**Assigned Data:** ***August 13 Assigned date***
**Deadline:** ***August 30 Submission date***
**Filename:** ***CS_Yourname_RollNumber_TheoryAssignment I .pdf***
***CS_Yourname_RollNumber_LabAssignment I .pdf***

**Note:-** *Lab assignments should consist of full 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 1**

1) What is *Object Oriented Programming (OOP)*? Also explain the main features of OOP.
2) How do structures in C and C++ differ? Illustrate with an example.
3) Differentiate between *default constructor* and *parameterized constructor* with suitable example.
4) What is the purpose of *constructor* and *destructor*? What types of constructor are created by the compiler by default? Explain with an example.
5) Define Structure. How does it differ with Union? Illustrate with an example.
6) What is the difference between array and pointer variables? In what way are they similar?
7) What do you mean by function? Illustrate the difference between pass by value and pass by reference.
8) Define *operator overloading*. Write a program that illustrates the concept of *unary operator overloading* and *binary operator overloading*. Also state the use of *friend function* in case of *binary operator overloading*.
9) When will you make a function *inline*? Why? In what ways is inline function advantageous?
10) What are *static members* and *static functions*? Explain with appropriate syntax.
11) What is the purpose of the copy *constructor*? When is a copy constructor invoked? Explain with an example.
12) What is a *constructor*? Is it mandatory to use constructors in a class? Why? What are the types of constructors?
13) What is Inheritance? Why is it important? Briefly describe the different types of Inheritance with suitable examples?
14) What is an overriding function? How does it differ from function overloading?
15) What sort of ambiguity can be resolved by defining virtual base classes?
16) What is a *friend function*? What are the merits and demerits of using *friend functions*? Illustrate the use of friend function in operator overloading with suitable examples.
17) What is function overloading and operator overloading? What is the difference between declaration and definition of a function? Explain with examples.
18) What do you mean by *function overloading*? When do we use this concept?
19) When a class member is defined outside the class, which operator can be used to associate the function definition to a particular class? Define that operator with a suitable example.

**Lab Assignment I**

20) Write a C++ program that reads three coefficients a, b and c for quadratic equations and finds whether the solutions are in real or imaginary. (ax2 + bx + c = 0 if b2-4ac >=0 then the solutions are real.). Also find the roots.

21) Write a C++ program that reads ten positive numbers from the user and finally prints the largest of all. (use for loop, if condition and function.)

22) Write a C++ program using function (pass by reference) that calculates the values of x and y from the two linear equations.

    a) ax + by = m
    b) cx + dy = n

    The solutions are given as

    c) x = (md - bn)/(ad - cb)
    d) y = (na - mc)/(ad - cb)

    The function should take eight arguments and return nothing.

23) Do you remember a graph paper; plotting x-axis, y-axis and origin (0,0). A Point consists of two values; one is x-axis value and other one is y-axis value. Considering only a first quadrant and two such points, write a program that finds the distance between each other. Use class and objects.

24) Define a class called "Rectangle" with following attributes: length and breadth of data type Integer. Also include the following member functions:
    void setSize(int length, int breadth); // this function should set the value of length and breadth of the Rectangle.
    int getArea( ); // this function should return the area of the rectangle.
    int getPerimeter( ); // this function should return the perimeter of the rectangle.

    a) // formula to calculate: area = length * breadth.
    b) // formula to calculate: perimeter = 2 * (length + breadth).

    Write a driven program as well.

25)

| class Complex |
| --- |
| private:<br>int x;<br>int y; |
| public:<br>Complex( );<br>Complex( int x, int y); |

- Define a class called Complex.
- Define member functions that overload the following operators:
    - ❏ Minus unary operator. Returns void
    - ❏ Scalar multiplication. (you may use friend function) and returns Complex
    - ❏ Plus binary operator (+). => Returns Complex
    - ❏ Minus binary operator. => Returns Complex
    - ❏ += Shorthand operator. => Returns void
    - ❏ == Equals to operator. => Returns TRUE or FALSE

- ❏ Greater than operator. => Returns TRUE or FALSE
- ❏ ! = Not equal to operator. => Returns TRUE or FALSE
- ❏ Pre Increment operator. => Returns Complex
- ❏ Post Increment operator. => Returns Complex
- ❏ << Stream Insertion operator. (use friend function. Why?????) => Returns ostream&

Write a main( ) function to implement the above overloaded operators.

26) Create a base class called Shape. Use this class to store two double type values that could be used to compute the area of figures, Derive two specific classes called Triangle and Rectangle from the base Shape. Add to the base class, a member function set_data() to initialize base class data members and another member function display_data() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.

| class Shape |
|---|
| public side_one: double<br>public side_two: double |
| public shape(): Constructor<br>public shape(double, double): Constructor<br>public set_data(double, double): void<br>public virtual display_are(): void |
| class Rectangle: Shape<br>public display_area(): void |
| class Triangle: Shape<br>public display_area(): void |

27) Why do we need operator overloading in C++? Write a program in C++ to overload arithmetic assignment operator (+=) in order to add distance values of two objects obj1 and obj2 of class my_Distance. While considering the distance value of each object it must be in terms of meter and centimeter, i.e. int meter and float centimeter. Use two member functions get_Dist() and show_Dist() for accepting user inputs and displaying the result,respectively.

28) Create a class called **time** that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. Another member function should display it, in HH:MM:SS format. The final member function should add two objects of type time using operator overloading ( Overload '+' operator). Write a main() function to test your program.

29) Implement a **Time** class. Each object of this class will represent a specific time of day specifying hour, minute and second. Include a default and a parameterized constructor. Also overload the following operators for the same class.
   a) + operator to add two Time objects. (e.g. T3 = T1 + T2;)
   b) >> operator to read hour, minute and second from a user. (e.g. cin >> T4;)
   c) << operator to print our Time object in 00:00:00 format. (e.g. cour << T5;)

   Write a driven program (main program) to test those operators.