Object Oriented Development with Java

(CT038-3-2-OODJ and Version VC1)

Java Iterative Constructs

Topic & Structure of the lesson

Iteration / Loop

- The while Statement
- The for Statement
- The do … while Statement
- Reading Input In A Loop
- The break statements in Loops

Learning Outcomes

At the end of this topic, you should be able to:

- Write Java programs implementing the looping constructs

Key Terms you must be able to use

If you have mastered this topic, you should be able to use the following terms correctly in your assignments and exams:

- **initialization**
- **termination**
- **increment**

while statement

Generally speaking, a while statement performs some action while a certain condition remains true. The general syntax of the while statement is:

```
while (condition)
        statements
```

That is, while condition is true, do statement.

while statement

If more than 1 statements to be executed :

```
while (condition)   {
                    statement-1;
                    statement-2;
}
```

If we're careless in writing the loop and the condition never becomes false, the loop will continue to execute forever - *an infinite loop.*

while statement

Here is an expanded program from the previous Temperature program. When executed, produces the table of equivalent Centigrade and Fahrenheit.

```
class Temperature {

   public static void main (String[] args) {
     final double
       LOW_TEMP = -10.0,
       HIGH_TEMP = 10.0;
     // The Centigrade ,Fahrenheit temperature.
     double cent, fahr;

       System.out.println("DEGREES C\tDEGREES F");
       cent = LOW_TEMP;
       while (cent <= HIGH_TEMP) {
           fahr = (9.0/5.0) * cent + 32.0;       // Convert C to F
           System.out.println("\t" + cent + "\t\t" + fahr);
           cent = cent + 1.0;   // Increment the Centigrade value.
       }
} }
```

while statement

Example 2

Suppose we want the computer to print :

10 in a bed and the little one said,
"Roll over, roll over."
They all rolled over and one fell out,
9 in a bed and the little one said,
"Roll over, roll over."
They all rolled over and one fell out,
8 in a bed and the little one said,
:
1 in a bed and the little one said,
"Alone at last."

while statement

Example 2

The 1st thing to do is to decide :

1. which pieces of the song are to be printed just once and

2. which pieces will be repeated.

==> the pieces to be printed once will go either before or after the loop;

==> the ones that are repeated can be put inside the loop.

We organize our program schematically as :

while statement

Example 2

We organize our program schematically as :

```
print first line of verse
while (more verses)  {
        print rest of verse
        print first line of next
verse
}
print first line of next verse
```

while statement

Example 2 : Sample Program

```
class TenInABed {

   final int MAX_NUMBER_IN_BED = 10;

   public  static void main (String[] args) {

      int numberInBed;

      System.out.println(MAX_NUMBER_IN_BED + " in a bed and the little one said,");
      numberInBed = MAX_NUMBER_IN_BED - 1;

      while (numberInBed > 0) {
         System.out.println("     \"Roll over, roll over.\"");
         System.out.println("They  all rolled over and one fell out,");
         System.out.println(numberInBed + " in a bed and the little one said,");
         numberInBed = numberInBed - 1;
      }

      System.out.println("     \"Alone at last.\"");
   }
}
```

for statement

Use the for loop when you know the constraints of the loop (its initialization instruction, termination criteria, and increment instruction). The general form of the for statement can be expressed like this:

```
for (initialization; termination; increment)
        statements
```

for statement

initialization ==> a statement that initializes the loop-- its executed once at the beginning of the loop.

termination ==> expression that determines when to terminate the loop. This expression is evaluated at the top of each iteration of the loop. When the expression evaluates to false, the for loop terminates.

increment ==> **expression that gets invoked for each iteration through the loop.**

for statement

For instance, for loops are often used to iterate over the elements in an array, or the characters in a string.

```
// a is an array of some kind
. . .
int i;
int length = a.length;
for (i = 0; i < length; i++) {
   . . .
   // do something to the i th element of a
   . . .
}
```

for statement

Example 1

Let's review at the Temperature program. :

```
while (cent <= HIGH_TEMP) {
    fahr = (9.0/5.0) * cent + 32.0; // Convert C to F
     System.out.println("\t" + cent + "\t\t" + fahr);
       cent = cent + 1.0;   // Increment the Centigrade value.
}
```

for statement

Example 1

Using a for statement, our temperature table loop can be rewritten as

```
for (cent = LOW_TEMP; cent <= HIGH_TEMP; cent =cent+1.0 ) {
        fahr = (9.0/5.0) * cent + 32.0; // Convert C to F
        System.out.println("\t" + cent + "\t\t" + fahr);
    }
```

