Course code : **CSE1004**

Course title : **Problem Solving using Java**

Java – Iterative Statements

Objectives

This session will give the knowledge about

- Iterative Statements
- while
- do… while
- for
- enhance for
- continue statements

Iterative Statements

- Iterative Statements are statements which executes one or set of statements based on a condition for multiple times.

- There are three types of Conditional Statements in java
  - while
  - do… while
  - for
  - enhanced for

while loop

Syntax

```
while(condition)
{
        Body of the loop
}
```

while loop – Example

```
/* This is an example of a while loop*/

public class WhileDemo
{
    public static void main(String args[])
    {
        int i = 0;
        while (i < 5)
        {
            System.out.println("i: "+i);
            i++;
        }
    }
}
```

do-while loop

Syntax:

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

do…. while loop – Example

```
/* This is an example of a do….while loop*/

    public class DowhileDemo {
        public static void main(String[ ] args) {
                int i =5;
                do {
                        System.out.println("i: "+i);
                        i = i + 1;
                } while (i < 5);
        }
    }
```

for loop

Syntax

```
for(initialization; condition; increment/decrement)
{
        Body of the loop
}
```

for loop - Example

```
/* This is an example of a for loop*/

public class ForDemo {
        public static void main(String[] args) {
                for (int i=1; i<=5; i++ )
                {
                        System.out.println("i: "+i);
                }
        }
}
```

Enhanced for loop

Syntax:

```
for(declaration : expression)
{
    Body of loop
}
```

Enhanced for loop - Example

```
/* This is an example of a enhanced for loop */
public class Sample {
    public static void main(String[] args)
    {
        int [] numbers = {10, 20, 30, 40, 50};
        for(int i : numbers )                    // for(int i=0; i<numbers.length(); i++)
        {
            System.out.println("i: "+i );        // System.out.println( "i: "+numbers[i] );
        }
    }
}
```

continue statement

- The continue statement skips the current iteration of a loop
- In while and do loops, continue causes the control to go directly to the test condition and then continue the iteration process
- In case of for loop, the increment section of the loop is executed before the test-condition is evaluated

continue - Example

```
/* This is an example of a continue statement*/

public class ContinueDemo {
        public static void main(String[] args) {
                int [] numbers = {10, 20, 30, 40, 50};
                for(int i : numbers )
                {
                        if( i == 3 )
                                continue;
                }
                System.out.println( "i: "+i );
        }
}
```

Quiz - 1

What will be the result, if we try to compile and execute the following code ?

```
class Sample{
    public static void main(String[] args) {
        boolean b = true;
        if(b){
                System.out.println(" if block ");
        }
        else {
                System.out.println(“ else block ");
        }
  }
}
```

Quiz - 2

What would be the output if we try to compile and execute the following java code?

```
class Sample {
    public static void main(String[] args)
    {
        while(false)
            System.out.println("while loop");
    }
}
```

Quiz - 3

What would be the output if we try to compile and execute the following java code?

```
class Sample {
    public static void main(String[] args) {
        for( ; ; )
            System.out.println("For loop");
    }
}
```

Summary

We have discussed about

- Iterative Statements
- while
- do… while
- for
- enhance for
- continue statements