Lab 5 PDF

Name: Haichuan Wei

Lab: Lab 5

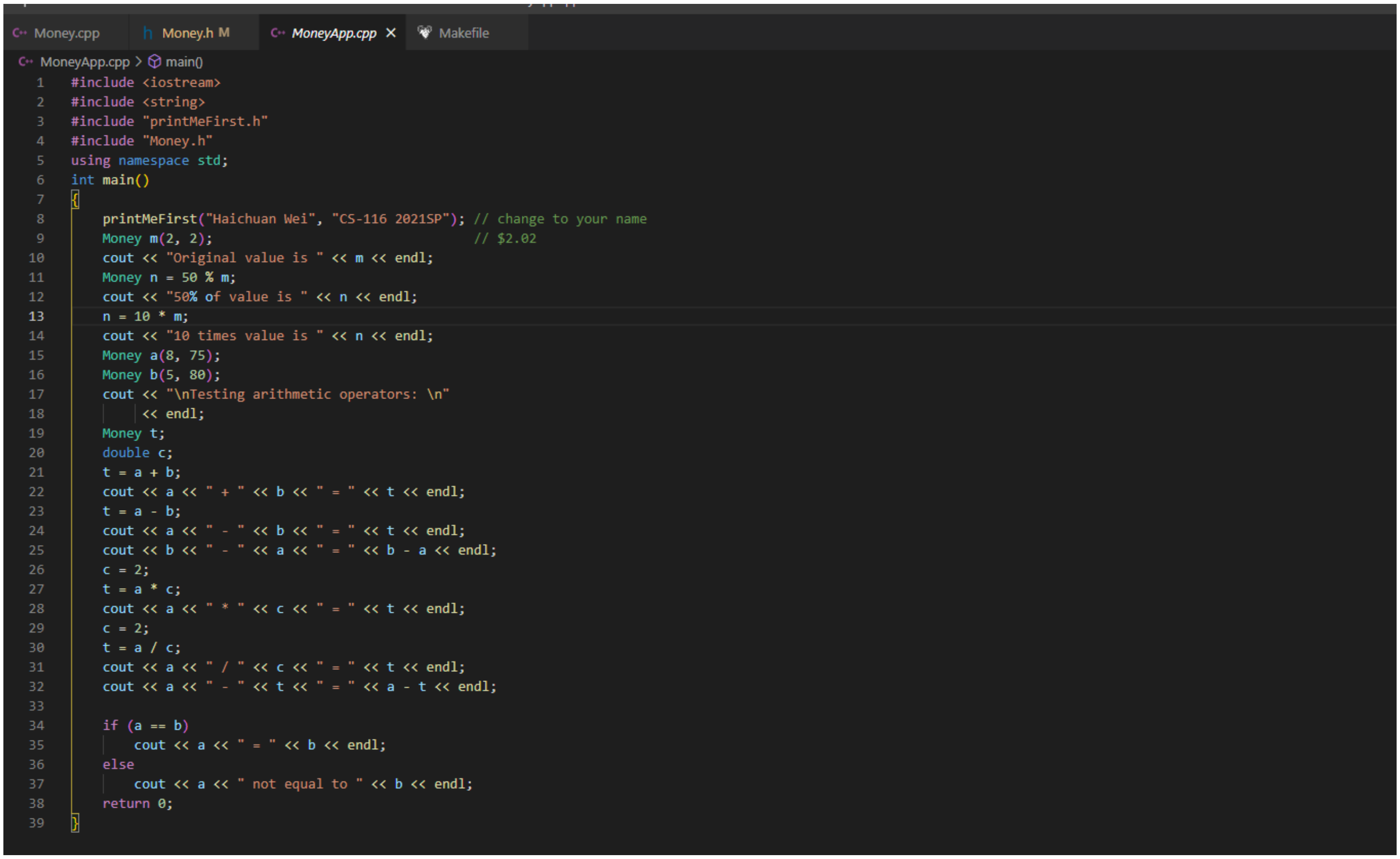
Date: 10/17/2021

Program Description:

This program uses overloaded operators to calculate numbers set by the users. The user passes the data from the driver file and they the Operators, (+ - * / % == <<) were all defied to behave differently to help calculate the data. The main purpose of this program was to learn how to use overladed operators.

