

Jeremy Scheuerman

Dr. Wang

Lab_6.2

6.5

Exercise 1

GLOBAL	Main	Main (inner 1)	Main (inner 2)	Area	Circumference
const PI	Float radius	Float area	Float radius	Float rad	Float length
Const RATE			Float circumference	Float answer	Float distance
Void findArea					
Void findCircumference					

Exercise 2

```
D:\downloads\Lab_export\5\Lab6.2\scope.exe
Main function outer block
globals and radius are active here

Main function first inner block
Globals, radius and area are active here

The radius = 12.00
The area = 0.00

Main function second inner block
Globals, radius, area, and circumference are active here

The radius = 10.00
The circumference = 823639652164023977670107675492352.00

Main function after all the calls
Globals and radius are active here

Process returned 0 (0x0)   execution time : 0.370 s
Press any key to continue.
```

Exercise 3

```
D:\downloads\Lab_export(3)\Lab6.2\scope.exe
Main function outer block
globals and radius are active here

Main function first inner block
Globals, radius and area are active here

AREA FUNCTION

Globals, rad , and answer are active here

The radius = 12.00
The area = 452.16

Main function second inner block
Globals, radius, area, and circumference are active here

CIRCUMFERENCE FUNCTION

Globals ,length and distance are active here

The radius = 10.00
The circumference = 62.80

Main function after all the calls
Globals and radius are active here

Process returned 0 (0x0)   execution time : 1.931 s
Press any key to continue.
```

Exercise 4

Radius will be 10

Radius will be 12

This is because of variable scoping also radius is re defined in the second block

Exercise 5/source code

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
// This program will demonstrate the scope rules.
```

```
// PLACE YOUR NAME HERE
```

```
const double PI = 3.14;
```

```
const double RATE = 0.25;
```

```
void findArea(float, float&);
```

```
void findCircumference(float, float&);
```

```
int main()
```

```
{
```

```
    cout << fixed << showpoint << setprecision(2);
```

```
    float radius = 12;
```

```
    cout << " Main function outer block" << endl;
```

```
    cout << " globals and radius are active here" << endl << endl;
```

```
    {
```

```
        float area;
```

```
        cout << "Main function first inner block" << endl;
```

```
cout << "Globals, radius and area are active here" << endl << endl;

// Fill in the code to call findArea here

findArea(radius,area);

cout << "The radius = " << radius << endl;

cout << "The area = " << area << endl << endl;

}

{

float radius = 10;

float circumference;

cout << "Main function second inner block" << endl;

cout << "Globals, radius, area, and circumference are active here" << endl <<

endl;

// Fill in the code to call findCircumference here

findCircumference(radius,circumference);

cout << "The radius = " << radius << endl;

cout << "The circumference = " << circumference << endl << endl;

}
```

```

    cout << "Main function after all the calls" << endl;

    cout << "Globals and radius are active here" << endl << endl;


    return 0;

}


// *****

//    findArea

//

//    task:    This function finds the area of a circle given its radius

//    data in: radius of a circle

//    data out: answer (which alters the corresponding actual parameter)

//

//    *****

void findArea(float rad, float& answer)
{

    cout << "AREA FUNCTION" << endl << endl;

    cout << "Globals, rad , and answer are active here" << endl << endl;


    // FILL in the code, given that parameter rad contains the radius, that

    // will find the area to be stored in answer

```

```

        answer=PI*(rad*rad);

    }

//

*****

*****

//    findCircumference

//

//    task:    This function finds the circumference of a circle given its radius

//    data in: radius of a circle

//    data out: distance (which alters the corresponding actual parameter)

//

//

*****

*****

void findCircumference(float length, float& distance)

{

    cout << "CIRCUMFERENCE FUNCTION" << endl << endl;

    cout << "Globals ,length and distance are active here" << endl << endl;

