National University of Computer and Emerging Sciences

Lab Manual 06
Object Oriented Programming – CL1004

Course Instructor	Dr. Saira Karim
Lab Instructor(s)	Ms. Amna Zulfiqar Mr. Muhammad Adeel
Section	BCS-2B
Semester	Spring 2023
Date	14-03-2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 06– Class and Operator Overloading

Important Note:

- **You may find the syntax to accomplish these exercises from lecture demo.**
- **Add Necessary Comments in you code to justify your logic.**
- **Comment exercise number or statement at the start of your code**
- **Save each exercise in .cpp file with your roll no, ex and lab number e.g.**
- **22LXXXX_EX01_Lab01.cpp**
- **Place all of your exercises in a folder a Zip it (Do not create .rar file) with roll no and lab no. e.g. 22LXXX_Lab01.zip**
- **Make sure that the interface of your program is user friendly i.e. properly display information.**
- **Properly follow the coding standards.**

1. Exercise: **You have created the following Matrix Class in LAB 04 and LAB 05**

Create a class "Matrix" that represents a 2D matrix with private member rows, column, and ****data** (a pointer to a dynamically allocated 2D integer array).

Add public member functions

1. **Matrix(int rows, int cols): allocate memory for the matrix**
2. **~Matrix():** Destructor to deallocate memory for the matrix
3. **void input()** to initialize the matrix with user input
4. **void print()** to print the matrix
5. **void transpose()** to transpose the matrix.
6. **A Deep Copy Constructor**

Now add following new functions to you class and **submit Complete Matrix Class**

1. Overloaded assignment operator (a member function)
2. A non-member overloaded plus operator to add to matrixes.
3. A non-member overloaded plus operator to add a scalar to a matix.

Also Add Usage Example of each function of your class in main()

2. Exercise – Overloading Operators:

Create a class **Money** that represents a money value (combination of dollars and cents).

Private Member Variables: 1. int dollars 2. int cents

Functions:

1. **Money()** // Default Constructor
   Initializes dollars and cents to zero
2. **Money(int dollar, int cents)** // Parameterized Constructor
   Update dollar and cents accordingly
3. **int getDollars()**
   Return the value dollar
4. **int getCents()**
   Return the value of cent

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

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

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

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

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

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

11. **friend ostream& operator<<(ostream& os, const Money& m)**
    Overloaded stream insertion operator that allows us to output a Money object to an output stream

12. **friend istream& operator>>(istream& is, Money& m)**
    Overloaded stream extraction operator that allows us to extract a Money object from an input stream

13. **~Money()**