Program Source Code:

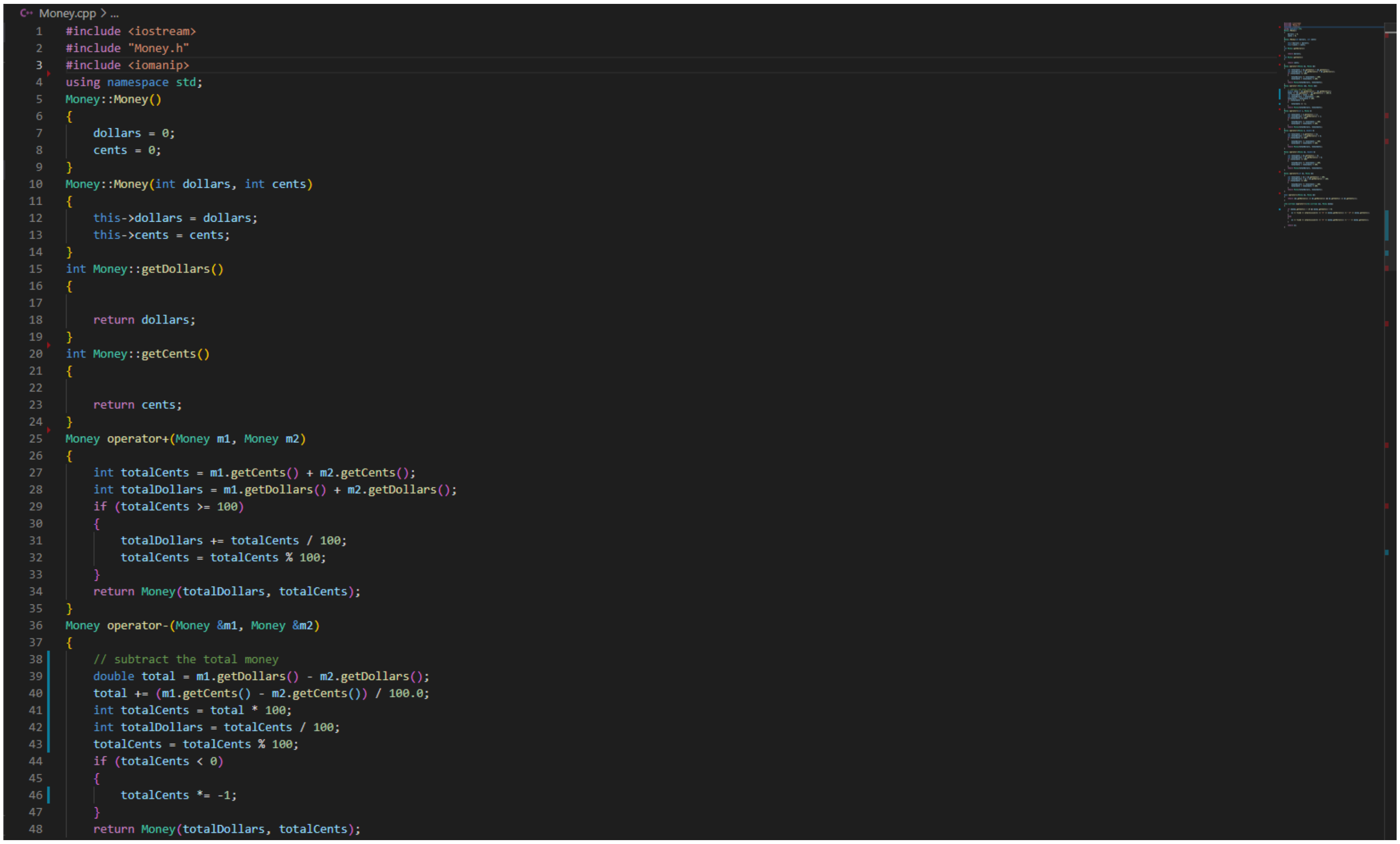
MoneyApp.cpp:



```cpp
#include <iostream>
#include <string>
#include "printMeFirst.h"
#include "Money.h"
using namespace std;
int main()
{
    printMeFirst("Haichuan Wei", "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;
}
```

