Competitive Coding In Class Exercise

Variables and Assignments

Paul Cao

Variables, as the basic building blocks of the C++ programming language, play an important role in the structuring of any C++ programs. In this lab, we will focus on the basic principles implemented in C++ variables. In addition, we will also practice formatted output using cout.

1. Variables

All variables used in a C++ program must be declared. The declaration of a variable can be done in several different ways; 1) right at the beginning of the main function, 2) right before its use in the program, and 3) right before the main function. In general, the syntax for variable declaration is:

Type_name Variable_Name_1, Variable_Name_2,…;

Here is the list of variable types that we have used:

Variable Name	Variable Name
string – a sequence of symbols	double - 8 bytes, 15 digits precision *double x; or double x = 34.56;*
int - 4 bytes *int x; or int x = 34;*	char - holds any single character on the keyboard *char c; or char c = 'a';*
long long – a large whole number	bool – true or false kind of value

Every variable is identified by a name that is referred to as identifier. An identifier must start with either a letter or the underscore symbol. The remaining characters must all be letters, digits, or the underscore symbol.

In general, every variable should have a type. An attempt to assign a value of a different type other than the original assigned type to a variable is referred to as "type mismatch". One has to be very careful that not all compilers will allow type mismatch, i.e., may produce an error when such attempts are made.

2. Arithmetic Operators and Expressions

To perform mathematical calculations, we combine variables and/or numbers using arithmetic operators such as: + for addition, - for subtraction, * for multiplication, / for division, and % to find the remainder of a division. The arithmetic operators can be used with numbers of type int, double, float, long double, short, or with a combination of different types. When you use arithmetic operators, you need to be careful that if the correct variable types are not used, the result can be very different from what you were expecting. For example: 7.0/2 = 3.5, 7/2.0 = 3.5, however, 7/2 = 3 (not 3.5). The reason for such a significant difference is because the last one is actually using integer division. Perhaps the most significant problem will be seen in cases where the numerator is smaller than the denominator. In such cases the integer division will result in 0. An example of such a case is 5/9 = 0.

There are some shorthand notations that combine the assignment operator (=) and an arithmetic

operator. The general form for this notation is:

Variable Operator = Expression, which is equivalent to: *Variable = Variable Operator* (*Expression*).

The expression can be another variable, a constant, or a more complicated arithmetic expression. Here are some examples:

Shorthand Notation	**Equivalent**
total += 2;	total = total+2;
amount += tax - rebate	amount = amount + (tax - rebate)

Similarly, we may use -=, *=, /=, and %=.

Activity

1. Answer the following questions. Put your answers to these questions as comments in your problem 2 code. If you are not sure what the answer is, feel free to test the expression in C++.

I. What is the answer to the following arithmetic operations if they are written in a C++ code?
   a. (1/5) * 3
   b. (8/2) * 4
   c. (1.0/5.0) * 3

   A : 0.6
   B : 0
   C : NoA

   (1/5.0) * 3

   Is there any difference between (a) and (c)?
   Let's then make some calculations using / and %!!

II. Convert the following mathematical expressions into C++ expressions:
   a. 4x
   b. (2x + y)/3y
   c. (2x - 3y)/(z -1)
   d. $y = ax^3 + 7$

III. Translate the following English statements into C++ expressions:
   a. Assign 2.7 to the variable e
   b. Assign the remainder of division of 5 by 3 to the variable rem
   c. Assign the product of variables e and rem to the variable prod

2. Programming: Quiz Averages

Write a C++ program that prompts the user to input 3 quiz scores (whole numbers from 0 to 100). After that, the program calculates the average quiz score. The program then displays the results of its calculations on the screen. For example, if the user input 90, 88, 75 as input, your output should be as follows. Try to use tabs to align things but it is ok if you simply use spaces to separate those output.

Quiz 1	Quiz 2	Quiz 3	Average
90	88	75	84.33

3. Debugging exercise

When I enter paul and then enter 44, it should output *Hey paul, you will be 45 years old next year*

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    int name;
    cout>>"Enter your first name: ";
    cin<<name;

    string age;
    cout<<"How old are you: ";
    cin>>age;

    cout<<"Hey name, you will be "<<"age+1"<<" years old next year"<<"endl";
}

```