**Programming with C Language**

**Tutorial 03**

Q1. Write four different C statements that each add 1 to integer variable x.

* x=x+1
* x++
* x+=1
* ++x

Q2. Write a single C statement to accomplish each of the following:

1. Assign the sum of x and y to z and increment the value of x by 1 after the calculation.

* z= x++ +y

1. Multiply the variable product by 2 using the \*= operator.

* product\*=2

1. Multiply the variable product by 2 using the = and \* operators.

* product= product \* 2

1. Test if the value of the variable count is greater than 10. If it is, print “Count is greater than 10.”

* If (count >10) {

Printf(“Count is Greater than 10\n ”);

}

1. Decrement the variable x by 1, then subtract it from the variable total.

* total - = --x ;

1. Add the variable x to the variable total, then decrement x by 1.

* total += x- - ;

g) Calculate the remainder after q is divided by divisor and assign the result to q. Write this statement two different ways. q = q % divisor; q %= divisor;

h) Print the value 123.4567 with 2 digits of precision. What value is printed?

* printf( “%.2f”, 123.4567 );

i) Print the floating-point value 3.14159 with three digits to the right of the decimal point. What value is printed?

* printf( “%.3f ” , 3.14159);

Q3. Write single C statements that

a) Input integer variable x with scanf. – scanf(“%d”, &x);

b) Input integer variable y with scanf. – scanf(“%d”, &y);

c) Initialize integer variable i to 1. - i=1;

d) Initialize integer variable power to 1.- power=1;

e) Multiply variable power by x and assign the result to power.- power\*x=power; or power\* =x;

f) Increment variable i by 1. – i++

g) Test i to see if it’s less than or equal to y in the condition of a while statement. – while(i<=y)

h) Output integer variable power with printf. – printf(“%d”, power);