Tutorial 4
CSC 201

Java Programming Concepts

مبادئ البرمجة باستخدام الجافا

Chapter 5

1. **Iterative Statements** الجمل التكرارية
   - What is an Iterative Statement?
   - Java's Iterative Statements
2. **The while Loop** while حلقة
   - Purpose
   - General Syntax
   - A Real-life Situation
   - Examples
3. **The do-while Loop** do-while حلقة
   - General Syntax
   - Examples
4. **The For Loop** for حلقة
   - General Syntax
   - Examples
5. **Comparing for & while** for & while مقارنة
   - General form
   - Examples
6. **Nested Loops** حلقات متداخلة
   - What is a Nested Loop?
   - Examples
7. **Common Loop Errors** الأخطاء الشائعة فى الحلقات
   - Infinite Loops
   - Off by One
   - Empty Intervals

Iterative Statements

الجمل التكرارية

What is an Iterative Statement ?

An iterative statement is a type of control structure. An Iterative Statement allows the repetition of a block of statements according to a condition

نوع من هياكل التحكم. الجملة التكرارية تسمح بتكرار تنفيذ جملة / مجموعة جمل (بلوك) تبعا لشرط معين

Java provides the following Iterative Statements:

- The while loop
- The do-while loop
- The for loop

While Loop

General Syntax:

```
while (expression)
 {
   //body of loop
 }
```

The purpose of the while loop is to repeat a block of statements while the condition (expression) is true.

الغرض منها تكرار مجموعة جمل طالما الشرط متحقق (التعبير المنطقي قيمته صحيح)

A Simple Example:

```
public class WhileExample
{
    public static void main(String args[])
  {
    int n=1;
    while (n<=10)
     {
       System.out.println("iteration# " +n);
       n++;
     }
    System.out.println("Done with loop ");
  }
}
```

Program Output:

```
iteration# 1
iteration# 2
iteration# 3
iteration# 4
iteration# 5
iteration# 6
iteration# 7
iteration# 8
iteration# 9
iteration# 10
Done with loop
```

While Loop

بعد معرفة الفكرة الاساسية يمكن استعراض بعض الأمثلة

Now that we know the general idea, let's take a look at some examples:

Example1

In this example, the program will print the multiples of 15 from 0 to 100.

برنامج يطبع مضاعفات العدد ١٥ من ٠ إلى ١٠٠

```
public class WhileExample2
{
    public static void main(String[] args)
   {
        int number = 0;
        while (number <= 100)
        {
           if (number % 15 == 0)
           System.out.println(number + " is a multiple of 15");
           number++;
        }
    }
}
```

Program Output:

```
 0  is a multiple of 15
15 is a multiple of 15
30 is a multiple of 15
45 is a multiple of 15
60 is a multiple of 15
75 is a multiple of 15
90 is a multiple of 15
```

Tip: Try changing this program to print the multiples of any number entered by the user.

جرب تغيير البرنامج ليطبع مضاعفات أي عدد آخر

While Loop

Example2

In this example, we want to print the sum of the numbers from 1 to n.

برنامج يطبع مجموع الاعداد من ١ إلى ن

```
import java.io.*;
public class WhileExample2
{
   public static void main(String[] args) throws IOException
   {
      int n;       int i=1;       int sum = 0;

      InputStreamReader inStream = new InputStreamReader( System.in );
      BufferedReader stdin = new BufferedReader( inStream );

     System.out.print("Enter n: ");
     n = Integer.parseInt(stdin.readLine());
      while (i<=n)
      {
         sum += i;
         i++;
      }
      System.out.println("The sum is: "+sum);
   }
}
```

Sample Run:

Enter n: 10

The sum is: 55

Tip: try different values for n and observe the output.

جرب قيم مختلفة لـ ن ولاحظ الناتج

While Loop

Example3

In this example, if you invested 10000 euros with 5% compounded annually. You would like to print out after how many years will this amount be doubled.

