

Lab 5 – Money class object – 2021 SP CS-116

Define class object Money with the following characteristics:

Private attribute:

- int dollars;
- int cents

Define any Methods/functions/operators you need for this program.

The class object should also have the following overloaded operators:

–	<i>Subtraction operator.</i> If one <code>Money</code> object is subtracted from another, the operator should give the difference between the two Money. For example, if \$5.80 is subtracted from \$8.75, the result will be \$2.95.
+	<i>Addition operator.</i> If one <code>Money</code> object is added from another, the operator should give the sum between the two Money. For example, if \$5.80 is subtracted from \$8.75, the result will be \$14.55.
*	<i>Multiplication operator.</i> If one <code>Money</code> object is multiplying by x the operator should return x * Money object. For example, \$8.75 * 2 will return \$17.50.
==	<i>Equality operator.</i> If one <code>Money</code> object is compared from another (x == y), the operator should return true if x is same as y. For example, \$10.01 == 10.01 will return true.

Use the following test driver program:

```
int main()
{
    printMeFirst("Ron Sha", "CS-116 2021SP"); // change to your name
    Money m(2, 2); // $2.02

    cout << "Original value is " << m << endl;
    Money n = 50 % m;
    cout << "50% of value is " << n << endl;
    n = 10 * m;
    cout << "10 times value is " << n << endl;
    Money a (8, 75);
    Money b (5, 80);
    cout << "\nTesting arithmetic operators: \n" << endl;
    Money t;
    double c;
    t = a + b;
    cout << a << " + " << b << " = " << t << endl;
    t = a - b;
    cout << a << " - " << b << " = " << t << endl;
    cout << b << " - " << a << " = " << b-a << endl;
    c = 2;
    t = a * c;
    cout << a << " * " << c << " = " << t << endl;
    c = 2;
    t = a / c;
    cout << a << " / " << c << " = " << t << endl;
    cout << a << " - " << t << " = " << a-t << endl;

    if (a == b)
        cout << a << " = " << b << endl;
    else
        cout << a << " not equal to " << b << endl;
    return 0;
}
```

Your program should have similar output:

```
int main()
{
    printMeFirst("Ron Sha", "CS-116 2021SP"); // change to your name
    Money m(2, 2); // $2.02

    cout << "Original value is " << m << endl;
    Money n = 50 % m;
    cout << "50% of value is " << n << endl;
    n = 10 * m;
    cout << "10 times value is " << n << endl;
    Money a (8, 75);
    Money b (5, 80);
    cout << "\nTesting arithmetic operators: \n" << endl;
    Money t;
    double c;
    t = a + b;
    cout << a << " + " << b << " = " << t << endl;
    t = a - b;
    cout << a << " - " << b << " = " << t << endl;
    cout << b << " - " << a << " = " << b-a << endl;
    c = 2;
    t = a * c;
    cout << a << " * " << c << " = " << t << endl;
    c = 2;
    t = a / c;
    cout << a << " / " << c << " = " << t << endl;
    cout << a << " - " << t << " = " << a-t << endl;

    if (a == b)
        cout << a << " = " << b << endl;
    else
        cout << a << " not equal to " << b << endl;
    return 0;
}
```

```
Program written by: Ron Sha
Course Info: CS-116 2021SP
Date: Fri Mar 05 09:02:39 2021
```

```
Original value is $2.02
50% of value is $1.01
10 times value is $20.20
```

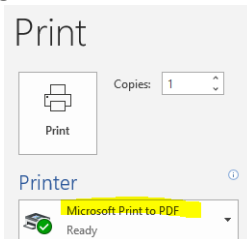
```
Testing arithmetic operators:
```

```
$8.75 + $5.80 = $14.55
$8.75 - $5.80 = $2.95
$5.80 - $8.75 = - $2.95
$8.75 * 2 = $17.50
$8.75 / 2 = $4.37
$8.75 - $4.37 = $4.38
$8.75 not equal to $5.80
```

Lab Submission: See the lab submission requirements published in canvas.

To submit your assignment in canvas, you must submit **TWO files (one pdf and one zip) as follows:**

1. **Attach pdf file** which contains source codes you have written and program output screenshot so I can easily read in one file. You can use a word editor to place all the required programs and screenshots, and then use 'Print' to 'Microsoft Print to PDF' to save to a pdf file



The **pdf file** **MUST** have the following sections (1. Program Description, 2. Program Source Code and 3. Program Output).

