

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:

```
C++ MoneyApp.cpp > main()
1 #include <iostream>
2 #include <string>
3 #include "printMeFirst.h"
4 #include "Money.h"
5 using namespace std;
6 int main()
7 {
8     printMeFirst("Haichuan Wei", "CS-116 2021SP"); // change to your name
9     Money m(2, 2); // $2.02
10    cout << "Original value is " << m << endl;
11    Money n = 50 % m;
12    cout << "50% of value is " << n << endl;
13    n = 10 * m;
14    cout << "10 times value is " << n << endl;
15    Money a(8, 75);
16    Money b(5, 80);
17    cout << "\nTesting arithmetic operators: \n"
18        << endl;
19    Money t;
20    double c;
21    t = a + b;
22    cout << a << " + " << b << " = " << t << endl;
23    t = a - b;
24    cout << a << " - " << b << " = " << t << endl;
25    cout << b << " - " << a << " = " << b - a << endl;
26    c = 2;
27    t = a * c;
28    cout << a << " * " << c << " = " << t << endl;
29    c = 2;
30    t = a / c;
31    cout << a << " / " << c << " = " << t << endl;
32    cout << a << " - " << t << " = " << a - t << endl;
33
34    if (a == b)
35        cout << a << " = " << b << endl;
36    else
37        cout << a << " not equal to " << b << endl;
38    return 0;
39 }
```

Lab 5 PDF

Money.cpp

```
C> Money.cpp > ...
1  #include <iostream>
2  #include "Money.h"
3  #include <iomanip>
4  using namespace std;
5  Money::Money()
6  {
7      dollars = 0;
8      cents = 0;
9  }
10 Money::Money(int dollars, int cents)
11 {
12     this->dollars = dollars;
13     this->cents = cents;
14 }
15 int Money::getDollars()
16 {
17     return dollars;
18 }
19 int Money::getCents()
20 {
21     return cents;
22 }
23 Money operator+(Money m1, Money m2)
24 {
25     int totalCents = m1.getCents() + m2.getCents();
26     int totalDollars = m1.getDollars() + m2.getDollars();
27     if (totalCents >= 100)
28     {
29         totalDollars += totalCents / 100;
30         totalCents = totalCents % 100;
31     }
32     return Money(totalDollars, totalCents);
33 }
34 Money operator-(Money &m1, Money &m2)
35 {
36     // subtract the total money
37     double total = m1.getDollars() - m2.getDollars();
38     total += (m1.getCents() - m2.getCents()) / 100.0;
39     int totalCents = total * 100;
40     int totalDollars = totalCents / 100;
41     totalCents = totalCents % 100;
42     if (totalCents < 0)
43     {
44         totalCents *= -1;
45     }
46     return Money(totalDollars, totalCents);
47 }
48
```

Lab 5 PDF

```
50 Money operator*(int i, Money m)
51 {
52     int totalCents = m.getCents() * i;
53     int totalDollars = m.getDollars() * i;
54     if (totalCents >= 100)
55     {
56         totalDollars += totalCents / 100;
57         totalCents = totalCents % 100;
58     }
59     return Money(totalDollars, totalCents);
60 }
61 Money operator*(Money m, double d)
62 {
63     int totalCents = m.getCents() * d;
64     int totalDollars = m.getDollars() * d;
65     if (totalCents >= 100)
66     {
67         totalDollars += totalCents / 100;
68         totalCents = totalCents % 100;
69     }
70     return Money(totalDollars, totalCents);
71 }
72
73 Money operator/(Money m1, double d)
74 {
75     int totalCents = m1.getCents() / d;
76     int totalDollars = m1.getDollars() / d;
77     if (totalCents >= 100)
78     {
79         totalDollars += totalCents / 100;
80         totalCents = totalCents % 100;
81     }
82     return Money(totalDollars, totalCents);
83 }
84
85 Money operator%(int m1, Money m2)
86 {
87     int totalCents = m1 * m2.getCents() / 100;
88     int totalDollars = m1 * m2.getDollars() / 100;
89     if (totalCents >= 100)
90     {
91         totalDollars += totalCents / 100;
92         totalCents = totalCents % 100;
93     }
94     return Money(totalDollars, totalCents);
95 }
96
97 bool operator==(Money m1, Money m2)
98 {
99     return (m1.getDollars() == m2.getDollars() && m1.getCents() == m2.getCents());
100 }
101
102 std::ostream &operator<<(std::ostream &os, Money money)
103 {
104     if (money.getCents() < 10 && money.getCents() > 0)
105     {
106         os << fixed << setprecision(3) << "$" << money.getDollars() << ".0" << money.getCents();
107     }
108     else
109     {
110         os << fixed << setprecision(3) << "$" << money.getDollars() << "." << money.getCents();
111     }
112     return os;
113 }
114
115 }
```

Money.h

```

h Money.h > ...
3  @author: madison w...
4  @version 1.0 9/2/21
5  @param dollars - the amount of dollars
6  @param cents - the amount of cents
7  @param i , d - user defined numbers
8  @function class Money - constructor
9  @function class Money(int dollars, int cents) - constructor
10 @function class getDollars() - returns the amount of dollars
11 @function class getCents() - returns the amount of cents
12 @operator + - * / % == << - overloaded operators
13 @return- Money
14 */
15 #ifndef __MONEY_H__
16 #define __MONEY_H__
17 using namespace std;
18 #include <iostream>
19
20 class Money
21 {
22 public:
23     Money();
24     Money(int dollars, int cents);
25     int getDollars();
26     int getCents();
27
28 private:
29     int dollars;
30     int cents;
31 };
32
33 Money operator%(int i, Money m);
34 Money operator*(int i, Money m);
35 Money operator*(Money m, double d);
36 Money operator+(Money m1, Money m2);
37 Money operator-(Money &m1, Money &m2);
38 Money operator/(Money m, double i);
39 bool operator==(Money m1, Money m2);
40 ostream &operator<<(std::ostream &os, Money money);
41
42 #endif // __MONEY_H__
43

```

Lab 5 PDF

printMeFirst.cpp

```
C++ printMeFirst.cpp > ...
1  #include "printMeFirst.h"
2  #include <string>
3  #include <iostream>
4  #include <iomanip>
5  #include <ctime>
6  using namespace std;
7
8  /*
9   Purpose- Prints the information of the developer.
10  @author Haichuan Wei
11  @version 1.0 9/2/21
12  @using CLion
13  @param name - none
14  @param courseInfo - CS-116 OOP C++
15  @return-
16  */
17
18  void printMeFirst(string name, string courseInfo)
19  {
20      cout << "Program written by: " << name << endl;
21      cout << "Course Info: " << courseInfo << endl;
22      time_t now = time(0);
23      char *dt = ctime(&now);
24      cout << "Date: " << dt << endl;
25  }
26
```

Program Output

All values matched the example

```
arthur@DESKTOP-UP5LF24: /mi x arthur@DESKTOP-UP5LF24: /mi x + v
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:

$0.75 + $5.80 = $14.55
$0.75 - $5.80 = $2.95
$5.80 - $0.75 = $-2.95
$0.75 * 2.000 = $1.50
$0.75 / 2.000 = $0.37
$0.75 - $0.37 = $0.38
$0.75 not equal to $5.80
arthur@DESKTOP-UP5LF24: /mnt/c/Users/Arthur/Documents/Github/Cpp_Projects/Intermediate C++/Lab 5$
```