**ITCS 1212L**

**Pre-lab 2**

**You are going to practice on the following topics:**

Sequence, Flowcharts, Identifiers, Fundamental data types, Declaration of variables, Initialization of variables, assignment operators, arithmetic operators, compound assignments, increase and decrease, standard input and output.

**Learning Objectives:**

* **Learn how to declare variables**
* **Make use of the arithmetic and the assignment operators**
* **Read input, generate output, make use of a logical sequence and flowcharts**

**Read the hand out named Lab-lessons2.pdf.**

1. **Answer the following questions based on what you learnt in lab lesson 2:**
2. Explain why in a program you need variables? To reserve a place in memory to place values (for various reasons), and to declare what format they will take. Or, how the Lab 2 Lessons defined it: *A variable is a named location in the computer’s memory whose contents can change as the program is running.* (**You need to know this definition)!**
3. Person’s name to a person is similar to \_an identifier\_ to \_variable\_ in a computer program.
4. What are the identifiers in the following statements?
   1. x = 5;
   2. y = 2;
   3. r = x- y

Identifiers here are: r, x, y.

1. How do you declare variables? You must declare a variable within the scope with which you intend to use it (i.e. declare it in main if it’s going to be used there. Declare it globally if you want to use it in several functions, though this is generally considered bad form). You must start the statement with the variable’s data type (int, dbl, float, string, char, etc…). The next part of the statement is the variable’s name (also called the identifier). There are naming restrictions, such as: It must always start with a letter, it can only use letters, numbers, and underscore. It may not use the same name as a key word for your compiler or the C++ language.
2. Give different examples of declaring variables of different data types.

Int numberOfPennies;

float totalDollarValue;

char ‘B’;

1. What do we mean by variable assignment? Give an example for it.

It is when a value is given to a variable (this can be a letter, symbol, string, or numeric value). It may be assigned through user input by using the command cin >>, or by assigning it through the program’s lines of code itself. It is important to remember that this value, regardless of what it is, is being assigned a location and size in memory.

Example: x = q + h;

(This adds the values contained within q and h, then assigns that new value to the variable x)

1. Explain how does this operation work in details?

Like I said in part (F), a user can assign value with cin, or the program itself can assign value with operators such as the = sign. A programmer can also choose to give a variable a value immediately upon declaring the variable:

int x = 10; Or even:

int x (10);

Later we will learn that the values inside of variables can be passed to different functions, but those functions will need their own declared variables to house these values, or pass the variable’s data and location with a pointer allowing the program to change the value of a variable from within a different function…. Though that’s probably a few chapters away from where we are now.

1. Starting from the basic definition and functionality of a computer system, explain the concept of ‘logical sequence’?

# I cannot put it better than the Lab 2 Lessons put it: The order you write the statements in your code is the order they will be executed in. All programs must have a function named main( )

Therefore, we cannot write code in any sequence we desire and expect it run smoothly or run at all. There must be a logical flow to the statements for a well-executed program.

1. List the basic arithmetic operators.

Modulus, multiplication, division, addition, subtraction, and comparing values: %, \*, /, +, -, and < > <= >=

1. What do we mean by operator precedence? How do we calculate k in the following cases?

Just like in math class, we learned order of operations, to inform us of the correct order to perform arithmetic in long equations. Computers must follow similar rules or it would spit out non-sensical garbage. Inside parentheses always go first.

k = (( a + c ) – d) \* a

So, the value stored in k would be: a + c, then that value minus d, then that value multiplied by a.

k = a + c – d \* a

So, the value stored in k would be: d \* a, then separately a + c, then the value of a + c subtracted by the value of d \* a.

