**Institute of Information Technology & Management, New Delhi**

**Lesson Plan for C++ Lab**

**Programme: BCA Semester: III (M2) Paper Code: 253 Academic Year: 2022-23**

Date of commencement of classes: 23rd July’ 2007

No of hours allotted to complete the syllabi\*: 44

**Course Objective:**

* To gain knowledge of objects, Class, Data Abstraction, Encapsulation, Inheritance, Polymorphism and Dynamic Binding.
* To know about constructing programs using Bottom-up design approach.
* The objectives of the course are to have students identify and practice the object-oriented programming concepts and techniques, practice the use of C++ classes and class libraries, develop C++ classes for simple applications, and practice the concepts of Object-Oriented programming.

**Course Outcomes:**

**CO1:** To be able to prepare object-oriented design for small/medium scale problems.

**CO2:** To demonstrate the differences between traditional imperative design and object- oriented design.

**CO3:** To understand class structures as fundamental, modular building blocks

**CO4:** To understand the role of inheritance, polymorphism, dynamic binding and generic structures in building reusable code

**CO5:** To write small/medium scales C++ Project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Outcomes** | **Programme Outcomes** | | | | |
|  | **PO1**  **Core Subject knowledge** | **PO2**  **Enquiry-based learning** | **PO3**  **Cognitive skills and critical thinking** | **PO4**  **Communication, Adaptive & Interactional Skills** | **PO5**  **Holistic Outlook** |
| **CO1** | **✓** |  |  | **✓** |  |
| **CO2** | **✓** |  | **✓** |  |  |
| **CO3** | **✓** |  | **✓** |  |  |
| **CO4** | **✓** | **✓** | **✓** | **✓** |  |
| **CO5** |  | **✓** | **✓** | **✓** | **✓** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lab No** | **Unit No** | **Application Area/Functionality/ Concept on which practical is based** | **Problem Statement** | **No. of SessionsAllotted** |
| L1 | I | If-else construct, loops, Arrays and Pointers  cin, cout, new and delete operators, Reference variable, class | 1. **WAP to check whether the number is prime or not.** 2. **WAP to find the reverse of an inputted number.** 3. **WAP to find the sum of the digits of a number.** 4. **WAP to find the factorial of a number.** 5. **WAP to print Fibonacci series** 6. **WAP for matrix addition, subtraction and multiplication.** | 4 |
| L2 | I | Structures, Strings in c++ | 1. **W AP to create a structure Student and add functions to read and show details.** 2. **W AP to find the length of a string.** 3. **W AP to concatenate two strings using your defined Function.** | 4 |
| L3 | I,II | Class, Constructor, Destructor, Default Parameter, Polymorphism – function overloading | 1. **W AP to show the working of a class Student add functions to read and show details.** 2. **WAP to define a class circle and add functions to perform following tasks. Read, display and calculate area.** 3. **W AP to overloaded function area ( ) for calculating the area of triangle and circle** 4. **W AP to create Student class and add constructors, other functions and destructor.** 5. **W AP a program to demonstrate constructor function overloading.** | 4 |
| L4 | II | Polymorphism-Operator overloading | 1. **Program to implement const keyword. Write a program to create class complex having two data members real and imag. Define constructors default and parameterized. Define function display to display the object values with const keyword.** 2. **W AP to create class DISTANCE and overload Binary + ,- operator to add and subtract two objects and compare two objects using > operator, overload += and -= operators to add subtract with conceptual implementation use of specified operators. Use member functions.** 3. **W AP to create class COMPLEX and overload Binary +,- and \* and / operators on objects. define member functions to display objects.** 4. **W AP to create class COMPLEX and overload Binary +,- and \* and / operators using friend functions(operator).** 5. **Write a program to define object to int conversion for class distance using operator overloading.(consider all values converted to inches­ for assignment to integer variable)** | 4 |
| L5 | II | Polymorphism-Operator overloading | 1. **WAP to overload binary + operator to concatenate two objects of user defined class String.** 2. **WAP to overload binary +operator to add two objects of user defined Time class with data members HH MM SS.** 3. **W AP to show overloading of ++operator.** 4. **W AP to show overloading of ( )operator.** 5. **W AP to show overloading of [ ]operator.** | 4 |
| L6 | II | Inheritance | 1. **Write a program to define class person having data members name, address, city, mobile. Person class is having two subclasses faculty having data members (course, subjects, number of subjects) and staff having data members(shorthand speed and typing speed). Implement the class hierarchy and define appropriate constructors and methods.** 2. **W AP to demonstrate the working of virtual function. Define class shape having one data member of float type. Define two child classes square and circle. Implement calculate\_area () function to return area in both derived classes. Define class hierarchy and appropriate functions and constructors definition.** | 4 |
| L7 |  | Exception Handling | 1. **W AP to implement a class Stack that throws Overflow and Underflow Exceptions.** 2. **W AP to implement a class Customer that throws Negative Balance Exceptions.** | 2 |
| L8 | III | Template function and classes | 1. **W AP to implement template function that interchanges two data values.** 2. **W AP to implement a generic class Stack, add necessary constructors and other functions** | 2 |
| L9 | IV | File Handling and Stream Class Hierarchy | 1. **W AP that reads a text file.** 2. **W AP that writes paragraph to a text file.** 3. **W AP that writes Student records to a data file.** 4. **WAP that uses file pointer to display any arbitrary record from a pre existing file.** 5. **W AP that writes Student records to a data file. Read records based on name entered** | 4 |
| L10 | IV | Friend function and static | 1. **Friend function as a bridge between two classes rupee and dollar** 2. **Static data member and member function** | 2 |

**TEXT BOOKS:**

**T1:** Ashok N. Kamthane, “Object-Oriented Programming With Ansi And Turbo C++”, Pearson Education.

**T2:** A.R.Venugopal, Rajkumar, T. Ravishanker “Mastering C++”, TMH, 1997.

**T3:** E. Balguruswamy, “C++ ”, TMH Publication ISBN 0-07-462038-x .

**REFERENCES:**

**R1:**MaheshBhave, “Object Oriented Programming with C++”, Pearson Education

**R2:** Bjarne Stroustrup , “The C++ Programming Language”, Addison Welsley, 3rd Ed.

**R3:**Schildt Herbert, “C++: The Complete Reference”, Tata McGraw Hill, 4th Ed., 1999.

**R4:** R. Lafore, “Object Oriented Programming using C++”, Galgotia Publications, 2004

Dr. Gaurav Kumar Mr. Ashish Nayyar Prof. (Dr.) Sudhir Kumar Shrama

**Subject Faculty Programme Coordinator HoD-CS**