Tutorial 03

ID-25645

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

```
x++;
x=x+1;
x+=1;
++X;
```

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

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

```
X=Z+Y
X++
```

(b) Multiply the variable product by 2 using the *= operator.

```
product*=2;
```

c) Multiply the variable product by 2 using the = and * operators.

```
product=product*2;
```

d) 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");
}
```

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

```
x--;
total -= x;
```

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

```
total += x;
x--;
```

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

```
q %= divisor;
```

```
q = q % divisor;
```

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

```
printf("%.2f\n", 123.4567);
//Output will be "123.46"
```

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

```
printf("%.3f\n", 3.14159);
//Output will be "3.142"
```

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

```
int i = 1;
```

d) Initialize integer variable power to 1

```
int power = 1;
```

e) Multiply variable power by x and assign the result to power

```
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("Power = %d\n",power);
```