Iteration Statements

Java iteration statements are used to repeat the set of statements until the condition of the termination is true. There are three types of iteration statements in java.

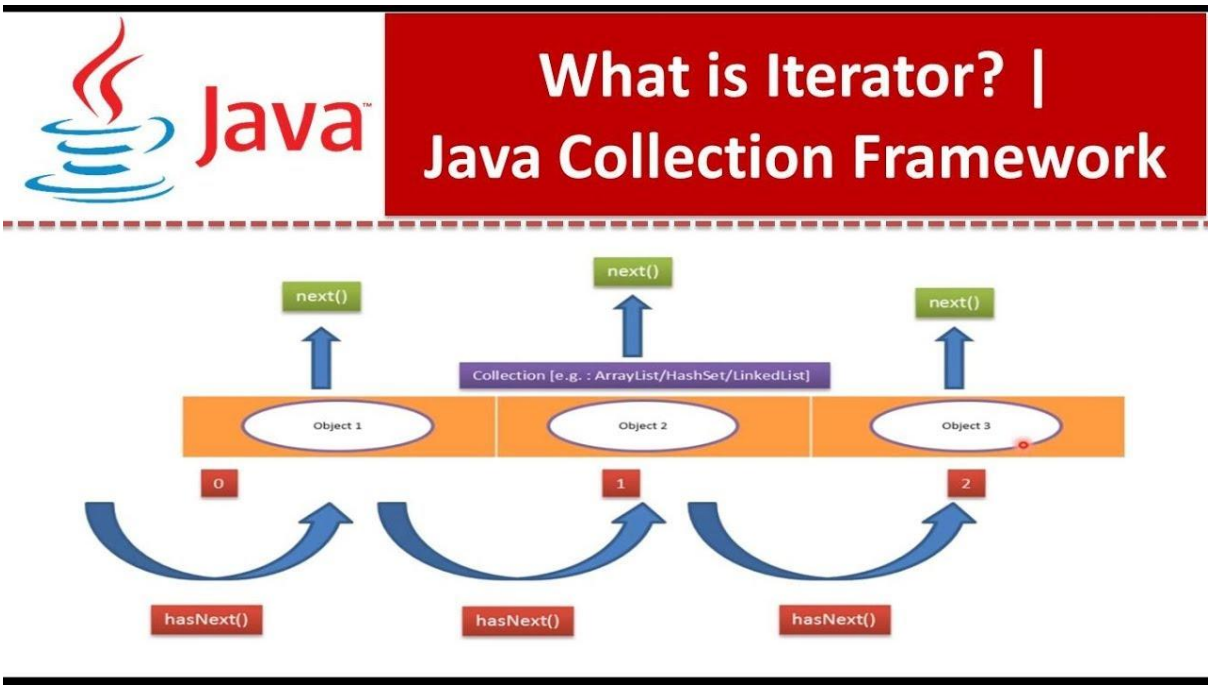
i) while loop: Java while loop is used to repeat a statement or block of the statement until a condition is true. We can use a while loop if the number of iterations is not fixed. The condition of a while loop is any boolean expression. The loop will run till the condition is true, if the condition becomes false the control goes to the next statement immediately following the loop.

Syntax:

```
initialization;
while (condition) {
    // statements


    update_expression;
}
```

To better understand, look at the diagram given below:



Example:

```
public class JavaWhileLoop {
   public static void main(String arg[]) {

      // Initialization
      int i = 1;


       // While loop run till the
       // condition is true
       while(i < = 10) {
           System.out.print(i + " ");


           // Increment the value by 1.
           i++;
       }
    }
}
```

Output:

1 2 3 4 5 6 7 8 9 10

ii) do-while loop: Java do-while loop is also used to repeat a statement until a condition is true. Sometimes in our program we want to execute the body of the loop at least once, even if the conditional expression is false, in other words, there are times when you would like to test the conditional expression at the end of the loop rather than the beginning. Then we should go for a do-while loop in java. It executes the body loop at least once because the conditional expression is checking at the end. Let's look at the example to understand this.

Syntax:

```
initialization;
do {
    // statements
    // update_expression;
}
while (condition);
```

Example:

```
public class JavaDoWhileLoop {
  public static void main(String arg[]) {

      // Initialization
      int i = 1;
      do {
         System.out.print(i + " ");
         i++;
```

```
        }

        // Checking condition at the end
        while(i <= 10);
    }
}
```

Output:

1 2 3 4 5 6 7 8 9 10

iii) for loop: The for loop in java is used to iterate a part of the program several times. It consumes the initialization, condition checking, and increment/decrement a value in one line. If the number of iterations is fixed then it is recommended you use Java for loop.

Syntax:

```
for(initialization condition; testing condition; increment/decrement) {
    // statements
}
```

Example:

```
public class JavaForLoop {
  public static void main(String args[]) {

      // for loop to print the value
      // from 1 to 10
      for(int i = 1; i <= 10; i++)
          System.out.print(i + " ");
  }
}
```

Output:

1 2 3 4 5 6 7 8 9 10

Java Enhanced for loop: Java Enhanced for loop provides a simpler way to iterate through the elements of a collection or array. It is used when we need to sequentially iterate through elements without knowing the index of the currently processing element.

Syntax:

```
for(T element: collection obj / array) {
    // statements
}
```

Example:

```
public class JavaEnhancedForLoop {
   public static void main(String arg[]) {


      // array of string
      String array[] = {“Coding”, “Ninjas”, “Welcomes”, “You”};


      // Enhanced for loop
      for(String x : array) {
         System.out.println(x);
      }
   }
}
```

Output:

Coding
Ninjas
Welcomes
You