8. Programs using Operator Overloading & Type conversion

CO2: Map real-world objects into programming objects.

1. Create a class Number with three private members. Overload – operator to negate objects of Number class.

2. Create a class Number with one integer variable as private member. Overload ++ and -- operators' to increment and decrement objects of Number class respectively as both prefix and suffix notation.

3. Write a C++ program to overload < operator and display the smallest number out of two objects using friend function.

4. Create a class TIME that has separate integer data members for hours, minutes, and seconds. Include member functions to initialize it to default value, to a specified value, display (overload "<<" through a friend function), add two TIME objects (use operator overloading for "+"). Write a program to exercise this class in a suitable manner.

5. Create a class called STRING with a character array as data member. Write a non member overloaded function overload == operator to compare two objects of STRING class .

6. Create a class of COMPLEX numbers containing two float data type members x and y. Write a overloaded member function to overload +, - operator to Add and subtract two objects of COMPLEX class respectively. Write an overloaded non member function to overload *, / operator to multiply and divide two objects of COMPLEX class respectively.

7. Create a class called MATRIX with its sizes (M*N) – rows and cols as the member variables. Define all possible matrix operations (addition, subtraction and multiplication) for Matrix objects. Overload the +, - and * operators respectively.

8. Write a C++ program to evaluate the equation A= 3 x B, where A and B are objects of same class. Use friend function.

9. Create a class RationalNumber (fractions) with the following capabilities:

    a. Create a constructor that prevents a 0 denominator in a fraction, reduces or simplifies fractions that are not in reduced form and avoids negative denominators.
    b. Overload the addition, subtraction, multiplication and division operators for this class.

c. Overload the relational and equality operators for this class.

10. Create a class Distance with feet (integer) and inch (float) as data members. Write a program to prompt the user to enter meters (float) and convert to Distance object. Have necessary methods to get and display the Distance values**. [float to class – basic to class conversion]**

11. Create a class Distance with feet (integer) and inch (float) as data members. Have necessary methods to get and display the Distance values. Create an object of distance and convert to meters (float). **[class to float – class to basic conversion]**

12. Create a class Centimeter with one integer member that represents centimeter. Create another class Meter with one integer member that represents meter. Write a program to convert a meter to centimeter. **[class to class]**