

CSE 1321L: Programming and Problem Solving I Lab

Assignment 1 – 100 points

Solving Problems

What students will learn:

- 1) Problem solving
- 2) Terminology
- 3) Basic program structure
- 4) Input and output with the user
- 5) Basic calculations and those calculations requiring an intermediate solution

Overview: For most of you, this will be the first time you've done any programming, which is exciting! The write-up of this first assignment will be a little longer than others because we want you to have an understanding of how things are going to roll out the rest of the semester. Advice: Start early (certainly not the day the assignment is due), practice, and ask a lot of questions.

Unless calculations are trivial, you'll almost always want to use an intermediate variable – where you store part of the solution. For example, you might remember the equation for gravity as:

$$F = G \frac{m_1 * m_2}{R^2}$$

How would you write this as code? You might solve it in parts instead of one shot because it makes it easier to check. It would look something like:

```
MAIN
    CREATE f1, m1, m2, r, g, temp1, temp2, temp3
    temp1 = m1 * m2
    temp2 = r * r
    temp3 = temp1/temp2
    f1 = g * temp3
END MAIN
```

Final note: **don't cheat**. If your temptation is to look online, don't. Come see us instead and ask questions – we're here to help. Remember, you're going to have to write code in your future job interviews, so learn it now in order to secure a high-paying job later.

Assignment 1A:

Build a Website (URL): Did you know that each part of a website's address has a specific name? For example, the website "<https://ccse.kennesaw.edu/first-year-experience>" is comprised of the following:

Scheme	Subdomain	Second-level domain	Top-level domain	Subdirectory
https	ccse	kennesaw	edu	first-year-experience

For this first assignment, you will prompt the user to give you values for the five parts of a URL (*Uniform Resource Locator*) shown in the table. Then, you will combine them and print out the full URL to the user.

Save your source code in a file called **Assignment1A** (with a file extension of **.cpp**, **.cs**, **.java**, or **.py**)

Sample Output:

Let's make a website (URL)!

Scheme: **gopher**
Subdomain: **marietta**
Second-level domain: **ksu**
Top-level domain: **net**
Subdirectory: **gopher-protocol**

Your URL is:
gopher://marietta.ksu.net/gopher-protocol

Assignment 1B:

Color Distance. If you've ever used an art program, you're likely aware that computers store colors using a series of numbers. The most common way is known as RGB – Red, Green, and Blue. For instance, pure red might be shown as (255, 0, 0), while a shade of purple might be (128, 0, 128).

We can use this information to calculate the “color distance” between two colors – how similar they are to each other. There's a lot of advanced formulas to calculate this, but for this assignment we'll practice converting a simple algorithm into a working program. The formula is as follows:

$$\text{distance} = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}.$$

Source: https://en.wikipedia.org/wiki/Color_difference

Hint: You can replicate exponents using multiplication. For example, x^2 is the same as $(x * x)$. You will also need to use square root. We will learn more about functions later in the course, but for now you can use the following command to get this value:

Java	C#	C++	Python
Math.sqrt (variable)	Math.Sqrt (variable)	#include <cmath> sqrt (variable)	import math math.sqrt (variable)

At the start of your program, ask your user to give you two RGB values. Then apply the above formula and tell the user what the color distance is between the two colors.

You do not need to match the number of decimals precisely, as long as the answer is correct. We'll learn how to do that later.

Call the file **Assignment1B** (with a file extension of **.cpp**, **.cs**, **.java**, or **.py**)

Sample Output:

First color

R: **255**

G: **0**

B: **0**

Second color

R: **128**

G: **0**

B: **128**

Color Distance: 180.3163

Assignment 1C:

Food Bank: You volunteer at the Greater Marietta Community Food Pantry and your manager found out that you're taking a computer programming class. They've asked you to build a program that calculates how many people can be fed based on the quantity of certain items donated.

Food Item	People fed per item (approx)
14.5 oz Canned Soup	2
1 lb Bag of Brown Rice	4
14.5 oz Canned Vegetables	3.5
16 oz Peanut Butter	7

In your program, ask the user how many of each item they're donating. Calculate how many people can be fed from the donation, and let the user know both the total and per item information.

Examples of the program are shown below. Save your source code in a file called **Assignment1C** (with a file extension of **.cpp**, **.cs**, **.java**, or **.py**)

Sample Output:

How far your donation goes

```
How many cans of soups? 2
How many bags of rice? 1
How many cans of vegetables? 3
How many cans of peanut butter? 7
```

```
Your donation will help feed 67.5 people!
4 people with the 2 can(s) of soup
4 people with the 1 bag(s) of rice
10.5 people with the 3 can(s) of vegetables
49 people with the 7 can(s) of peanut butter
```

Submission:

1. You will submit 3 separate files – one for each of the assignments above.
2. Upload separate files (one for each assignment) to the assignment submission folder in Gradescope. Do NOT submit homework in D2L.
3. Submit your work by the due date.