Assignment 2

Due Wednesday, June 8th, 2011, by 11:59pm

- Submit deliverables to the Course Management System: <u>https://courses.cs.sfu.ca/</u>
- Late penalty is 10% per calendar day (each 0 to 24 hour period past due).
 - Friday, June 10th is the last possible day to submit (11:59 pm).
- This assignment is to be done individually. Do not show another student your code, do not copy code found online, and do not post questions about the assignment online. Please direct all questions to the instructor or TA:
 - For CMPT125, email <u>cmpt-125-d2-help@sfu.ca;</u>
 - For CMPT128, email <u>cmpt-128-help@sfu.ca</u>.
- See the marking guide for details on how each part will be marked.

1. Expression Trees

Using a computer program (for example MS Word, PowerPoint, or any other) to draw an expression tree for each of the following C++ expressions. For each tree:

- i) Draw the expression tree; and
- ii) Evaluate the expression using the tree. You may want to prove you got the correct answer using the C++ compiler to evaluate the statements.

Expressions:

- a) res = 1 + 52 % 5;
- b) res = 8 * 2 + 5 / 2;
- c) res = -1 * 2.0 + 3 / 2 * 1 * 10

Assume that res is declared to be an int.

For Submission:

To submit, you will need to "print" your work to a PDF file named expressionTrees.pdf. If your machine does not already support creating PDF files, you may need to install a PDF printer (for example, <u>PDFCreator</u>). One is installed in the CSIL lab which you may use to generate the PDF.

2. Costly Habits

On tests for this course, you will have to write programs on paper without the use of a compiler. For this question, you are asked to first create a solution by hand on paper, and then implement it on the computer. Your hand-written paper submission will be marked for completeness, and your electronic submission will be marked for correctness. Therefore, please **write it out on paper first** for the practice, and then implement it on the computer.

Problem Description:

- You are worried about how much your friends are paying to support their habits (such as buying a cup of coffee each day); so you decide to write a program to show them.
- You want to show the user how much their habit costs them in a single year (365 days), and what it will cost them between now and when they are 75 years old.
- To do this, you realize you will need to ask the user's age (in years), and the cost of their habit for one day (such as 4.75 for a coffee each day).
- Because you may be dealing with large amounts, you also want to show them how many million dollars that is! (For example, \$25,000 is 0.025 million dollars).

Your Tasks:

- a. On paper, write a program which solves the above problem.
 - A sample run of the program is shown below. You don't have to match it exactly, but your output should be similar.
 - Remember that you are writing a program to solve the problem; you are not writing a program to solve just the one example shown below.
 - You must use named constants to store any hard-coded values in your program. (Values such as 0 or 1 don't need named constants, but most other values do).
 - Your program might not show two decimal places for dollar amounts; that is OK.
 - Ignore leap years, and ignore that the user may be part way through the year (i.e., if the user said they are 60, then assume there are 15 full years until they are 75).
 - Assume the user is no older than 75, and that the user enters correct data.

```
How old are you? <u>19</u>
How much does your habit cost per per day? <u>10.25</u>
This year your habit will cost you $3741.25.
Between now and when you are 75, it will cost you $209510!
That's 0.20951 million dollars!
```

- b. Implement your solution in the computer, naming the program habit.cpp. Debug your application. You do not have to correct your paper submission; it will only be marked for completeness. Do not implement it on the computer first! This question is to help you practice writing programs on paper.
- c. Compare your electronic solution to your hard copy. Make a note to yourself about what you had to correct to make it work in the computer. Keep this in mind when you are writing tests! (Nothing is marked for this part of the question).

3. Planet Calculator

Write a program named planet.cpp which calculates how high you could jump, and how old you would be in different places in our solar system.

- Start by asking the user how high they can jump, and how old they are (on Earth).
- Then, using the table below, calculate how high they could jump, and how old they would be for each of the places listed in the table.