    // FILL in the code, given that parameter length contains the radius,

    // that will find the circumference to be stored in distance

```

```
distance=2*length*PI;
```

```
}
```

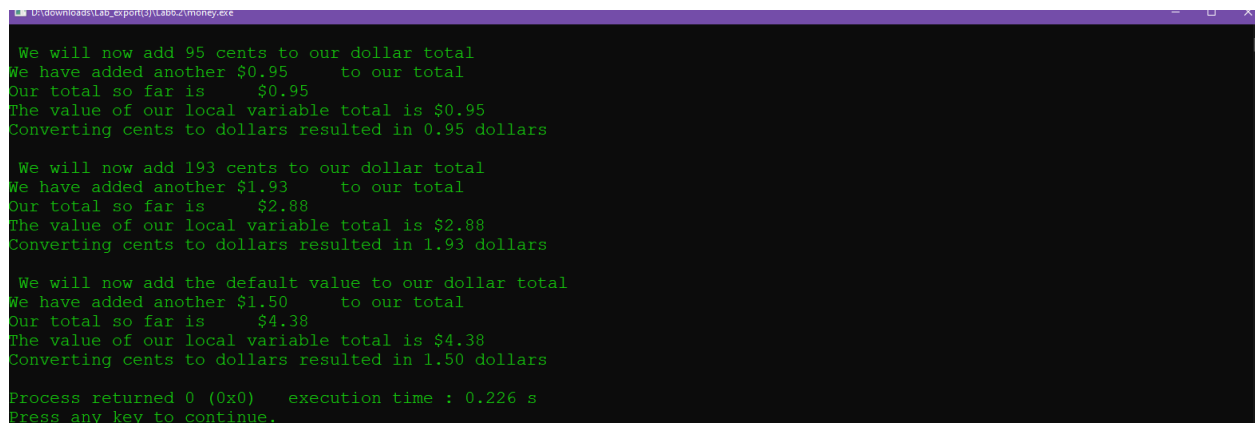
6.6

Exercise 1

\$1.93

However right now it just prints 0 because things haven't been defined

Exercise 2



```
U:\downloads\lab_export(3)\Lab2\money.exe
We will now add 95 cents to our dollar total
We have added another $0.95 to our total
Our total so far is $0.95
The value of our local variable total is $0.95
Converting cents to dollars resulted in 0.95 dollars

We will now add 193 cents to our dollar total
We have added another $1.93 to our total
Our total so far is $2.88
The value of our local variable total is $2.88
Converting cents to dollars resulted in 1.93 dollars

We will now add the default value to our dollar total
We have added another $1.50 to our total
Our total so far is $4.38
The value of our local variable total is $4.38
Converting cents to dollars resulted in 1.50 dollars

