Lab 8: Static Members and Operator Overloading

Task 1

We want to create a class of **Counter**, the object of which holds the count of anything. Also we want to keep track of total objects of class and serial No of each object. Write a class Counter. This class has three private data members:

count : An integer that holds a count value.
objCount: A static integer that holds the count of objects.
serialNo: An integer that holds the serial number of objects of Counter class.

1.1 Write a default constructor that initializes each data member of the class such that count with 0, obj-Count that increments the count of the objects and serialNo with the correct serial no (object 1 should have serial no 1, object2 should have serial no 2 and so on).
Counter( )

1.2 Write a constructor that accepts an argument int c that is assigned to the data member count. Also initialize objCount with the count of the objects and serialNo with the correct serial no.
Counter(int c)

1.3 Write destructor of the class which adjusts the count of Objects.

1.4 Create the getter-setter functions for the data members.
void setCount(int c)
int getCount()const
int getSerialNo()const
static int getObjCount()
static int IncrementObjCount()

1.5 Define operator = that assigns the value of count to the left hand operand. i.e. **c2=c1**

1.6 Define unary operator - that inverts the value of count data member for counter class and should allow the statements like **c1 =- c1;**

1.7 Write main function to test all the implemented functionality.

Task 2

Write a class **Distance** that holds distances or measurements expressed in feet and inches. This class has two private data members:
feet: An integer that holds the feet.
inches: An integer that holds the inches.

2.1 Write a constructor with default parameters that initializes each data member of the class. If inches are greater than equal to 12 then they must be appropriately converted to corresponding feet

2.2 Generate appropriate getter-setter functions for the data members.

bool setFeet(int f)
int getFeet()const
bool setInches(int i) It should ensure proper conversion to feet.
int getInches()const

2.3 Define an operator + that overloads the standard + math operator and allows one Distance object to be added to another.
Distance operator+(const Distance &obj)

2.4 Define an operator- function that overloads the standard - math operator and allows subtracting one Distance object from another.
Distance operator-(const Distance &obj)

2.7 Define an operator= function that overloads the = operator and assign one Distance object to another.
const Distance operator=(const Distance &obj)

2.8 Write main function to test all the implemented functionality.

Task 3

Write a class Matrix. This class has three private data members

rows: An integer that holds the numbers of rows for matrix

columns: An integer that holds the numbers of columns for matrix

matrix: An integer pointer to pointer that points to 2D array (rows x columns).

The class has the following member functions.

Matrix (int r, int c)	Constructs a new Matrix object to represent the given matrix
operator =	Overload = operator to assign values
operator ==	Overload == operator to compare whether matrices are equal or not
M2=M1+1	Overload + operator which takes integer as argument. It preforms scalar addition.
M2=M1-4	Overload - operator which takes integer as argument. It preforms scalar subtraction.
M3=M1+M2	Overload + operator which takes matrix object as argument. It adds two matrixes and returns the result.
M3=M1-M2	Overload - operator which takes matrix object as argument. It subtracts two matrixes and returns the result.