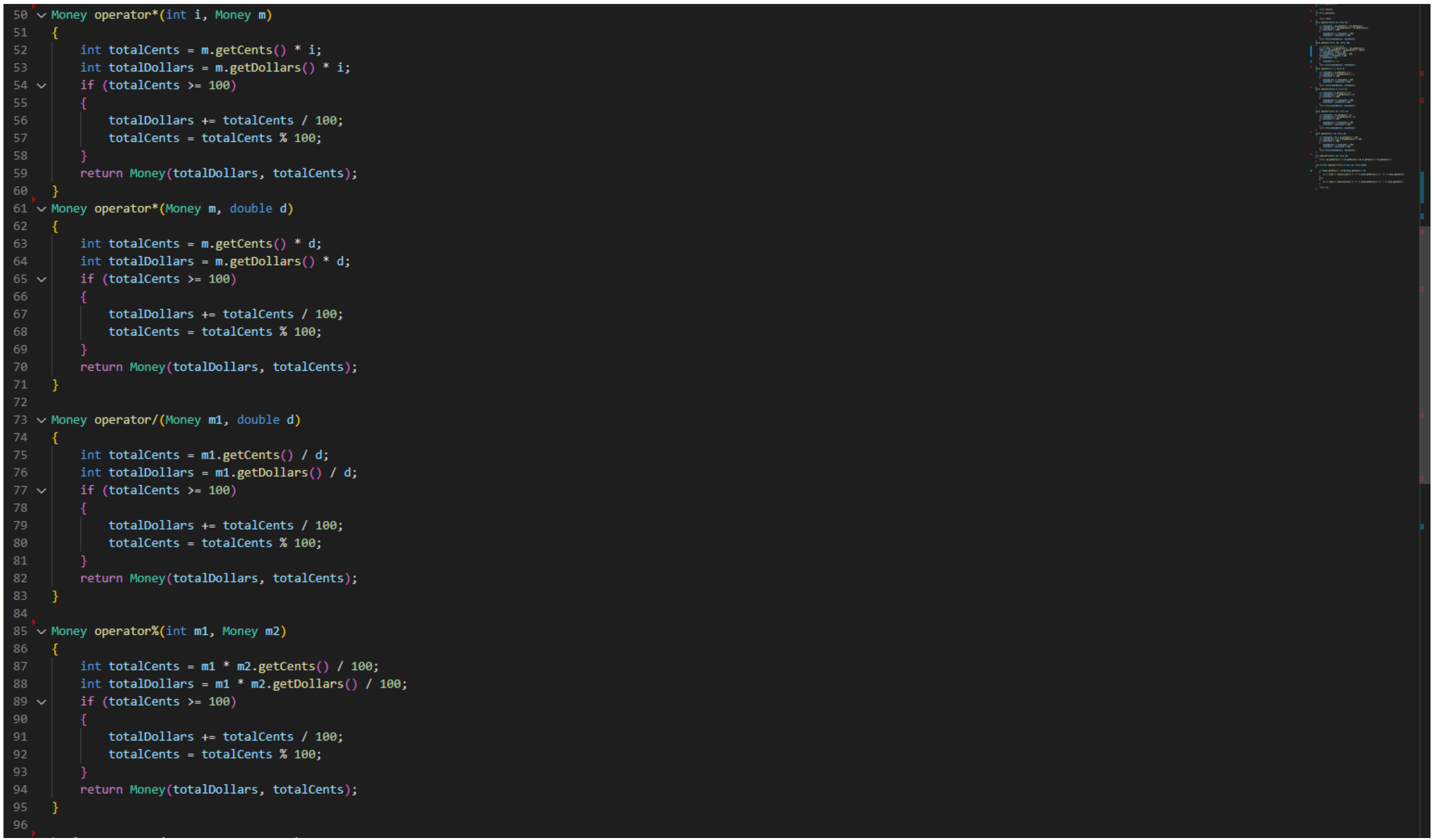
Lab 5 PDF

Money.cpp

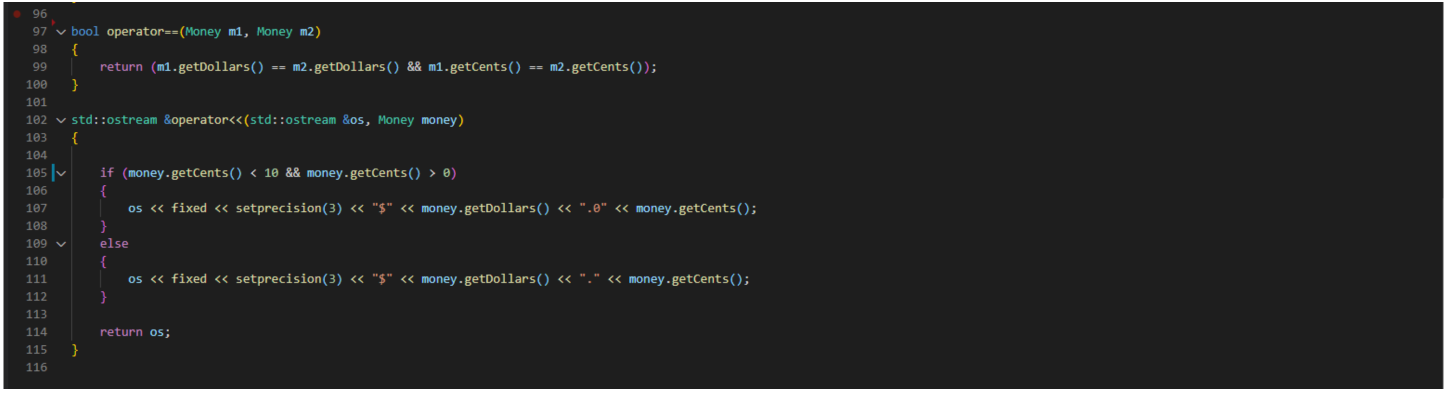


```cpp
#include <iostream>
#include "Money.h"
#include <iomanip>
using namespace std;
Money::Money()
{
    dollars = 0;
    cents = 0;
}
Money::Money(int dollars, int cents)
{
    this->dollars = dollars;
    this->cents = cents;
}
int Money::getDollars()
{

    return dollars;
}
int Money::getCents()
{

    return cents;
}
Money operator+(Money m1, Money m2)
{
    int totalCents = m1.getCents() + m2.getCents();
    int totalDollars = m1.getDollars() + m2.getDollars();
    if (totalCents >= 100)
    {
        totalDollars += totalCents / 100;
        totalCents = totalCents % 100;
    }
    return Money(totalDollars, totalCents);
}
Money operator-(Money &m1, Money &m2)
{
    // subtract the total money
    double total = m1.getDollars() - m2.getDollars();
    total += (m1.getCents() - m2.getCents()) / 100.0;
    int totalCents = total * 100;
    int totalDollars = totalCents / 100;
    totalCents = totalCents % 100;
    if (totalCents < 0)
    {
        totalCents *= -1;
    }
    return Money(totalDollars, totalCents);
```

