Assigned Date: 22nd September
Deadline: 6th October
Filename: CS_Yourname_RollNumber_TheoryAssignment II .pdf
CS_Yourname_RollNumber_LabAssignment II .pdf

Note:- *Lab assignments should consist of fully functional code with screenshots of your executed program. Theory assignments also consist of some practical questions there also requires full functional code with screenshots of your executed program. CR will make a folder named assignment inside that folder there will be two more folders named Lab and Theory everyone should upload their task in the respective folders.*

Theory Assignment II

1) Differentiate between function overriding and function overloading. Explain with a suitable example.
2) What does this pointer point to? What are the applications of this pointer? Explain with an example.
3) What do you mean by Function Template and Class Template? Illustrate with examples.
4) Define an abstract class. Differentiate between virtual and pure virtual functions.
5) What is Inheritance? Why is it important? Explain different modes of Inheritance in C++ with examples.
6) What does polymorphism mean in the C++ language? How is polymorphism achieved in compile time and run time? Explain with suitable examples.
7) How do you achieve reusability in C++? What are the advantages of reusability? Class D is derived from class B. Class D doesn't contain any data members on its own. Does class D require constructors? If yes, illustrate with an example.
8) What sort of ambiguity can be resolved by defining virtual base classes? Differentiate between a virtual base class and virtual function.
9) What is generic programming? How is it implemented in C++? Explain with an example.
10) Explain how Composition differs from Inheritance? Some programmers prefer not to use protected access because they believe it breaks the encapsulation of the base class. Discuss the relative merits of using protected access vs. using private access in base classes. In what case, a protected access specifier is generally used?
11) Define template. Differentiate function overloading & template overloading with examples.

Lab Assignment II

1) Define a virtual base class. Write a program to create a base class i.e. Employee. Derive two classes: administrative and academic from the base class. Derive another class HOD from these two derived classes. Create appropriate data members and member functions in each class to show the implementation of a virtual base class.
2) Create an abstract class shape. Derive three specific classes called triangle, rectangle, and circle. Using these four classes, design a program that will accept dimensions of a triangle, rectangle, or circle interactively, and display the area.

3) Define inheritance. Write a program to create a base class named “circle”. Derive another class “cylinder” from it. The program should calculate the total surface area of the circle and cylinder using the concept of inheritance. The circle should have data fields i.e. radius, area and method getRadius ( ) should take the value of radius from the user. The cylinder should have an additional data field i.e. height and additional method getHeight ( ) to assign the value of height. Similarly, area ( ) function calculates area, and display ( ) function prints the final area of the created objects i.e circle and cylinder. (Circle surface area=pi*r*r and cylinder surface area=2*pi*r*r+2*r*h)
4) Write a template function that returns the average of all the elements of an array. The arguments to the function should be the array name and the size of the array (type int). In main (), demonstrate the concept of the template with arrays of type int, long, double, and char.
5) Create a class Person and two derived classes Employee and student inherited from class Person. Now create a class Manager which is derived from two base classes Employee and Student. Show the use of a virtual base class.
6) A restaurant offers three different types of Pizza. And the price list is as follows.
    a) Mushroom Pizza Rs. 200
    b) Ham Pizza Rs. 150
    c) Cheese Pizza Ra. 100

    Define an abstract class called Pizza that has a pure virtual function called get_price(). This function, when derived in subclasses, should return the price of a specific Pizza. Create three classes inheriting from Pizza class to represent three different Pizzas. Use polymorphism to print the name of each Pizza and its price.
7) Write a program that illustrates the concept of Exception Handling in C++.
8) Write a function template sort(array[ ]) that sorts the given list of numbers. Write a program that inputs, sorts and outputs an int array and a float array.

```
// algorithm for sorting
for(m=0; m<size-1; m++)
{
for(n=m+1; n<size; n++)
{
if(array[m]>array[n]) // for ascending order
{
T temp;
temp = array[m];
array[m] = array[n];
array[n] = temp;
} } }// end of the sorting algorithm.
```