1. Program Description

- brief *description* of the purpose of the *program and*
- an explanation of what your software does and what problem it solves

2. Program Source Code

- Include all the source codes (program files) you have written for this lab (screenshots are ok)
- Your program must have adequate documentation for your source codes:
 - Program description – see above on Program Description
 - **You must put the following function headers for each function (including class member functions) the function header **MUST** be placed just above the function declaration in your source code). For Functions, it must have:**
 - *Function name: name of this function*
 - *Function description: the purpose of this function and how to use it*
 - *@param param_name and what the parameter/argument is used for*

- *@return what is returned from the function*
- Separate all .h (definition files) from .cpp (implementation files).
- All member functions MUST be defined outside the class definition. You CAN NOT define member function as inline (inside the class definition) declaration.
- Make sure your screenshots are readable (not too small)

Below is an example of source code

File: print_me_first_main.cpp (list the program file individually)

```

1 //print_me_first_main.cpp
2 #include <iostream>
3 #include <string>
4 #include <iomanip>
5
6 using namespace std;
7
8 /**
9  * @Purpose - this function print out the person who wrote the program,
10 * and date/time the program run.
11 * @parm - name - the author of the program
12 * @parm - courseInfo - the name of the course
13 * @return - none
14 * @author - Ron Sha
15 */
16
17 void PrintMeFirst(string name, string courseInfo)
18 {
19     cout << " Program written by: " << name << endl; // put your name here
20     cout << " Course info: " << courseInfo << endl;
21     time_t now = time(0); // current date/time based on current system
22     char* dt = ctime(&now); // convert now to string for
23     cout << " Date: " << dt << endl;
24 }

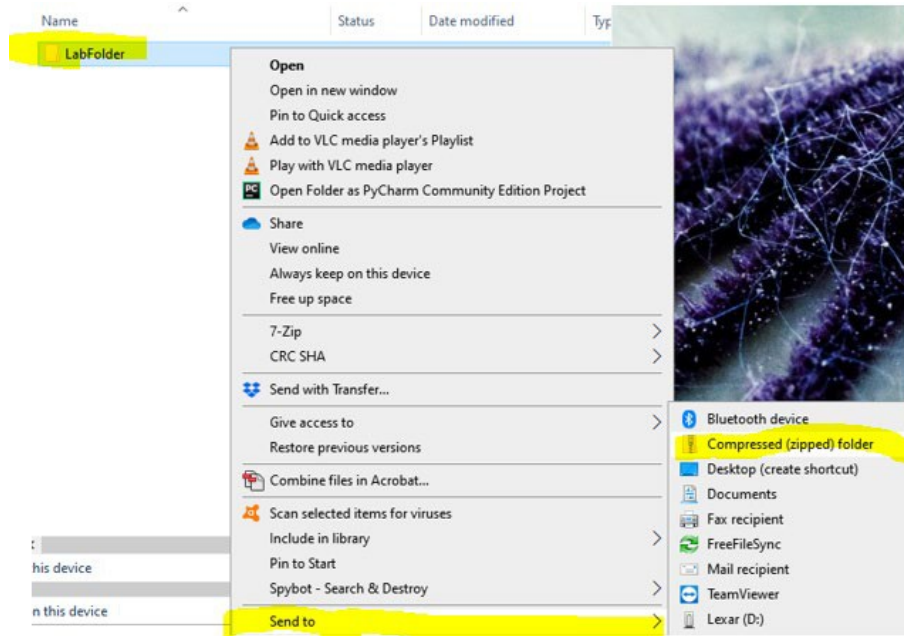
```

File: list other cpp files separately

3. Program Output

- Attach all the program outputs (screenshots)
 - Don't place Source code and program output side by side as it is not readable in screenshot
 - Make sure your screenshots are readable (not too small)
 - Your main program outputs MUST include your name printout (use the print_me_first function/program).
2. **Attach zip file** which contains all your source code (you can zip the folder) and functions. Even if you only have one source file, you MUST still do a zip file of the folder. I must be able to compile and run your program from all the source code programs after I unzip your zip file.

You should create a folder for each lab, and place all your programs, functions and all other files related to this lab in this lab folder. To submit the lab folder, you can use "Send to -> compress" in window file explorer to create a zip file of the folder.



3. Now you can upload both the **pdf file** and **zip file** separately as your assignment submission in canvas.