Lab 5 PDF

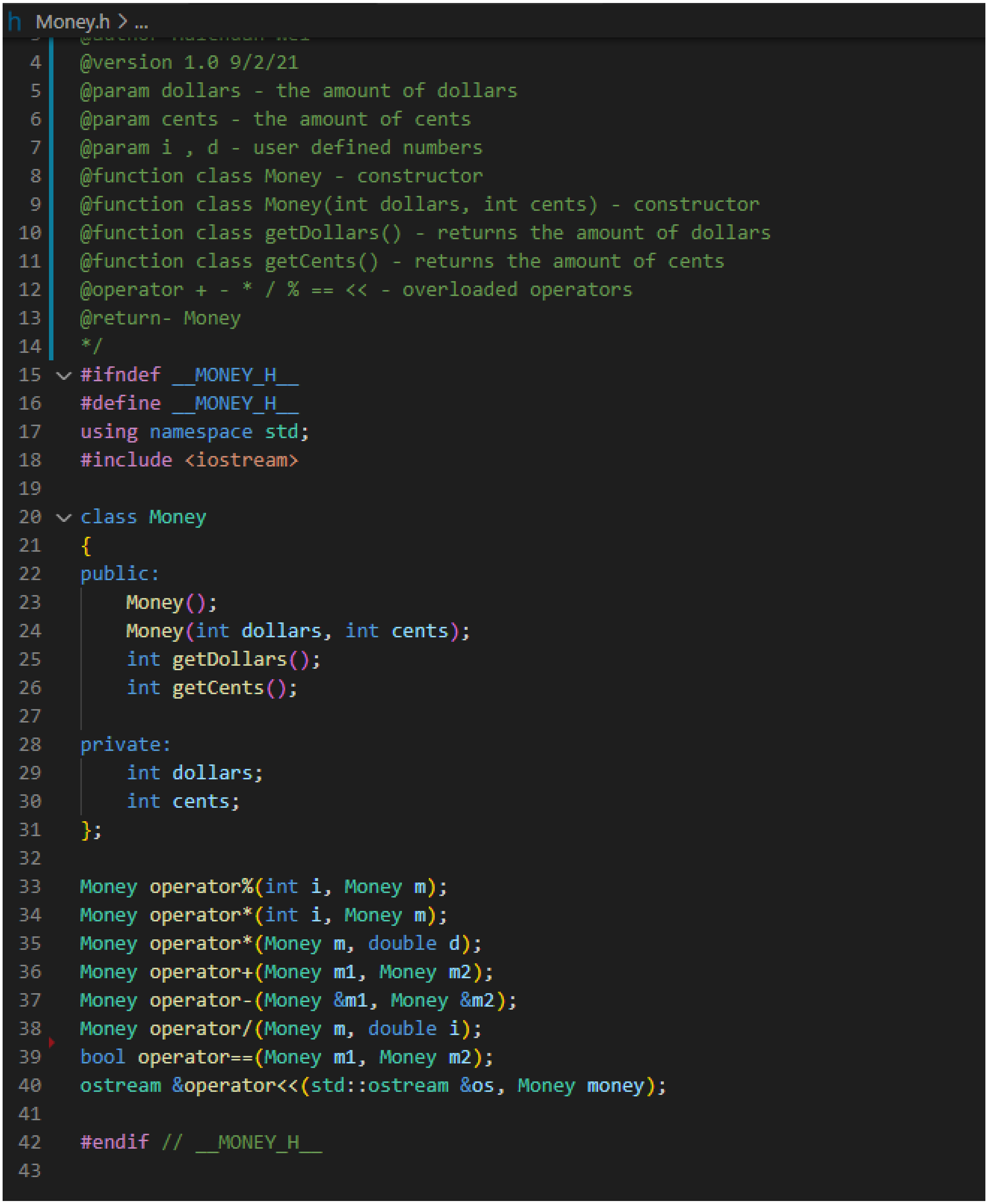


```cpp
Money operator*(int i, Money m)
{
    int totalCents = m.getCents() * i;
    int totalDollars = m.getDollars() * i;
    if (totalCents >= 100)
    {
        totalDollars += totalCents / 100;
        totalCents = totalCents % 100;
    }
    return Money(totalDollars, totalCents);
}
Money operator*(Money m, double d)
{
    int totalCents = m.getCents() * d;
    int totalDollars = m.getDollars() * d;
    if (totalCents >= 100)
    {
        totalDollars += totalCents / 100;
        totalCents = totalCents % 100;
    }
    return Money(totalDollars, totalCents);
}

Money operator/(Money m1, double d)
{
    int totalCents = m1.getCents() / d;
    int totalDollars = m1.getDollars() / d;
    if (totalCents >= 100)
    {
        totalDollars += totalCents / 100;
        totalCents = totalCents % 100;
    }
    return Money(totalDollars, totalCents);
}

Money operator%(int m1, Money m2)
{
    int totalCents = m1 * m2.getCents() / 100;
    int totalDollars = m1 * m2.getDollars() / 100;
    if (totalCents >= 100)
    {
        totalDollars += totalCents / 100;
        totalCents = totalCents % 100;
    }
    return Money(totalDollars, totalCents);
}

