COMP 248

OBJECT ORIENTED PROGRAMMING

Tutorial #6: While and Do…While Loops – SOLUTIONS

Question 1:

What is the output of the following?

A.

```
int count = 0;
while (count <= 6)
{
        System.out.print(count + " ");
        count = count + 2;
}
System.out.println( );
```

Output:

```
0 2 4 6
```

B.

```
int count = 7;
while (count >= 4)
{
        System.out.print(count + " ");
        count = count - 1;
}
System.out.println( );
```

Output:

```
7 6 5 4
```

C.

```
int i; int j;
boolean again = true;

for (i = 1; i < 5; i++)
{
    again = !again;
    for (j = 1; j < 5; j += 2)
    {
            System.out.print(i + " " + j);
            if (again)
                    System.out.print("-");
            else
                    System.out.print("*");
            }
            System.out.println();
    }
```

Output:

```
1 1*1 3*
2 1-2 3-
3 1*3 3*
4 1-4 3-
```

D.

```
int a = 30;
int b = 3;

while (a >= b)
{
    System.out.println("while " + a + " " + b);
    if ((a % b) == 0)
    {
           a = a / b;
           b++;
    }
    else
    {
           a = a - 1;
           b = b - 1;
    }
}
System.out.println("the end " + a + " " + b);
```

Output:

```
while 30 3
while 10 4
while 9 3
the end 3 4
```

E.

```
int i = 5, count = 0;
while (i != 1)
{
    System.out.println(count + " " + i);
    count++;
    if ((i % 2) == 0)
           i /= 2;
    else
           i = 3 * i + 1;
}
```

Output:

```
0 5
1 16
2 8
3 4
4 2
```

F.

```
boolean sign = true;
int sum = 0;
int n = 0;

while (sum < 30)
{
    if (sign)
           sum = sum + n;
    else
           sum = sum - n;
    System.out.print(sum);
    sign = !sign;
    n = n + 10;
}
```

Output:

```
0-1010-2020-3030
```

G.

```
int x = 0;
while (x != 8);
{
    System.out.print("Hello");
    x = x + 1;
}
```

Answer: Infinite loop.

Since the boolean expression `(x !=8)` is always true, so the loop “`while (x != 8);`” will be execute forever. The key error is that a “`;`” was put right after condition of the while loop. This means that the while loop is ending without any statement.

Correct version:

```
int x = 0;
while (x != 8)
{
    System.out.print("Hello");
    x = x + 1;
}

output:
    HelloHelloHelloHelloHelloHelloHelloHello
```

H.

```
int x = 0;
while (x != 8);
{
    System.out.print("Hello");
}
```

Answer: Infinite loop.

Same problem with the `";"` right after the condition of the while loop as last question and the value of x never changes.

Correct version:

```
int x=0;
while (x != 8)
{
    System.out.print("Hello");
    x=x+1; // Don't forget to increment the control variable
            //"x" inside the statement of while loop.
}
```

Question 2:

Write Java code that uses a ***do…while*** loop that prints even numbers from 2 through 10.

Answer:

```
int number = 2;
do
{
       System.out.println(number);
       number += 2;
}
while (number <= 10);
```

Question 3:

Write Java code that uses a ***while*** loop to print even numbers from 2 through 10.

Answer:

```
int number = 2;
while (number <= 10)
{
      System.out.println(number);
      number += 2;
}
```

Question 4:

Write Java code that uses a ***do while*** loop to sum the numbers from 1 through 50. Display the total sum to the console.

Answer:

```
int sum = 0;
int i = 1;
while (i <= 50)
{
      sum += i++;
}
System.out.println(sum);
```

Question 5:

Write a Java program using a ***while*** loop to find and display the smallest positive integer whose remainder:

- when divided by 3 is 1,
- when divided by 5 is 2, and
- when divided by 7 is 3.

Answer:

```
public static void main(String[] args) {
   int n = 0;
   while (!((n % 3 == 1) && (n % 5 == 2) && (n % 7 == 3)))
      n++;

   System.out.println("The smallest positive integer whose "
         + "remainder when divided by 3 is 1, when divided by 5 "
         + " is 2, and when divided by 7 is 3 is " + n);
}
```