Understanding for-each loop

- For each Introduced in 1.5version, It acts as an alternative to for loop, while loop etc for retrieving the elements in the array.
- By using for-each loop we can easily retrieve the elements easily from the arrays.
- It cannot traverse the elements in reverse order because it does not work on index values
- For-each also acts as alternative for iterator when retrieving elements from collections.

Syntax:

- Step 1 :-Declare a variable that is the same type as the base type of the array
- Step 2 :-Write the Colon (:)
- Step 3 :-Then write the array name
- Step 4 :-In the loop body we have to use the variable which we have created

For-each loop is also known as enhanced for loop.

```
1  package com.pack1;
2
3  public class ClassA
4  {
5      void meth1()
6      {
7          System.out.println("meth1() called");
8
9          String names[]=new String[5];
10
11         names[0]="Sujatha";
12         names[1]="Anjum";
13         names[2]="Lakshmi";
14         names[3]="Cristine";
15         names[4]="Athena";
16
17         names[5]="Mounica";
18
19
20     }
21     public static void main(String[] args)
22     {
23         ClassA aobj=new ClassA();
```

```
meth1() called
Exception in thread "main" java.lang.Arr
        at Training/com.pack1.ClassA.met
        at Training/com.pack1.ClassA.mai
```


Core Java Class-40
Iteration-(for-each)
Jump- (break, continue &return)

```
meth1() called
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for len
        at Training/com.pack1.ClassA.meth1(ClassA.java:17)
        at Training/com.pack1.ClassA.main(ClassA.java:24)
```

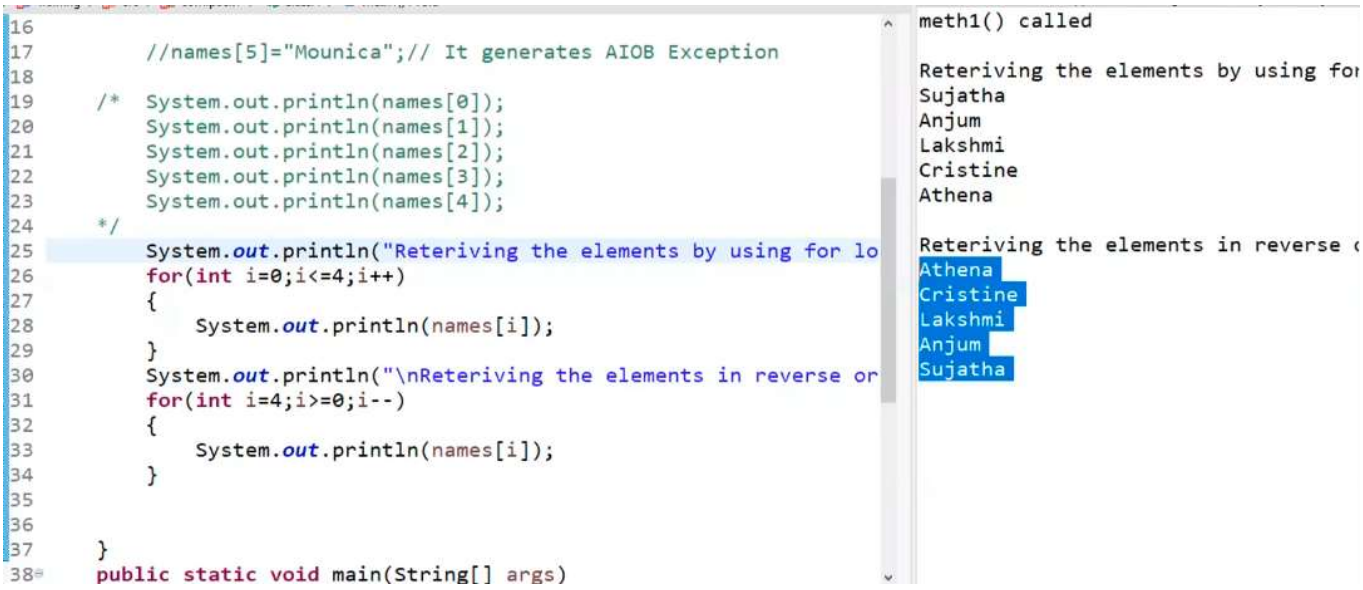
```
6       {
7           System.out.println("meth1() called\n");
8
9           String names[]=new String[5];
10
11          names[0]="Sujatha";
12          names[1]="Anjum";
13          names[2]="Lakshmi";
14          names[3]="Cristine";
15          names[4]="Athena";
16
17          //names[5]="Mounica";// It generates AIOB Exception
18
19          System.out.println(names[0]);
20          System.out.println(names[1]);
21          System.out.println(names[2]);
22          System.out.println(names[3]);
23          System.out.println(names[4]);
24
25
26      }
27      public static void main(String[] args)
28      {
```

```
meth1() called

Sujatha
Anjum
Lakshmi
Cristine
Athena
```

```
9           String names[]=new String[5];
10
11          names[0]="Sujatha";
12          names[1]="Anjum";
13          names[2]="Lakshmi";
14          names[3]="Cristine";
15          names[4]="Athena";
16
17          //names[5]="Mounica";// It generates AIOB Exception
18
19      /*  System.out.println(names[0]);
20          System.out.println(names[1]);
21          System.out.println(names[2]);
22          System.out.println(names[3]);
23          System.out.println(names[4]);
24      */
25          System.out.println("Reteriving the elements by using for lo
26          for(int i=0;i<=4;i++)
27          {
28              System.out.println(names[i]);
29          }
```