```



```cpp

bool operator==(Money m1, Money m2)
{
    return (m1.getDollars() == m2.getDollars() && m1.getCents() == m2.getCents());
}

std::ostream &operator<<(std::ostream &os, Money money)
{

    if (money.getCents() < 10 && money.getCents() > 0)
    {
        os << fixed << setprecision(3) << "$" << money.getDollars() << ".0" << money.getCents();
    }
    else
    {
        os << fixed << setprecision(3) << "$" << money.getDollars() << "." << money.getCents();
    }

    return os;
}

```

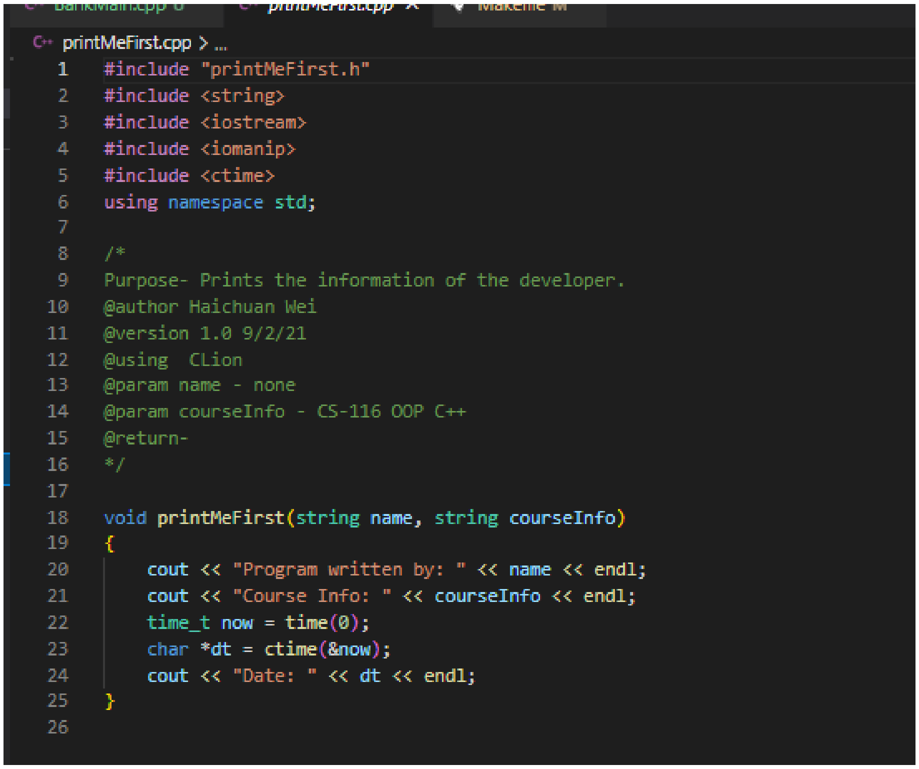
Money.h



```cpp
@author Haichuan Wei
@version 1.0 9/2/21
@param dollars - the amount of dollars
@param cents - the amount of cents
@param i , d - user defined numbers
@function class Money - constructor
@function class Money(int dollars, int cents) - constructor
@function class getDollars() - returns the amount of dollars
@function class getCents() - returns the amount of cents
@operator + - * / % == << - overloaded operators
@return- Money
*/
#ifndef __MONEY_H__
#define __MONEY_H__
using namespace std;
#include <iostream>

class Money
{
public:
    Money();
    Money(int dollars, int cents);
    int getDollars();
    int getCents();

private:
    int dollars;
    int cents;
};

Money operator%(int i, Money m);
Money operator*(int i, Money m);
Money operator*(Money m, double d);
Money operator+(Money m1, Money m2);
Money operator-(Money &m1, Money &m2);
Money operator/(Money m, double i);
bool operator==(Money m1, Money m2);
ostream &operator<<(std::ostream &os, Money money);

#endif // __MONEY_H__

