PRELAB #3 – CSC 150

Purpose: Demonstrate use of:

- Assignment statements
- Variables
- Input/output
- Typecasting
- Complex mathematical expressions
- 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) int grade
2. exam letter grade (no plus or minus) int letter
3. distance from Earth to Sun (whole meters) int dist
4. your GPA int z

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.

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

```
int z;
cout << "please input x";
cin >> x;
cout << "please input y";
cin >> y;
z = 3y-(y*x*x)
P = z/(2-y)
cout << "The value of P is" << p << endl;
return (0);
```

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

- Chapter 3: Sections 3.4 (page 64)
  - 1. Capital M change it to lower case
  - 2. Remove semicolon after int main ()
  - 3. Change parenthesis to brackets
  - 4. Change capital R to lowercase
  - 5. No errors
- Chapter 4: Sections 4.1 (page 77), 4.3, item 2 & 3 (page 87)
  - Page 77
    - 1. Okay
    - 2. Change single quote to double quote
    - 3. Change double quote to single quote
    - 4. Remove double quotes
    - 5. Remove single quotes
    - 6. Change single quote at end to double quote
  - Page 87
    - 2.
      - B. change – to _
      - D. change name avg is an operator
      - E. remove &
      - G. not allowed to use 1 at beginning
      - I. spaces not allowed
      - J. cannot use #
      - K. reserved word
    - 3.
      - A. need separate lines for declaring variables
      - B. a=b=0 always
      - C. not defining b as double or int
      - D. double quotes around A instead of single
      - E. a char cannot be a digit
      - F. improper declaration, int a, int b, int c, int d
      - G. char is a character not a string of words
- Chapter 5: Section 5.1 (page 110-111)
  - 1a. C++ is very cool
    1b. Hello
    World
  - 1c. Computers are useless.
    They can only give you answers.
    Pablo Picasso
  - 2. C.
  - 3. B.
  - 4. A.
  - 5.
    - A. change >> to <<
    - B. add semicolon at end
    - C. change cin >> to cout << ; cin >>;
    - E. use double quotes around Enter size\n
    - F. use only one \n or \t
    - G. remove “\n”

- H. remove "\n"

- Chapter 6: Sections 6.1 (page 145)
  - 1. What is 5x, are you multiplying?
  - 2. Need to declare a power with power function and implicitly state multiplication
  - 4. Replace 0 with a variable

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. 4.2 5 + c / d
2. 2.50 d / 2.0
3. 0 c % b
4. 1.5 x / w
5. 0.75 x / b
6. 5.00 x + y
7. -3 (c - b) / b
8. 4 a + b * x + w
9. 5 d + c % b
10. INV y % d / w

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. 2 int (y) / x
2. 2.50 double (d) / b
3. 2.50 double (d / b)
4. INV a + d /(double) (b) + w
5. 3 int (y) / int (x)
6. 4.50 int (y) * x

VI. Math functions

Evaluate the following. Assume that <cmath> has been included.

1. 8.0 pow(2.0, 3)
2. 2.0 pow(4.0, 0.5)
3. 2.0 pow(8.0, 1/3)
4. 2.0 pow(8.0, 1.0/3.0)
5. 5.0 sqrt(25.0)

VII. Program.

Write a program that calculates how far a golf ball will travel if it is hit at a given speed and at a given angle. Speed and angle will be input by the user.

The formula is $r = (1/32) * v^2 * \sin(2 * \text{angle})$

This formula determines where the ball will first hit the ground. The ball may bounce or roll but we don't care about that. Just where it first touches the ground.

The sin( ) function takes angles in radians but most people are familiar with degrees.

To convert between degrees and radians, recall that there are 2 PI radians in 360 degrees so multiply the entered angle by 2 PI and divide by 360 to get the number of radians.

For this problem, 0 degrees is horizontal, 90 degrees is straight up. If you hit the ball at an angle greater than 90 degrees, where does the ball go?

To use the sin( ) function, be sure to include the <cmath> library.

Sample runs of the program. User input in ***bold italics***.

```
Enter the speed in feet per second : 100
Enter angle in degrees : 15
You hit the ball 156.25 feet
```

```
Enter the speed in feet per second : 100
Enter angle in degrees : 45
You hit the ball 312.5 feet
```

```
Enter the speed in feet per second : 100
Enter angle in degrees : 85
You hit the ball 54.2651 feet
```

```
Enter the speed in feet per second : 100
Enter angle in degrees : 120
You hit the ball -270.633 feet
```

My program----------

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

int main ()
{
	double r;
	double v;
	double angle;
	double pi;
	double fin;


	fin = 0;
	v = 0;
	r = 0;

	pi = 3.14159265358979323846;

	cout << "Please enter speed (in feet per second): ";
	cin >> v;
	cout << "Please enter angle (in degrees): ";
	cin >> angle;

	fin = (((2 * pi) * angle) / 360);

	r = (((1/32) * (v * v)) * ( sin (2*fin) ));

	cout << "You hit the ball " << r << " feet" << endl;

	return (0);
}
```