Marks | |
| **Technology Involved** (Multimedia, Overhead Projector, Web, etc.)  Multimedia, Web | | |
| **Practiced Techniques** (Class Room Lecture, Presentation, Workshop, Group Discussion, Survey, etc.)  Class Room Lecture, Presentation, Group Discussion. | | |

**Theory**

|  |  |  |
| --- | --- | --- |
| **Week** | **Topic** | **Learning Activities** (Assignments, Quizzes, Presentations, etc.) |
| 1 | Types of programming. Different approaches. Modular approach. Object Oriented Approach. Different Systems as objects. Structured programming, procedural programming. OOP features. Data Hiding, Encapsulation, Reusability. |  |
| 2 | Classes. Data as characteristics. Functions as behaviour. Class definition, prototype, Access Specifiers. Private, Public. Classes and their objects. | Assignment |
| 3 | Functions in classes. Parameterized functions and parameter- less functions. Constructors and their properties. Function Overloading. Function Overriding. |  |
| 4 | Constructors & their properties. Parameter-less constructor. Parameterized constructor. Sequence of execution, Default Constructor. Copy constructor. Destructors. | Quiz |
| 5 | Inheritance, reusability, extensibility. Base class/Parent class, Inherited class / Child class / Derived class. |  |
| 6 | Inheritance modes. Types of Inheritance, Multilevel Inheritance, Multiple Inheritance. Diamond Problem. Virtual Multiple Inheritance. |  |
| 7 | Polymorphism, Virtual functions, Pointer to class objects. Dynamic memory allocation. | Reading Assignment |
| 8 | Abstract classes/ Concrete classes, Pure virtual functions. |  |

|  |  |  |
| --- | --- | --- |
| **Week** | **Topic** | **Learning Activities** (Assignments, Quizzes, Presentations, etc.) |
| 9 | Composition. Composition Vs. Inheritance. Composition examples. |  |
| 10 | Multiple Composition, Multilevel Composition. Indirect composition. |  |
| 11 | Friend functions. Friend classes. Friendship and different properties. | Presentation |
| 12 | Operator Overloading. Overloading Unary operators. Overloading Binary Operators. |  |
| 13 | Overloading stream insertion and stream extraction operators. Overloading copy constructor and assignment operator. | Quiz |
| 14 | File Handling. File Operations, create, read, write, modify, delete. |  |
| 15 | File pointers, exception handling. | Assignment |
| 16 | Project assignment and discussions |  |
| **End Term Exam** | | |

|  |
| --- |
| **Lab Objectives**  The basic goal is to develop such programs that are clear, reliable and easily maintainable. The students should be able to implement concepts like Data Encapsulation, Classes, Objects, Constructors & Destructors, Operator and Function overloading, Virtual functions & Polymorphism, Inheritance, I/O & File handling, Templates & Exception handling. |
| **Lab Outcomes**  Upon completion of the practical work the student should be able to implement object-oriented programming concepts like- classes, instance variables, instance methods, constructors, advanced object-oriented programming concepts - inheritance, polymorphism, principle of composition. |
| **Textbooks / Lab Manual**   * How to Program C++ by Deitel&Deitel |
| **Reference Books / Material**   * How to Program C++ by Deitel&Deitel * Object Oriented by Robert Lafore * C++ Programming by Satish Jain * Object Oriented Programming with C++ David Parsons |
| **Technology Involved** (Multimedia, Overhead Projector, Web, etc.)  **Multimedia, Web** |
| **Practiced Techniques** (Lab Lecture, Presentation, Workshop, Group Discussion, Survey, etc.)  Lab Lecture, Presentation, Programming Assignments, Group Discussion. |

**Lab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | | **Topic** | | **Learning Activities** (Practical, Case Study, etc.) |
| 1 | | Revision of structured programming, comparison of structured and Object Oriented concepts. | | Lab |
| 2 | | Classes. Class definition, prototype, Access Specifiers. Private, Public. Classes. Class objects. | | Lab |
| 3 | | Parameterized functions and parameter- less functions. Constructors. Function Overloading. Function Overriding. | | Lab |
| 4 | | Constructors, Parameter-less constructor, Parameterized constructor. Default Constructor. Copy constructor. Destructors. | | Lab |
| 5 | | Inheritance, Base class/Parent class, Inherited class / Child class / Derived class. | | Lab |
| 6 | | Inheritance modes. Multilevel Inheritance, Multiple Inheritance. | | Lab |
| 7 | | Polymorphism, Virtual functions. Pointer to class objects. | | Lab |
| 8 | | Abstract classes/ Concrete classes, Pure virtual functions. | | Lab |
| 9 | Composition. | | Lab | |
| 10 | Multiple Composition, Multilevel Composition. Indirect composition. | | Lab | |
| 11 | Friend functions. Friend classes. | | Lab | |
| 12 | Operator Overloading. Overloading Unary operators. Overloading Binary Operators. | | Lab | |
| 13 | Overloading stream insertion and stream extraction operators. | | Lab | |
| 14 | Overloading copy constructor and assignment operator. | | Lab | |
| 15 | File Operations, create, read, write, modify, delete. | | Lab | |
| 16 | File pointers. | | Lab | |
| **End Term Exam** | | | | |