Process returned 0 (0x0) execution time : 0.226 s
Press any key to continue.
```

Source Code

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
// PLACE YOUR NAME HERE
```

```
void normalizeMoney(float& dollars, int cents = 150);
```

```
// This function takes cents as an integer and converts it to dollars  
// and cents. The default value for cents is 150 which is converted  
// to 1.50 and stored in dollars
```

```
int main()  
{  
    int cents;  
    float dollars;  
  
    cout << setprecision(2) << fixed << showpoint;  
  
    cents = 95;  
    cout << "\n We will now add 95 cents to our dollar total\n";  
  
    //    Fill in the code to call normalizeMoney to add 95 cents  
    normalizeMoney(dollars,cents);  
  
    cout << "Converting cents to dollars resulted in " << dollars << " dollars\n";  
  
    cout << "\n We will now add 193 cents to our dollar total\n";  
  
    // Fill in the code to call normalizeMoney to add 193 cents  
    cents=193;
```



```
normalizeMoney(dollars,cents);
```

```
cout << "Converting cents to dollars resulted in " << dollars << " dollars\n";
```

```
cout << "\n We will now add the default value to our dollar total\n";
```

```
// Fill in the code to call normalizeMoney to add the default value of cents
```

```
normalizeMoney(dollars);
```

```
cout << "Converting cents to dollars resulted in " << dollars << " dollars\n";
```

```
return 0;
```

```
}
```

```
//*****
```

```
***
```

```
//      normalizeMoney
```

```
//
```

```
//      task:    This function is given a value in cents. It will convert cents
```

```
//              to dollars and cents which is stored in a local variable called
```

```
//              total which is sent back to the calling function through the
```

```
//              parameter dollars. It will keep a running total of all the money
```

```
//              processed in a local static variable called sum.
```

```
//
//      data in: cents which is an integer
//      data out: dollars (which alters the corresponding actual parameter)
//
//*****

*****

void normalizeMoney(float& dollars, int cents)
{
    static float total = 0;

    // Fill in the definition of sum as a static local variable

    static float sum = 0.0;

    // Fill in the code to convert cents to dollars

    dollars=cents*.01;

    total = total + dollars;

    sum += dollars;

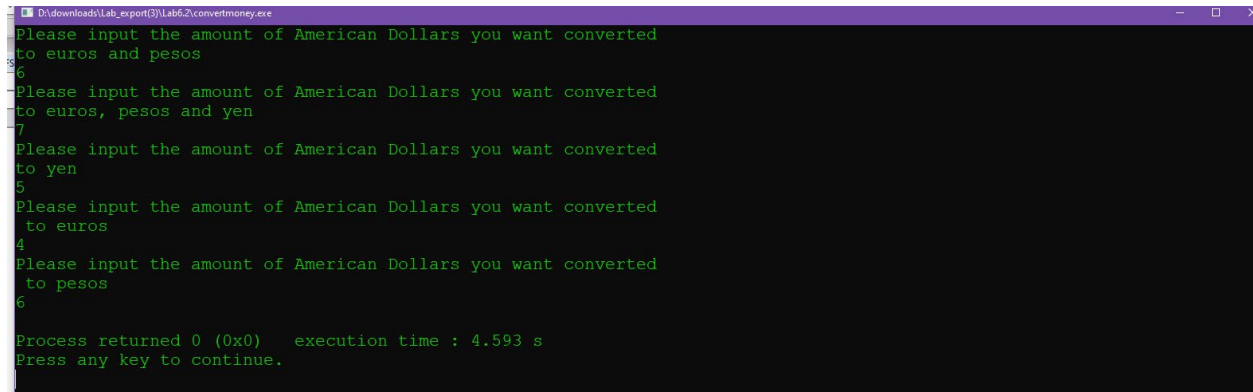
    cout << "We have added another $" << dollars << "      to our total" << endl;

    cout << "Our total so far is $" << sum << endl;
}
```

```
    cout << "The value of our local variable total is $" << total << endl;
}
```

6.7

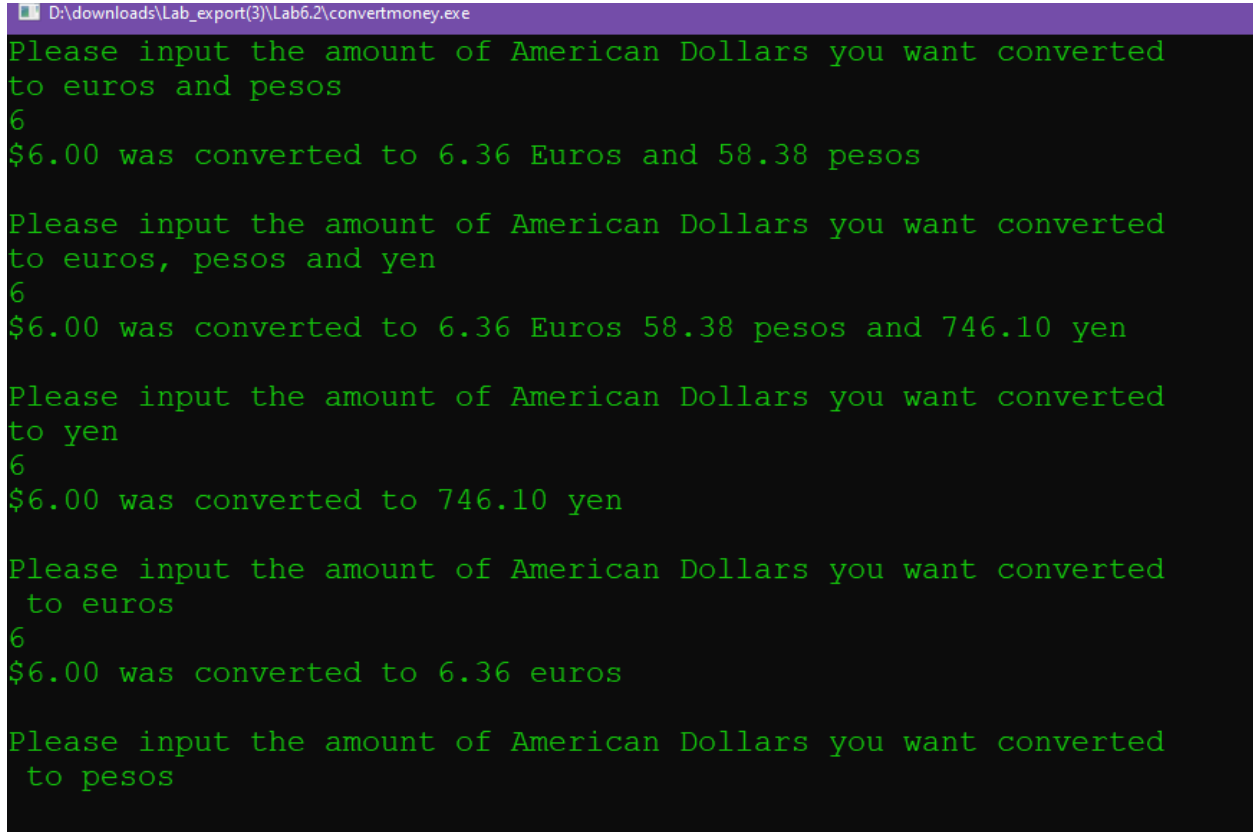
Exercise 1



```
D:\downloads\Lab_export(3)\Lab6.2\convertmoney.exe
Please input the amount of American Dollars you want converted
to euros and pesos
6
Please input the amount of American Dollars you want converted
to euros, pesos and yen
6
Please input the amount of American Dollars you want converted
to yen
6
Please input the amount of American Dollars you want converted
to euros
6
Please input the amount of American Dollars you want converted
to pesos
6
Process returned 0 (0x0)   execution time : 4.593 s
Press any key to continue.
```