for statement

Example 1 : Sample Program

```
class Temperature {

    public static void main (String[] args) {
      final double LOW_TEMP = -10.0, HIGH_TEMP = 10.0;

      double cent, fahr;

        System.out.println("DEGREES  C\tDEGREES F");
        cent = LOW_TEMP;
        for (cent = HIGH_TEMP; cent <= HIGH_TEMP; cent =cent+1.0 ) {
            fahr = (9.0/5.0) * cent + 32.0; // Convert C to F
            System.out.println("\t"  + cent + "\t\t" + fahr);
        }
    }
}
```

do … while statement

Java provides another loop, the **do-while loop**, which is similar to the while loop :

```
do {
        statements
} while (condition);
```

do-while Vs while : The main difference between these two is that in *do-while loop the expression is evaluated at the bottom of the loop*, thus, the body of the loop is always executed AT LEAST ONCE.

do … while statement

The do-while statement is a less commonly used loop construct in programming but does have its uses. For example, the do-while is convenient to use when the statements within the loop must be executed at least once.

do … while statement

Example 1

When reading information from a file, you know that you will always have to read at least one character. Therefore, the do-while loop will be appropriate to be used in this case :

```
int c;
Scanner in = new Scanner(System.in);
. . .
do {
      c = in.nextInt();
       . . .
} while (c != -1);
```

Quick Review Question

Write and run a program that prints the average of 5 random double value. Your output should look like this :

Average = 0.533764716823967

(hint : use the method Math.random())

Quick Review Question

Sample answer

```
public class averageRandom  {

    public static void main(String[] args)  {
            double randNo, sum=0.0;
            for(int i=0; i<5; i++)  {
                    randNo = Math.random();
                    sum = sum + randNo;


            System.out.println(randNo+"");
            }
            System.out.println("average = " +
sum/5);
    }
}
```

break statement

As in switch statements, the break statement can also be used *terminate the execution of the iteration* statements while and for.

The use of a break in loops can simplify writing code in what is sometimes called the loop-and-a-half problem. For example :

Quick Review Questions

Write an application that accepts a sequence of inputs that describe the quantities and types of coins held by a person. The application should then process the inputs and display the total value of these coins.

Note:

Half-dollar = 50 cents

Quarter = 25 cents

Dime = 10 cents

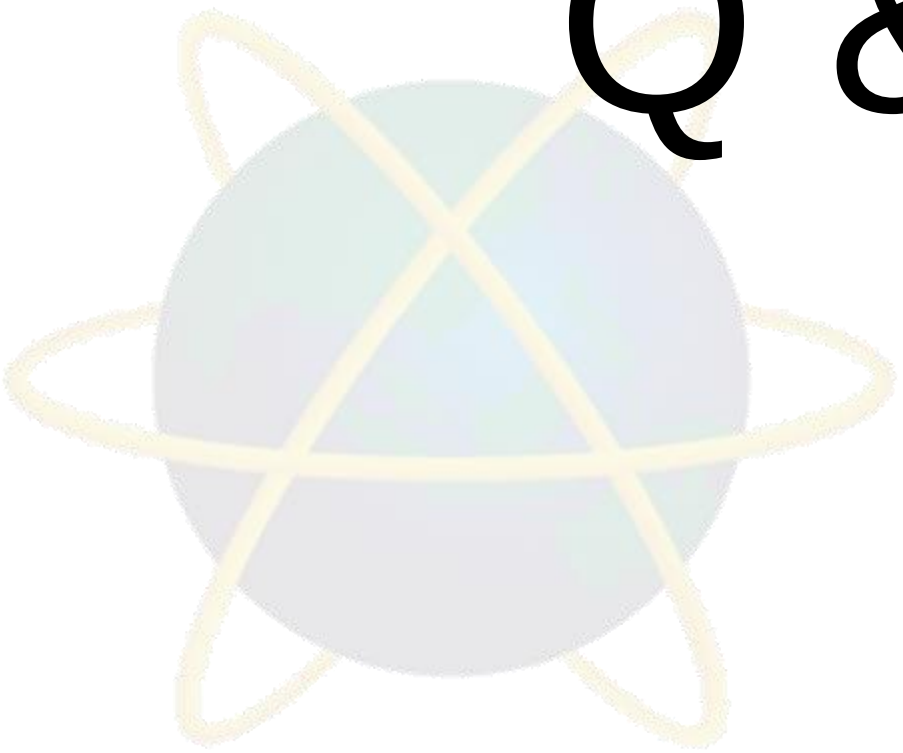
Nickel = 5 cents

Summary of Main Teaching Points

Iteration / Loop

- The while Statement
- The for Statement
- The do … while Statement
- Reading Input In A Loop
- The break statements in Loops

Question and Answer Session



Q & A

Next Session

-Objects

-Class

-Principles of Object Orientation

-Package