في هذا البرنامج إذا كنت استثمرت ١٠٠٠٠ يورو بفائدة ٥٪ مركبة سنويا.
نريد طباعة بعد كم سنة سيتضاعف المبلغ

```
public class WhileExample3
 {
    public static void main(String[] args)
   {
      double balance = 10000;
      double rate = 0.05;
      double targetBalance = 20000;
      int year = 0;
      while (balance < targetBalance)
                {
                     year++;
                     double interest = balance * rate;
                      balance += interest;
                }
      System.out.println("The investment will be doubled after "+year+" years");

   }
 }
```

Program Output:
The investment will be doubled after 15 years

Tip: Try generalizing this program to work for any interest rate, entered by the user.

حاول تعميم البرنامج ليعمل على أي معدل فائدة يقوم المستخدم بإدخاله

Do-while Loop

General Syntax

```
do
    {
        //body of loop
    }
while (expression);
```

The main difference between the while loop and the do-while loop is that the while loop checks that the condition is true first and repeats the block of statements meanwhile. Whereas the do-while loop performs the block of statements at least once regardless of the condition's initial value.

الفرق الأساسي بين حلقة while وحلقة do-while أن الأولى تتحقق من صحة الشرط أولا وتكرر تنفيذ البلوك طالما ظل الشرط صحيحا. بينما الثانية تنفذ البلوك على الاقل مرة واحدة بصرف النظر عن قيمة الشرط المبدئية

A Simple Example: Here, we want to sum the numbers from 1 to n.

```
public class DoWhileExample {
    public static void main(String[] args) {
     int i=0;
     int n=10;
     int sum = 0;
     do
     {
        sum += i;
        i++;
     }
     while (i<=n);
     System.out.println("Sum of numbers from 1 to "+n+" is "+sum);
  }
}
```

Sample Runs:
Run#1:
Enter n: 10
Sum of numbers from 1 to 10 is 55
Run#2:
Enter n: 15
Sum of numbers from 1 to 15 is 120

Tip: Try using other values for n, and observe the output.

جرب استخدام قيم أخرى لـ ن ولاحظ الناتج

Do-while Loop

Example 1

In this example, we want to print the average score of n scores.

```
Import java.io.*
public class DoWhileExample
{
   public static void main(String[] args) throws IOException
 {
    InputStreamReader inStream = new InputStreamReader( System.in );
    BufferedReader stdin = new BufferedReader( inStream);
    int i=0;
    int n;
    int sum = 0;
    double avg;
    System.out.print("Enter the Number of scores: ");
    n = Integer.parseInt(stdin.readLine());
    do
    {
      System.out.print("Enter the score: ");
      x = Integer.parseInt(stdin.readLine());
```

Do-while Loop

Example 1 - continued

```
      sum += x;
      i++;
     }
     while (i<=n);
     avg=sum/n;
     System.out.println(" The average of these "+n+" numbers is " +avg);
  }
}
```

Sample Runs:

Run#1:

```
Enter number of scores: 2
Enter score#1:7
Enter score#2:10
The average of these 2 numbers is 8.5
```

Run#2:

```
Enter number of scores: 5
Enter score#1:5
Enter score#2:8
Enter score#3:6
Enter score#4:4
Enter score#5:3
The average of these 5 numbers is 5.2
```

Do-while Loop

Example2: Verify that the user has entered a positive value.

```
import java.io.*;
public class DoWhileExample2
{
   public static void main(String[] args) throws IOException
  {
     InputStreamReader inStream = new InputStreamReader( System.in );
     BufferedReader stdin = new BufferedReader(inStream);
     int value;
     do
     {
       System.out.println("Please enter a positive value: ");
       value = Integer.parseInt(stdin.readLine());

     }
     while (value <= 0);
     System.out.println("Thank You!");
   }
}
```

Sample Run:

```
Please enter a positive value:
-1
Please enter a positive value:
-8
Please enter a positive value:
-7
Please enter a positive value:
5
Thank You!
```

For Loop

General Syntax

```
for (initialization;condition;update)
{
  // body of loop
}
```

حلقة for تقوم بتنفيذ جملة او مجموعة جمل (بلوك) عدد من المرات
تتكون من ثلاثة أجزاء: القيمة المبدئية – شرط الاستمرار– تعديل القيمة

A Simple Example

The following statement will print the numbers from 0 to 10.

```
for (x=0;x<10;x++)
    System.out.println("Count: "+x);
```