I assume scope and variable passing was not done correctly

Exercise 2



```
D:\downloads\Lab_export(3)\Lab6.2\convertmoney.exe
Please input the amount of American Dollars you want converted
to euros and pesos
6
$6.00 was converted to 6.36 Euros and 58.38 pesos

Please input the amount of American Dollars you want converted
to euros, pesos and yen
6
$6.00 was converted to 6.36 Euros 58.38 pesos and 746.10 yen

Please input the amount of American Dollars you want converted
to yen
6
$6.00 was converted to 746.10 yen

Please input the amount of American Dollars you want converted
to euros
6
$6.00 was converted to 6.36 euros

Please input the amount of American Dollars you want converted
to pesos
```

```
D:\downloads\Lab_export(3)\Lab6.2\convertmoney.exe
Please input the amount of American Dollars you want converted
to euros and pesos
4.56
$4.56 was converted to 4.83 Euros and 44.37 pesos

Please input the amount of American Dollars you want converted
to euros, pesos and yen
67.8
$67.80 was converted to 71.87 Euros 659.69 pesos and 8430.93 yen

Please input the amount of American Dollars you want converted
to yen
56.3
$56.30 was converted to 7000.90 yen

Please input the amount of American Dollars you want converted
to euros
23.6
$23.60 was converted to 25.02 euros

Please input the amount of American Dollars you want converted
to pesos
56.43
$56.43 was converted to 549.06 pesos

Process returned 0 (0x0)    execution time : 14.335 s
Press any key to continue.
```

Source code

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
// This program will input American money and convert it to foreign currency
```

```
// PLACE YOUR NAME HERE
```

```
// Prototypes of the functions

void convertMulti(float dollars, float& euros, float& pesos);

void convertMulti(float dollars, float& euros, float& pesos, float& yen);

float convertToYen(float dollars);

float convertToEuros(float dollars);

float convertToPesos(float dollars);


const float TOEUROS=1.06;

const float TOPESOS=9.73;

const float TOYEN=124.35;


int main()

{

    float dollars;

    float euros;

    float pesos;

    float yen;


    cout << fixed << showpoint << setprecision(2);


    cout << "Please input the amount of American Dollars you want converted "

        << endl;

    cout << "to euros and pesos" << endl;
```

```
cin >> dollars;
```

```
// Fill in the code to call convertMulti with parameters dollars, euros, and pesos
```

```
convertMulti(dollars,euros,pesos);
```

```
// Fill in the code to output the value of those dollars converted to both euros
```

