LAB TASK 09 - OVERLOADING-01

Problem 1:

Create a class **Money** that represents a money valuer (combination of dollars and cents). Overload the following operator.

Functions:

i) **Money() // Default Constructor**
Initializes dollars and cents to zero

ii) **Money(int dollar, int cents) // Parameterized Constructor**
Update dollar and cents accordingly

iii) **Money& operator= (const Money& right)**
Overload Assignment operator to assign Money objects to each other

iv) **Money& operator== (const Money& right)**
Overload Equal operator to check if Money objects are equal or not

v) **Money operator+ (const Money& right)**
Overload Addition operator to Add Money objects to each other

vi) **Money operator- (const Money& right)**
Overload Subtraction operator to Subtract smaller Money object from larger Money object.

vii) **Money& operator*= (int)**
Overload Multiplication operator to multiply Money object with an integer number

viii) **Money& operator/= (int)**
Overload Division operator to divide Money object with an integer number

ix) **~Money()**

Problem 2:

To get an idea about how does an inline function work, we create simple class called **Operations** having two member variables *oprand1(int)* and *operand2(int)* with the following functions:

i) **void getData()**
Takes input for operand1 and operand2

ii) **void sum()**
Performs sum operation on the operands for which input is taken in **getData** function

iii) **void product()**
Performs multiply operation on the operands for which input is taken in **getData** function

You are required to write only the prototypes of the above inline functions in the class, and define them outside the class i.e.
inline void operation :: getdata()

Inside main create an object of the class by using implicit default constructor (since we didn't design any constructor, so we still have access to the implicit default constructor) and call the functions.

Problem 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*integer	Overload * operator which takes integer as argument. It preforms scalar multiplication.
M2=M1/integer	Overload / operator which takes integer as argument. It preforms scalar division.
M3=M1*M2	Overload * operator which takes matrix object as argument. It multiplies two matrices and returns the result.
M3=M1/M2	Overload / operator which takes matrix object as argument. It divides two matrices and returns the result.