```

Lab 5 PDF

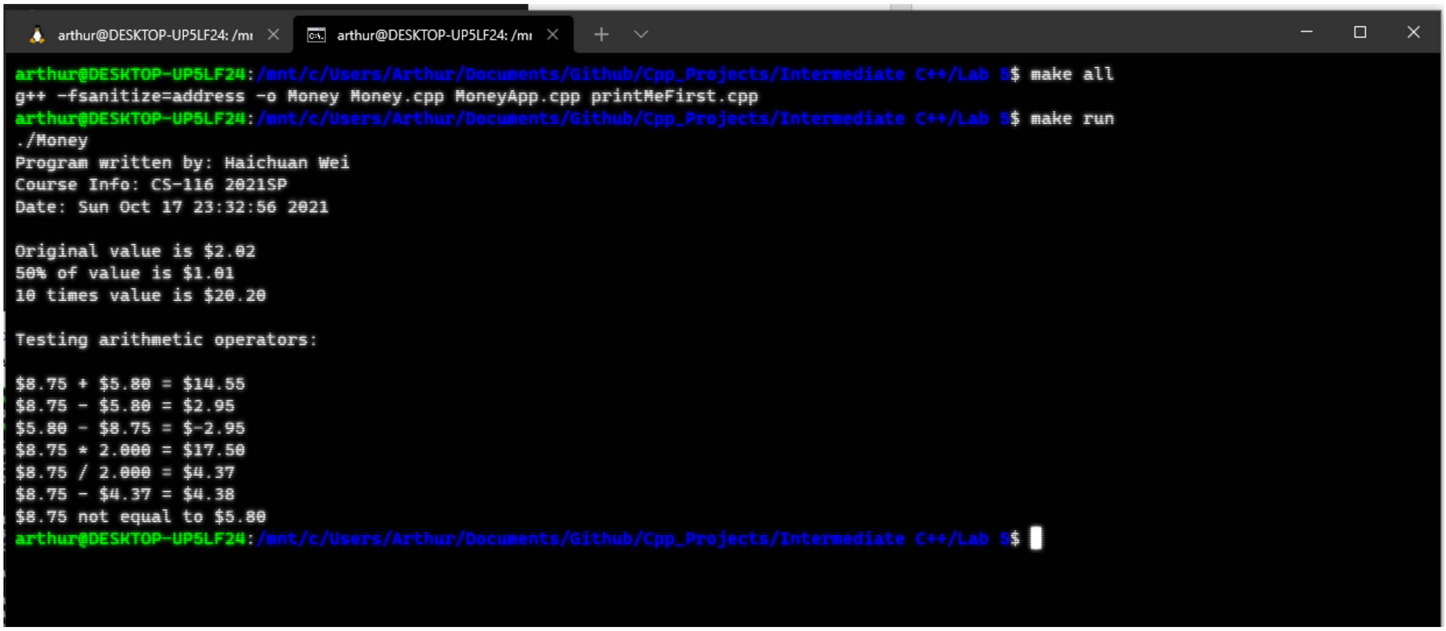
printMeFIrst.cpp



```cpp
#include "printMeFirst.h"
#include <string>
#include <iostream>
#include <iomanip>
#include <ctime>
using namespace std;

/*
Purpose- Prints the information of the developer.
@author Haichuan Wei
@version 1.0 9/2/21
@using  CLion
@param name - none
@param courseInfo - CS-116 OOP C++
@return-
*/

void printMeFirst(string name, string courseInfo)
{
    cout << "Program written by: " << name << endl;
    cout << "Course Info: " << courseInfo << endl;
    time_t now = time(0);
    char *dt = ctime(&now);
    cout << "Date: " << dt << endl;
}

```

Program Output

All values matched the example



```
arthur@DESKTOP-UP5LF24:/mnt/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 5$ make all
g++ -fsanitize=address -o Money Money.cpp MoneyApp.cpp printMeFirst.cpp
arthur@DESKTOP-UP5LF24:/mnt/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 5$ make run
./Money
Program written by: Haichuan Wei
Course Info: CS-116 2021SP
Date: Sun Oct 17 23:32:56 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 * 2.000 = $17.50
$8.75 / 2.000 = $4.37
$8.75 - $4.37 = $4.38
$8.75 not equal to $5.80
arthur@DESKTOP-UP5LF24:/mnt/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 5$
```