```
// and pesos
```

```
cout << "Please input the amount of American Dollars you want converted\n";
```

```
cout << "to euros, pesos and yen" << endl;
```

```
cin >> dollars;
```

```
// Fill in the code to call convertMulti with parameters dollars, euros, pesos and yen
```

```
convertMulti(dollars,euros,pesos,yen);
```

```
// Fill in the code to output the value of those dollars converted to euros,
```

```
// pesos and yen
```

```
cout << "Please input the amount of American Dollars you want converted\n";
```

```
cout << "to yen" << endl;
```

```
cin >> dollars;
```

```
// Fill in the code to call convertToYen
```

```
convertToYen(dollars);
```

```
// Fill in the code to output the value of those dollars converted to yen
```

```
cout << "Please input the amount of American Dollars you want converted\n";
```

```
cout << " to euros" << endl;
```

```
cin >> dollars;
```

```
// Fill in the code to call convert ToEuros
```

```
convertToEuros(dollars);
```

```
// Fill in the code to output the value of those dollars converted to euros
```

```
cout << "Please input the amount of American Dollars you want converted\n";
```

```
cout << " to pesos " << endl;
```

```
cin >> dollars;
```

```
// Fill in the code to call convertToPesos
```

```
convertToPesos(dollars);
```

```
// Fill in the code to output the value of those dollars converted to pesos
```

```
return 0;
```

```
}
```

```
// All of the functions are stubs that just serve to test the functions
```

```
// Replace with code that will cause the functions to execute properly
```

```
//
```

```
*****
```

```
**
```

```
//    convertMulti
```

```
//
```

```
//    task:    This function takes a dollar value and converts it to euros
```

```
//            and pesos
```

```
//    data in: dollars
```

```
//    data out: euros and pesos
```

```
//
```

```
//
```

```
*****
```

```
*
```

```
void convertMulti(float dollars, float& euros, float& pesos)
```

```
{
```

```
    pesos=dollars*TOPEOS;
```

```
    euros=dollars*TOEUROS;
```



```

cout<< "$"<< dollars

    << " was converted to " <<euros<< " Euros and "<<pesos<< " pesos "<< endl << endl;

}

// *****

//      convertMulti

//

//      task:   This function takes a dollar value and converts it to euros

//              pesos and yen

//      data in: dollars

//      data out: euros pesos yen

//

// *****

```

```

void convertMulti

```

6.8

Option 1

```
D:\Documents\Programming\lab_6.8_option_1.exe
Please input
1 Convert miles to kilometers
2 convert kilometers to miles
3 Quit
1
Please input the miles to be converted
120
120 miles = 193.2 kilometers
Please input
1 Convert miles to kilometers
2 convert kilometers to miles
3 Quit
2
Please input the kilometers to be converted
235
235 kilometers = 145.935 miles
Please input
1 Convert miles to kilometers
2 convert kilometers to miles
3 Quit
3

Process returned 0 (0x0)    execution time : 12.777 s
Press any key to continue.
```

Source Code

```
#include <iostream>

using namespace std;

const float TOKM=.621;

const float TOMI=1.61;

float kmtomi();

float mitokm();

float kmtomi(float kilo)

{
```

```
float miles=0;

miles=kilo*TOKM;

return miles;

}
```

```
float mitokm(float miles)

{

float kilo=0;

kilo=miles*TOMI;

return kilo;

}
```

```
int main ()

{

int choice=4;

float miles=0;

float kilo=0;

while (choice!=3)

{

cout<<"Please input"<<endl<<"1 Convert miles to kilometers"<<endl<<"2 convert

kilometers to miles"<<endl<<"3 Quit"<<endl;

cin>>choice;

if(choice==1)

{
```

```

        cout<<"Please input the miles to be converted"<<endl;

        cin>>miles;

        kilo=mitokm(miles);

        cout<<miles<<" miles = "<<kilo<<" kilometers"<<endl;

    }

    else if(choice==2)

    {

        cout<<"Please input the kilometers to be converted"<<endl;

        cin>>kilo;

        miles=kmtomi(kilo);

        cout<<kilo<<" kilometers = "<<miles<<" miles"<<endl;

    }

    else if (choice==3)

    {

        break;

    }

}

return 0;

}

```