

## CSE 2010 Fall 2022

### Lab 2: Simple programs with different data types (30 points)

**Due by:** Wednesday September 7, 5:30pm on Canvas

Learning Objectives:

- Write simple programs that use different data types.
- Use the `<cmath>` and `<string>` libraries
- Properly prompt the user for input and process input accordingly.

#### Preparation

In this week's lab, we will be writing programs using concepts covered in Week 2's lecture. Look over how to declare numerical data types, string data types, when to use certain data types, and how to use and call functions in the `<cmath>` and `<string>` libraries.

Open up your Terminal and navigate to your Lab02 directory.

C++ reference website is a fantastic resource to keep on hand when programming. It has extensive and helpful information on C++. For this lab, the following webpages may prove to be helpful:

<http://www.cplusplus.com/reference/cmath/>

<http://www.cplusplus.com/reference/string/string/>

#### Programming Exercise 1: Quadratic Formula (5 points)

In your Lab2 directory, create a file called ***quadratic.cpp***. In this file, type the code below to implement a quadratic formula program that calculates the two solutions of a quadratic formula (we went over this in class.) Be sure to include a multi-line comment header at the top of the file with your name, file name, and program description.

```
/*Header*/
#include <iostream>
#include <cmath> using
namespace std;

int main()
{
    double a = 1, b = 0, c = -4;
    double x1 = (-b + sqrt(b*b-4*a*c))/(2*a);
    double x2 = (-b - sqrt(b*b-4*a*c))/(2*a);

    cout <<"The two solutions are x1 = " << x1 <<"", and x2 = " << x2 << endl;

    return 0;
}
```

Save the file and compile to make sure you don't have any errors. Remember the command to compile follows the format:

```
g++ sourceFileName.cpp -o executableName
```

So for this program it would be:

```
g++ quadratic.cpp -o quadratic
```

Now, modify your program to make it interactive. Instead of giving a, b, and c values in the code(which is referred to as hard-coding), modify the code so that the user can enter the values of a,b, and c. (Hint: use cin >> )

After modifying the program, save, compile and run with the values below. You should get the expected values:

a = 1, b = 0, c = -4      Expected Values: x1,x2 = -2, 2  
a = 1, b = 5, c = 6      Expected Values: x1,x2 = -2, -3

Remember that after compiling, you can run the program with the following command:  
**./quadratic.**

Once your programming is working, take a screenshot of your output.

### **Programming Exercise 2: Change Counter (10 points)**

Now it's time for you to spread your wings and write your own program from scratch. In a file called **change.cpp** write a program that prompts the user to enter the number of pennies, nickels, dimes, and quarters that they have. Calculate their total and display to them.

Example:

If the user inputs:

- Number of pennies: 65
- Number of nickels: 50
- Number of dimes: 30
- Number of quarters: 20

The program should display: **You have \$11.15 in change**

Save, compile, and test your program with the example values above. Take a screenshot of your terminal window that shows your error free compilation and test runs.

### Programming Exercise 3: The Healthy Programmers Program (15 points)

Ahsoka Tano is a computer science student trying to make some lifestyle changes and be more active during her programming filled days. She wants to start working out and knows that monitoring your heart rate during a workout is important to make sure you challenge yourself properly without hurting yourself. Only problem is she doesn't know how to measure that.

She did some quick research and found that the Karvonen formula can help individuals determine their target heart rate (HR) training zone.

The Karvonen formula uses maximum and resting heart rate with the desired training intensity to get a target heart rate.

$$\text{Maximum Heart Rate (MHR)} = 220 - \text{Age}$$

$$\text{Target Heart Rate (THR)} = [(\text{MHR} - \text{Resting HR}) \times \% \text{Intensity}] + \text{Resting HR}$$

Ahsoka could just Google "Karvonen Formula Calculator", but that would be no fun. Help your fellow CS student by writing a C++ program that does the following:

- **Prompt the user for:**
  - First Name
  - Last Name
  - Age
  - Resting HR (Beats Per Min aka BPM)
  - Training Intensity Range (minimum %)
  - Training Intensity Range (maximum %)
- **Then, display a user summary that has the following information:**
  - The user's first and last initials
  - Age
  - Resting HR (BPM)
  - Max HR (BPM)
  - Target Heart Rates (BPM) for their range
- Once your program is working, test it with the values in the screenshot below. You should get the same output.
- Then, run your program and test it with different values.

```
Welcome to the Healthy Programmer Program!
```

```
This program will help you determine your target heart rate training zone!  
Please enter the following information.
```

```
First Name:Ahsoka  
Last Name:Tano  
Age:20  
Resting Heart Rate (BPM):60  
What is the MINIMUM intensity (%) you would like to train at:60  
What is the MAXIMUM intensity (%) you would like to train at:70
```

```
Thank you! Please see your summary below.
```

```
Initials: A.T  
Age: 20  
Resting Heart Rate (BPM): 60  
Max Heart Rate (BPM): 200  
If you want to work out at an intensity between 60%-70%, your target HR training zone is between 144-158 BPM.
```

```
Thank you for using the Healthy Programmer Program! Have a great workout!
```

Once your program works as expected, take a screenshot of the output. (You should have two screenshots)

**Once you have completed all 3 programming exercises, log on to Canvas. Under “Modules > Lab Assignments > Lab 02” , submit the following:**

- .cpp file for Exercise 1
- Screenshot for Exercise 1
- .cpp file for Exercise 2
- Screenshot for Exercise 2
- .cpp file for Exercise 3
- 2 Screenshots for Exercise 3