1325 Homework 1

The purpose of this assignment is to get you used to the C++ language (you are not expected to use classes or take the object-oriented paradigm into consideration yet).

HW Submission requirements:
1) Put your name and ID number on the top of EACH assignment (in comments for programs)
2) Name each file the name written in blue above each problem
3) Put all the files (and their READMEs) into individual folders
4) Put these individual folders into a single folder and zip this single folder
5) Name the zipped folder HW1.zip (-5 points for incorrect name)
6) Submit this single zipped folder on Canvas

NOTES:

1. **INCLUDE A README FOR EACH PROGRAM *(-10 points) Remember a README should let the user (my TA) know how to run your program. Include a sample run of your program on the README and include any issues that may crash your program (for example, if your program specifically doesn't accept string inputs, include that) or any missed functionalities from the assignment prompt***

2. **MAKE SURE YOUR PROGRAM IS PROPERLY INDENTED *(-10 points)***

3. **MAKE SURE ANY COMMENTS YOU INCLUDE ARE MEANINGFUL *(-10 points) Do not just put random comments on every line of code***

4. **DO NOT CODE EVERYTHING IN MAIN *(Automatic 0)***

Problem 1 (20 points)-True/False. Submit a file named Answers.doc

*Answer the following true/false questions using the following code fragment. You must correctly state **WHY** your answer is true or false in order to receive credit.*

```
int main(int argc, char ** argv)
{
        vector <string> exchange_rates;
        string answer="start";
        float money;

        cout << "$$ Money Conversion $$" << endl;


        while (answer.compare("exit")!=0)
        {
                cout << "\n--What to do? \n << endl;
                cin >> answer;
```

```
if (answer.compare("new")==0)
{
```

1. We can assume *using* is used in the completed code.
2. We can determine what the return type is of the *compare* function.
3. From the code fragment alone, **all** functions in the above code fragment have the same parameter types.
4. We can assume that a maximum of two headers are included in the program.
5. The *if* statement will only execute if the value of answer is *true*.

Problem 2 (40 points)-Write a program. Submit a file named weight.cpp

Create a program that continuously *(10 points for continuously running)* allows a user to enter his or her weight in either pounds or kilos. The program should then convert the given weight *(15 points for each unit of measurement-total 30 points*).

Possible Sample Run 1 (your program does not have to match the following exactly, but should have the same functionality):

```
Please enter your name and weight.  James 182 pounds
Hi James- you weigh 82.5 kilos.

Please enter your name and weight. Connor 78
Not enough info to convert.

Please enter your name and weight. Connor 78 kilos
Hi Connor-you weigh 171.9 pounds.

Please enter your name and weight.  exit
Exiting…
```

Problem 3 (40 points)-Write a program. Submit a file named shapes.cpp

Create a program that continuously *(2 points for running continuously)* allows a user to enter a 2D shape and output information about calculations for this shape. The program should keep track of the number of shapes input by the user.

YOU decide what shapes and information are available for display (meaning programs may intake and output different information). A minimum of 3 shapes *(12 points for each shape, total 36* *points)* must be available for display. If there is no shape information available, it should be indicated to the user*(2 points)*.

<u>Possible Sample Run 1 (your program does not have to match the following exactly, but should have the same functionality):</u>

```
********
Shapes!
********

1.  Shape: Triangle
area=(base*height)/2

2.  Shape: Square
area=length*length
perimeter=4*length

3.  Shape: Diamond
Sorry, information for this shape is not available.

4.  Shape: exit

-3 shapes entered.
Exiting…
```