[Ravi Patel] Instructor: Dr. Thamira Hindo
[CPSC 230]

Chapter 5- Homework…. (15 pts)

Note: Submit your homework document in the inbox (chapter 5 HW).

Part 1: (10pts.) Write the functions bellow:

1. Write a program that reads in a length in feet and inches and output the equivalent length in meters and centimeters. Use at least three functions: one for input, one or more for calculating and one for output. Include a loop that lets the user repeat this computation for new input values until the user says he or she wants to end the program. Use call by reference for the input, output and calculate to pass the data between the functions.

```
//CPSC 230 RAVI PATEL FT TO M CONVERTER CH 5 HW Q1
#include <iostream>
using namespace std;

void input(double &ft, double &in, double &meters, double &cm); //declare input
function
double change(double &ft, double &in, double &meters, double &cm); //declare
conversion function
void output(double ft, double in, double meters, double cm); //declare output
function

int main() //beginning of main program
{
     double ft, in, meters, cm; //assign values as double
     char c; //assign choice value as char

     do { //do this
               input(ft, in, meters, cm); //input function
               change(ft, in, meters, cm); //conversion function
               cm = in * 2.54; //added cm conversion
               output(ft, in, meters, cm); //output function

               cout<<"Would you like to do more calculations? 'y' for yes: ";
               cin>>c;

          } while(c == 'y' || c == 'Y'); //while the user wants to repeat
calculations

     return 0; //catch all
}

void input(double &ft, double &in, double &meters, double &cm) //input function
using reference
     {
     cout << "How many feet? : "; //ask user input
     cin >> ft; //assign user input
     cout <<"How many inches? : "; //ask user input
     cin >> in; //assign user input
```

```
     }
double change(double &ft, double &in, double &meters , double &cm) //conversion
function using reference
     {
     meters = ft * 0.3048; //.3048 meters in one foot, calculation here
     return meters; //return calculation
     }
void output(double ft, double in, double meters, double cm) //output function to
display
     {
     cout <<ft << " ft " << "and " << in << " in is equivalent to " <<meters<< "
meters " << "and " << cm << " cm\n";
     }

//SAMPLE DISPLAY OUTPUT:
//How many feet? : 5
//How many inches? : 12
//5 ft and 12 in is equivalent to 1.524 meters and 30.48 cm
//Would you like to do more calculations? 'y' for yes: y
//How many feet? : 6
//How many inches? : 4
//6 ft and 4 in is equivalent to 1.8288 meters and 10.16 cm
//Would you like to do more calculations? 'y' for yes: n
```

2. Write a program that tells what coins to give out for any amount of change from 1 cent to 99 cents. For example, if the amount is 86 cents, the output would be something like the following: 86 cents can be given as 3 quarter(s) 1 dime(s) and 1 penny (pennies). Write a function compute_coin and use the call by reference as in following declaration:

   void compute_coin(int change, int& no_quarters, int& no_dimes, int& no_nickles, int& no_cents);

Include a loop that lets the user repeat this computation for new input values until the user says he or she wants to end the program.

```
//CPSC 230 RAVI PATEL CH5 HW Q2
#include <iostream>
#include <cmath>
using namespace std;

void compute_coin(int change, int& no_quarters, int& no_cents) {
no_quarters = no_cents/ change;
no_cents = no_cents - (change*no_quarters);
}

int main() {

int amount = 0, quarters = 0, dimes = 0, nickels = 0, cents = 0;
char c;

do{
     cout << "Enter an amount to evaluate (cents): " ; //get user input
     cin>>amount; //assign user input

     if ( amount < 1 || amount > 99 ) {
     cout << "ERROR! Can only evaluate between 1 and 99 cents. Try again.\n";
     }

     else {

     cout<<amount<<" cents can be given as "; //display format

     compute_coin(25, quarters, amount); //pass thru compute_coin
     cout << quarters<< " quarter(s)"; //display quarters

     compute_coin (10, dimes, amount); //pass thru compute_coin
     cout << " and "<< dimes<< " dime(s)"; //display dimes

     compute_coin (5, nickels, amount); //pass thru compute_coin
     cout << " and "<< nickels<< " nickel(s)"; //display nickels

     compute_coin (1, cents, amount); //pass thru compute_coin
     cout << " and "<< cents<< " cent(s)"<<endl; //display cents

     cout << "Would you like to do another evaluation? 'y' for yes: "; //user
choice, repeat?
     cin >> c; //assign user choice
```

```
    }
} while((c == 'y' || c == 'Y')); //while choice is yes, do above
return (0); //catch all
}
```

```
//SAMPLE OUTPUT:
//Enter an amount to evaluate (cents): 86
//86 cents can be given as 3 quarter(s) and 1 dime(s) and 0 nickel(s) and 1
cent(s)
//Would you like to do another evaluation? 'y' for yes: y
//Enter an amount to evaluate (cents): 5
//5 cents can be given as 0 quarter(s) and 0 dime(s) and 1 nickel(s) and 0 cent(s)
//Would you like to do another evaluation? 'y' for yes: y
//Enter an amount to evaluate (cents): 73
//73 cents can be given as 2 quarter(s) and 2 dime(s) and 0 nickel(s) and 3
cent(s)
//Would you like to do another evaluation? 'y' for yes: n
```

Part 2 (10pts) MULTIPLE CHOICE - 1 -> C - 2 -> D - 3 -> B - 4 -> D - 5 -> A

1. Which of the following is a legal call to the displayOutput function?
   void displayOutput(int total);

   a. void displayOutput(myTotal);
   b. displayOutput(int mytotal);
   c. displayOutput(myTotal);
   d. cout << displayOutput(myTotal);

2. Which of the following is true for a void function?
   a. There cannot be a return statement.
   b. The value of void should be returned.
   c. The value of 0 should be returned.
   d. Nothing is returned.

3. Call-by-reference parameters are passed
   a. nothing
   b. the actual argument.
   c. the value in the actual argument.
   d. the address of the argument.

4. If you need a function to get both the number of items and the cost per item from a user, which would be a good function declaration to use?

a. int,float getData();
b. int getData(float cost);
c. void getData(int count, float cost);
d. void getData(int& count, float& cost);

5. What is the output of the following function and function call?

```
void calculateCost(int count, float& subTotal, float& taxCost);

float tax = 0.0,   subTotal = 0.0;

calculateCost(15, subTotal,tax);

cout << "The cost for 15 items is " << subtotal

     << ", and the tax for " << subTotal << " is " << tax << endl;

//end of fragment

void calculateCost(int count, float& subTotal, float& taxCost)

{   if ( count < 10)

    {

            subTotal = count * 0.50;

    }

    else

    {

            subTotal = count * 0.20;

    }

    taxCost = 0.1 * subTotal;

}
```

a. The cost for 15 items is 3.00, and the tax for 3.00 is 0.30;
b. The cost for 15 items is 0.00, and the tax for 3.00 is 0.00;
c. The cost for 15 items is 0.00, and the tax for 3.00 is 0.30;
d. The cost for 15 items is 3.00, and the tax for 3.00 is 0.00;