EE 1301
Lab 2: Sequence and Selection

Sequence refers to the idea that any computational process follows a specific order of execution. For example, in C++ each statement is completed prior to beginning any other statement. Furthermore, operators within statements have an execution order based on operator precedence. **Selection** describes the alteration of the sequence based on dynamic conditions. For example, an "if" statement can cause the sequence of execution to branch to a different location in the program code.

In this exercise you will continue to build your understanding of simple C++ programs, including those that modify processing based on simple dynamic selection. As usual, take advantage of all the help around you, and be sure to explore and play around on your own to get a better grasp on the code and concepts.

Note: Save the programs you write for the lab. They might be useful when you do the homework assignments. Each problem solution should be maintained in its own (separate) source file. You may reuse any code that you wrote in lab.

Warm-Up

(1) Special Relativity

In special relativity, an object that has length L centimeters when at rest with respect to the observer has a relativistic length of L_R centimeters when traveling at velocity v away from the observer.

$$L_R = L\sqrt{1 - v^2/c^2}$$

Here, c is the speed of light, approximately 3×10^{10} cm/sec.

Write a program that will calculate and display the relativistic length L_R of an object of rest length L that is traveling at velocity v. In your program:

- Use type `double` for L, v, c, and L_R.
- Initialize c to 3×10^{10} cm/sec.
- Have the user input values for L and v.
- Include the file cmath so you can use the sqrt() function.

Test your program using L = 32.5 cm and v = 2.2×10^{10} cm/sec as one of your test cases. When you enter the value of v, use e (scientific) notation for convenience (e.g., 3.0×10^{10} = 3.0e10). Also test your program with L = 42.0 cm and v = 3.2×10^{10} cm/sec (again, enter v using scientific notation). What happens?

Stretch

(1) Basal Metabolic Rate

The Harris-Benedict equation estimates the number of calories your body needs to maintain your weight if you do no exercise whatsoever. This is called your basal metabolic rate, or BMR. The formula that calculates the calories needed for a woman to maintain her weight is:

BMR = 655 + (4.3 × weight in pounds) + (4.7 × height in inches) − (4.7 × age in years).

The formula that calculates the calories needed for a man to maintain his weight is:

BMR = 66 + (6.3 × weight in pounds) + (12.9 × height in inches) − (6.8 × age in years).

A typical chocolate bar contains approximately 230 calories. Write a program that allows the user to input his or her weight in pounds, height in inches, age in years, and the character 'M' for male or 'F' for female. The program should then output the number of 230-calorie chocolate bars that the user needs to consume to maintain his/her weight.

Testing Discussion: Come up with 3 sets of test data for this program. When you have made up your test cases, discuss with someone around you why you chose the test cases you did, and more generally, what makes good test cases for a program like this.

(2) Temperature Conversion

For weather reporting, the daily temperature will generally be given in degrees Celsius or Fahrenheit. Write a C++ program that will convert a temperature given in one system of units to the other. For example, if the temperature is given in degrees Fahrenheit, convert it to Celsius, and vice versa.

Your program should prompt the user to provide a temperature value (object type double) and a single character ('f' or 'c') to indicate if the value is in degrees Fahrenheit or Celsius. Read in the temperature and scale values, then compute and display the corresponding equivalent temperature in the other scale system. Your output message should indicate whether the result is Celsius or Fahrenheit. For example:

Example 1 (underlined values are user inputs):

```
Enter the temperature: 26.6
Enter Celsius (c) or Fahrenheit (f): c
The temperature in Fahrenheit is 79.88
```

Example 2 (underlined values are user inputs):

```
Enter the temperature: 93.2
Enter Celsius (c) or Fahrenheit (f): f
The temperature in Celsius is 34
```

Use the following conversion formulae:
$F = C \times (9/5) + 32$
$C = (F - 32) \times (5/9)$

[**Hint**: Remember that there is a difference between division of integers and division of doubles. **Be careful** when translating the equations above into C++.]

Test your program using the following values, as well as some other test cases you think up:
(i) convert 37 degrees Celsius to Fahrenheit
(ii) convert 98.6 Fahrenheit to Celsius
(iii) convert 32 Fahrenheit to Celsius
(iv) convert -40 Fahrenheit to Celsius

(3) Simlple Integer Calculator

Write a simple integer calculator. This should be able to handle +, -, / (integer division), and *. The user should provide inputs in the format: [integer][operator][integer]. For example: 2+56 or 5*3 (it is easier to make the calculator work if there are no spaces between the integers and the operator). Design your program so that it does not crash as long as the input is in this format. Your program should only compute one output then stop. You should display both what the user entered and the numeric result.

Example 1 (underlined values are user inputs):

```
Enter an equation: 7+1
7+1 = 8
```

Workout

(1) Payroll

An employee is paid at the base rate of $16.78 per hour for the first 40 hours worked in a week. Any hours over that are paid at the overtime rate of 1.5 times the base rate. From the worker's total pay, 6% is withheld for Social Security tax, 14% is withheld for federal income tax, 5% is withheld for state income tax, and $10 per week is withheld for union dues. If the worker has three or more dependents, then an additional $35 is withheld to cover the extra cost of health insurance beyond what the employer pays. Your task is to write a program that will read in the number of hours worked in a week and the number of dependents as inputs, and will then output the worker's total pay, each withholding amount, and the net take-home pay (after withholdings) for the week. Solve this problem in four steps, as outlined below.

(i) **Problem solving and planning**: Use pencil and paper to plan out your solution. Spend at least 3 minutes. This may not be enough time to solve the problem, but it will be enough for you to check if you understand the problem, and to get started on solving it. If possible, compare your problem solving approach with a neighbor's. Are they similar or different?

(ii) **Outline:** Come up with a detailed outline of the problem. Don't use C++ yet. Instead use an English-like description (pseudocode) to outline all the steps the program will need to perform, in the order it will need to perform them.

(iii) **Code**: Once you have the outline written, turn it into a C++ program.

(iv) **Test and revise**: Devise a comprehensive set of test cases, test your program, correct it as needed, and continue testing and revising until your program is correct. Keep in mind that gdb can often be a great help in identifying problems with a program.

Check and Discussion

Before going on to the challenge question, write down:
(i) one important thing you learned in the lab so far
(ii) one question you still have about the C++ you used in the lab today
When you have both written down, discuss with a neighbor or TA.

Before going on to the challenge questions, have a TA check your work.

Challenge

These challenge problems are for those of you who just can't get enough! You should be able to complete the warm-up, stretch, and workout problems in the lab. Try this problem if you have extra time or would like additional practice.

(1) Ancient Greek Taxation

Suppose that in ancient Greece, where the unit of currency was the drachma, the following progressive income tax code was used.

First 10,000 drachmas of income are taxed at 0%
Next 20,000 drachmas of income are taxed at 10%
Next 40,000 drachmas of income are taxed at 20%
Drachmas after 70,000 are taxed at 30%

For example, someone earning 100,000 drachmas would owe:

$$10{,}000 \times 0.0 + 20{,}000 \times 0.1 + 40{,}000 \times 0.2 + 30{,}000 \times 0.3 = 19{,}000 \text{ drachmas.}$$

Write a program that reads in an income value in drachmas, and outputs the amount of tax.

(2) Selection Problem from EE

Think up a simple problem from electrical engineering whose computer solution involves selection. Write a program that computes a solution to the problem.

Logout

Remember to logout of the computer before you leave.