PRELAB #3 – CSC 150

Just so you know, all my answers are in green

Purpose: Demonstrate use of:

- Assignment statements
- Variables
- Input/output
- Typecasting
- Writing small programs.

I. Write variable declarations for the following. Choose a data type most appropriate to the data to be stored, and use meaningful variable names.
Example: quantity of apples `int num_apples;`

1. course grade (to hundredths) double course_grade;
2. exam letter grade (no plus or minus) char letter_grade;
3. distance from Earth to Sun (whole meters) const long long int EARTH_TO_SUN;
4. your GPA double gpa;

In general, which two data types should you use for integer values and for real values when writing programs in CSC 150?

int and double

II. Write the statement(s) necessary to code this formula. Use temporary variables and break up the computation if you desire. Write the declaration for any additional variables you use. Assume the variables p, x and y below are already declared as doubles and that x and y have been assigned some values. Implement the formula as close to as given as reasonable, don't do any algebra to attempt to simplify the problem. Use parentheses as needed, but don't use them excessively.

$$p = \frac{3y - y * x^2}{2 - y}$$

p = ((3y – (y * pow(x , 2.0))) / (2.0 - y))

III. Complete the section exercises in chapters 3-6.

- Chapter 3: Sections 3.4 (page 64)
- Chapter 4: Sections 4.1 (page 77), 4.3, item 2 & 3 (page 87)
- Chapter 5: Section 5.1 (page 110-111)
- Chapter 6: Sections 6.1 (page 145)

Ch. 3

1. “m” in Main should be lowercase
2. There shoiuldnt be a semicolon after main
3. There should be curly brackets after main
4. Return should have a lowercase “r”
5. Looks good!

Ch. 4

1. Numeric Literal
2. Illegal – Char has the single quotes, and must have only one character
3. String Literal
4. String Literal
5. Character Literal
6. Illegal – Hex must have a 0x before it, or it should be in “” or ‘’ for string/char
7. Illegal – the closing should be a “ not ‘ so it’s a string

Ch. 5

1. A. C++ is very cool
   B. Hello
   World
   C. Computers are useless.
   They can only give you answers.
   Pablo Picasso
2. (C) - <iostream>
3. (B) – cout
4. (A) – cin
5. A. Invalid – Arrows should point towards cout
   B. Invalid – needs semicolon
   C. Invalid – cin can’t output text
   D. Valid – but chaining cin like that is risky
   E. Invalid – output must be in quotes
   F. Valid.
   G. Invalid – cin still can’t output text
   H. Invalid – stop trying to make cin output text. Please.
   I. Valid. But chaining cin is still risky

Ch. 6

1. Valid.
2. Invalid. Only one variable on the left.
3. Valid. Though I would use parenthesis just to be sure it’s correct
4. Invalid. A variable must be on the left.

IV. Evaluating expressions

For the expressions below, determine the resulting value that will exist. Assume the following declarations before each statement. Carry out all decimal operations to the nearest 1/100th. If the answer is a floating point value that happens to be a whole number, show it to two decimal places (example: 32 / 8.00 → 4.00 not 4 .) If an expression is an invalid statement, write INV in the answer area.

```
int a = 0;          double v = 0.00;
int b = 2;          double w = 1.00;
int c = -4;         double x = 1.50;
int d = 5;          double y = 3.50;
```

1. _____ 5 + c / d = 5
2. _____ d / 2.0 = 2.5
3. _____ c % b = 0
4. _____ x / w = 1.5
5. _____ x / b = .75
6. _____ x + y 2.5
7. _____ (c - b) / b -3
8. _____ a + b * x + w 4
9. _____ d + c % b 5
10. _____ y % d / w 3.5

V. Evaluating expressions

For the expressions below, determine the resulting value that will exist. Assume the following declarations before each statement. Carry out all decimal operations to the nearest 1/100th. If the answer is a floating point value that happens to be a whole number, show it to two decimal places (example: 32 / 8.00 → 4.00 not 4 .) If an expression is an invalid statement, write INV in the answer area.

```
int a = 0;          double v = 0.00;
int b = 2;          double w = 1.00;
int c = -4;         double x = 1.50;
int d = 5;          double y = 3.50;
```

1. _____ int (y) / x 2
2. _____ double (d) / b 2.5
3. _____ double (d / b) 2
4. _____ a + d /(double) (b) + w 3.5
5. _____ int (y) / int (x) 3
6. _____ int (y) * x 4.5

VI. Program

The Quadratic Formula, a shortcut for solving some quadratic equations, is given as:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

where the original equation is in the form: $ax^2 + bx + c = 0$

Write a program that prompts the user for the values of a, b and c. Calculate the two roots of x. Display the original inputs and the two solutions in a manner similar to the sample below.

You may assume that b^2 will be greater than $4ac$, that $a \neq 0$, and that a, b, and c are integers. The values for x will be floating point values. Be sure to have good prompts before all input and to label your output appropriately. You will need to add ***#include <cmath>*** for the power function and the square root function.

`pow( n, m )` returns *n* raised to the *m* power.
sqrt(n) returns the square root of *n*.
m and *n* may be integer or floating point types.

A sample run of your program should look like this (user's input is in ***bold italics***).

```
Enter the coefficient of a: 2
Enter the coefficient of b: 6
Enter the coefficient of c: 3

The first value of x is:  -0.63
The second value of x is: -2.37
```

Using your program, find the roots for the following sets of coefficients.

1. a = 3 b = 4 c = 1 x = -.33, -1
2. a = -1 b = 3 c = 3 x = -.79, 3.79

```
using namespace std;
#include <cmath>
#include <iostream>
#include <iomanip>

int main() {

  cout << setprecision(2);
  cout << fixed;
  cout << showpoint;

  double a = 0;
  double b = 0;
  double c = 0;

  cout << "Enter A: ";
  cin >> a;
  cout << "Enter B: ";
  cin >> b;
  cout << "Enter C: ";
  cin >> c;

  cout << "Zero One: " << ((-b + sqrt(pow(b, 2.0) - (4.0 * a * c))) / (2.0 * a)) << endl;
  cout << "Zero Two: " << ((-b - sqrt(pow(b, 2.0) - (4.0 * a * c))) / (2.0 * a)) << endl;
}
```