Program Output:
Count: 0
Count: 1
Count: 2
Count: 3
Count: 4
Count: 5
Count: 6
Count: 7
Count: 8
Count: 9
Count: 10

For Loop

Example1

In this example, we want to print n asterisks.

```
public class ForExample1 {
    public static void main(String[] args) {
        int x,n;
        n = 5;
        for (x=0;x<n;x++)
            System.out.print("* ");
   }
}
```

Example 2

In this example, we will compute the sum of numbers from 1 to n.

```
public class ForExample2{
   public static void main(String[] args) {
     int n=10;
     int x;
     int sum=0;
     for (x=0;x<=n;x++)
        sum += x;
     System.out.println("Sum is "+sum);
   }
}
```

For Loop

Example 3

In this example, we will compute the factorial of any number n

```
public class ForLoopExample3
{
 public static void main(String[] args) throws IOException
 {
   int n=7;
   int fact = 1;
   if (n!=0)
   {
     for (int x=n;x>1;x--)
        fact *= x;
    }
   System.out.println("The Factorial of "+n+" is "+fact);
 }
}
```

Sample Runs:

Run#1:
The Factorial of 5 is 120

Run#2:
The Factorial of 3 is 6

Run#3:
The Factorial of 7 is 5040

Comparing For and while Loops

General Form

while	for

initialization;
while(condition)
{
  loop body;
  loop advancement;
}
	for (initialization; condition; loop advancement)
	{
	  loop body
	}

Comparing For and while Loops

Example1

Let us compare the investment example we previously discussed using the two loop structures.

while

```
int year = 0;
while (balance < targetBalance)
{
  year++;
  double interest = balance * rate / 100;
  balance += interest;
 }
```

for

```
for (int year =0; balance<targetBalance; year++)
{
   double interest = balance * rate/100;
   balance += interest


}
```

Example2

Let us now compare the n asterisks example

for

```
for (x=0;x<n;x++)
       System.out.print("* ");
```

while

```
x=0;
while (x<n)
 {
      System.out.println("*  ");
      x++;
  }
```

Nested Loops

What is a Nested Loop?

الحلقات المتداخلة: هي حلقة تقع داخل حلقة أخرى

A nested loop is a loop that lies inside another loop.

Example1

This is an example of a nested loop using the for loop. The aim of the program is to print an n-by-n checkerboard using the hash symbol (#).

```
public class NestedForLoopExample
{
  public static void main(String args [])
  {
    int n = 5;

     for(int x = 0; x < n; x++)
     {
       for(int y = 0; y < n; y++)
          System.out.print((x%2==0)?" #":"# ");
       System.out.println();
      }
  }
}
```

Sample Run:

```
 # # # # #
# # # # #
 # # # # #
# # # # #
 # # # # #
```

Nested Loops

Example2

In this program, we want to draw a triangle pattern.

```
public class Triangle
{
    public static void main(String[] args)
   {
        String r = " ";
        int n = 4;
        for (int i=1;i <= n;i++)
        {
           for(int j=1;j <= i;j++)
           r = r + "[]";
           r = r + "\n";
         }
         System.out.println(r);
    }
}
```

Program Output:

[]
[][]
[][][]
[][][][]

Tip: Can you re-write the above two programs once with **while**, and once with **do-while** ?

Common Loop Errors

الأخطاء الشائعة

1) Infinite Loops

الحلقات اللانهائية

This type of error occurs when the condition is always true.

Example

```
x=0;
while (x>0)
   {
      y--;
   }
```

2) Off-by-One

تنفيذ الحلقة ناقصة مرة واحدة

This type of error occurs by executing the loop either more or less times than the desired number of times. In the example of summing 1 to n numbers, if we had changed the condition from (i<=n) to (i<n) as follows:

```
while (i<n)
     {
        sum += i;
        i++;
      }
```

Here, the loop is executed one time less than we actually wanted it to.

3) Empty Intervals

الحلقات الفارغة – شرط الاستمرار غير صحيح من البداية

This type of error occurs when the initial condition is false, thus the loop body is never executed.

Example

```
      for (int x=0;x>3;x++)
         System.out.println("X is "+x);
```