1. For the above formula, develop step-by-step algorithms for calculating k.

k = a + c;

k =- d \* a;

1. The data type that holds only whole numbers with no fractional component is called a(n) \_integer\_(int)\_.
2. \_float\_and\_double\_\_\_\_\_ are data types that hold numbers with fractional components.
3. \_modulus\_(%)\_ is an arithmetic operator that gives the remainder of a division problem.
4. \_boolean\_ data types only have two values: true and false.
5. In C++, why do we use // or /\* \*/ ?

A couple of reasons. 1) It’s good form. 2) It allows a different programmer to more easily see what is going on inside of a written program. 3) It can help the original writer of the code to job his/her memory on how they were going about their program. 4) This note taking process in grayed out text is not seen by the compiler or the executable, which leads to my next reason 5) It’s a great way to debug certain sections of code. If you’re trying to pin point which particular section is giving you a headache, it’s much easier if you can ignore other sections of code temporarily.

1. Flowchart is \_a\_graphical\_ representation of program logic.
2. How do you specify the sequence in a flowchart?

Using arrows.

1. Calculate the results of the following operations? (order of precedence)
   1. -16
   2. -21 + 5
   3. 15 – 36 + 5
   4. 15 – 72/2 + 5
   5. 15 – 12 \* 6/2 + 5

All five lines equal -16.

1. For the following part, try to execute each of the following programs on the paper and find the outputs.
2. int numberA = 3.5 \* 5;

cout << numberA;

17

1. int numberB = 3.1 + 5 \* 3 – 1;

cout << numberB;

17

1. float numberC = 4/5 + 2;

cout << numberC;

2.8

1. int numberD = 2 \* 6 + 5/3 \* 8;

cout << numberD;

20

1. Calculate and compare the value of a for each of the following cases:

Case 1)

a = 3; a’s value will be equal to 3

a = ++b; a’s value will be replaced with b + 1

Case 2)

a = 3; a’s value will be equal to 3

a = b++; a’s value will be replaced with b + 1

1. What is the output when the following code fragment is executed? try to execute it in paper also add comments to this code.

#include <iostream> // The header file, so keywords and commands work

using namespace std; // Including the C++ standard library

main() // Every program has a main() function

{

int i = 5, j = 6, k = 7, n = 3; // Declaring integer variables and assigning them values

cout << i + j \* k-k % n << endl; // Using the variables that already have values to perform an arithmetic equation (behind the scenes. The user does not see this math), then displaying the answer of: 46

cout << i /n << endl; // The value of i divided by n. Which is 1 (since they are both integers).

return 0; // This returns a value of 0 to the executable call the program (ends the program).

}

6. Consider a case that a person uses a coupon to purchase a cookie. Thus he can get some discount on the original price and pay less amount of money. To simulate this case, the following program gets the price of an item and the quantity of the purchased item and the discount percent from the user and calculates the final purchase price. Trace the program line by line and comment what it does in each line.

#include <iostream> // The header file, so keywords and commands work

#include <cmath> // This allows additional arithmetic functions to be performed in the program (more advanced math one might say)

using namespace std; // Including the C++ standard library

int main() // Always gotta have main()

{

double itemPrice; // Declaring a double variable

double discountRate; // Declaring a double variable

double finalPrice; // Declaring a double variable

double quantity; // Declaring a double variable

double discount; // Declaring a double variable

cout<<"enter the item price:"; // Prompt user input

cin>> itemPrice; // Receive user input (the base cookie price)

cout<< "enter the quantity of item:"; // Prompt user input

cin>> quantity; // Receive user input (the number of cookies being purchased)

cout <<"enter the discount rate :"; // Prompt user input

cin>> discountRate; // Receive user input (the discount value)

discount= itemPrice \* quantity \* discountRate \* 0.01 // Calculates the savings given all the previous values provided

finalPrice = itemPrice \* quantity - discount; // Calculates the total price after (the cookie price times the number of cookies, then subtracting the discount determined in the line above)

cout<< "The final price is :" ; // Framing the output for the user

cout << finalPrice <<endl; // Displaying the final price

return 0; // Ends the program

}