Planet or Body	Gravity relative to Earth	Year relative to Earth
Earth	1	1
Moon	0.17	N/A
Mars	0.38	1.88
Pluto	0.07	247.7
Jupiter	2.36	11.86
Sun	28	N/A

♦ Hints:

- The gravity numbers in the table are relative to Earth's gravity. So, if you can jump 2 meters on Earth, you could jump (2m / 0.17) on the moon, which 11.8m.
- The year lengths are relative to an Earth's year. So if you are 21 on Earth, you would be (21/1.88) years old on Mars, which is 11.17 Mars-years.
- For places where the year value is meaningless (the moon and the sun), just display N/A for "not applicable".
- An example run is shown below.
 - Sour output does not have to be exactly the same, but it should be similar.
 - Note that tabs have been used to line up (most) of the information; it is OK if the age for some places do not line up with the rest of the data (limitation of tabs).
 - For different inputs (jump-height and age), your program might display different numbers of significant digits; this is also OK.

```
So, how would you do in other places in the solar system?
How high can you jump (in meters) on Earth?
2
How old are you (in years) on Earth?
21
Planet Jump
             Age (years)
_____
       ____
               _____
Earth
       2
               21
Moon 11.7647 N/A
Mars
      5.26316 11.1702
Pluto 28.5714 0.08478
Jupiter 0.847458
                      1.77036
Sun 0.0714286
                     N/A
```

4. Mowing Quotes

Your cousin is starting up a business to mow neighbour's lawns over the summer. He needs a good way of estimating how much to charge for his services. Here is what he knows:

- He's going to change \$15.75 for travel time, no matter how much time it takes him to get to the house.
- He's going to charge \$25.00 per hour he spends cutting the lawn; and his estimates will be based on cutting 250 square meters (m²) of lawn per hour.
 - He has decided he will round up the estimated number of hours for a job. For example, if someone has a 10m by 30m lawn (300m²), he'll estimate that it's a 2 hour job; but a 250m² lawn would only be a 1 hour job.
- Because he's starting out, he's going to give a 5% discount to all his clients. This discount will apply to both travel time, and to the cost of cutting the lawn.
- Finally, he needs to charge tax, which is currently 12% (applied after the 5% discount is given to the sum of travel cost and mowing cost).

Write a C++ program called lawnPrice.cpp which shows an estimate for a specific lawn.

- Use named constants appropriately, and comment your code.
 - Specifically, you must make at least the following named constants: the travel cost, how many square meters of lawn he cuts per hour, the cost per hour, the discount rate, and the tax rate.
 - Your output must use the constants where applicable. For example, don't hardcode the tax rate's output with something like:
- cout << "Tax (12%)..."; // WRONG! Use the named constant not 12!
 There are a number of calculations in the program; you should do most of the calculations outside of cout statements (i.e. don't put calculations inside your cout statements).
 - Hint: Calculating tax of 12% is the same as multiplying by 0.12.
- Your program's output should resemble (though it need not be identical to) the following.
 - Sour output should line up nicely, as shown below.
 - Sour dollar values might not be shown with exactly 2 decimal places; that is OK.

```
Enter the size of the lawn (width x height) in meters:
15.7 22.1
To cut a lawn 15.7 x 22.1 will take 2 hours.
Estimate:
_____
Cutting (2H)
              $50
              $15.75
Travel Cost
               _____
Subtotal
              $65.75
Discount (5%)
              - $3.2875
Tax (12%)
               + $7.4955
               ========
Total Owing:
              $69.958
```

5. Deliverables

Submit the following in hard copy at the start of lecture:

1. Hand-written bad-habit calculator code. You do not need to add comments to your handwritten code. It will be marked for completeness, not correctness. Make sure you have your name on your hand-written code!

You do **not** need to print your source code to any of the programs listed below.

Submit the items listed below to the Course Management System: https://courses.cs.sfu.ca/

- expressionTrees.pdf
- 🔷 badHabit.cpp
- > planet.cpp
- lawnPrice.cpp

Each of your .cpp files must begin with a comment stating your name, your SFU user ID, and your SFU student number.

Please remember that all submissions will automatically be compared for unexplainable similarities. We expect that everyone's submissions will be quite similar, given the nature of this assignment, but please make sure you do your own original work; we will still be checking.