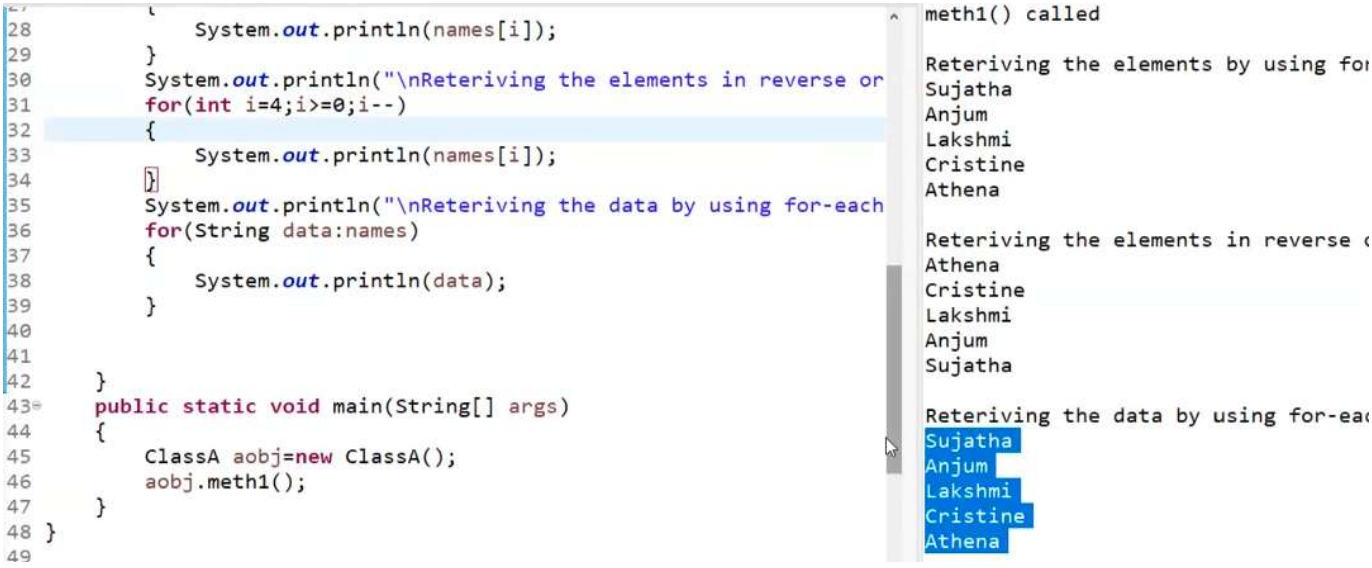
```
meth1() called

Reteriving the elements by using for
Sujatha
Anjum
Lakshmi
Cristine
Athena
```


Core Java Class-40
Iteration-(for-each)
Jump- (break, continue &return)



```
16
17          //names[5]="Mounica";// It generates AIOB Exception
18
19      /*  System.out.println(names[0]);
20          System.out.println(names[1]);
21          System.out.println(names[2]);
22          System.out.println(names[3]);
23          System.out.println(names[4]);
24      */
25          System.out.println("Reteriving the elements by using for lo
26          for(int i=0;i<=4;i++)
27          {
28              System.out.println(names[i]);
29          }
30          System.out.println("\nReteriving the elements in reverse or
31          for(int i=4;i>=0;i--)
32          {
33              System.out.println(names[i]);
34          }
35
36
37      }
38      public static void main(String[] args)
```

```
meth1() called

Reteriving the elements by using for
Sujatha
Anjum
Lakshmi
Cristine
Athena

Reteriving the elements in reverse
Athena
Cristine
Lakshmi
Anjum
Sujatha
```



```
28              System.out.println(names[i]);
29          }
30          System.out.println("\nReteriving the elements in reverse or
31          for(int i=4;i>=0;i--)
32          {
33              System.out.println(names[i]);
34          }
35          System.out.println("\nReteriving the data by using for-each
36          for(String data:names)
37          {
38              System.out.println(data);
39          }
40
41
42      }
43      public static void main(String[] args)
44      {
45          ClassA aobj=new ClassA();
46          aobj.meth1();
47      }
48  }
49
```

```
meth1() called

Reteriving the elements by using for
Sujatha
Anjum
Lakshmi
Cristine
Athena

Reteriving the elements in reverse
Athena
Cristine
Lakshmi
Anjum
Sujatha

Reteriving the data by using for-eac
Sujatha
Anjum
Lakshmi
Cristine
Athena
```

For loop uses index positions of an array to retrieve the data

So, we will be able to retrieve the data both in forward and forward directions from any array by using for loop.

For each loop we will work on streaming data

So, we will not be able to retrieve the data in a backward direction from an array.

The only use of for-each loop is to retrieve the data from array or from collection classes.

```
3  public class ClassA
4  {
5      void meth1()
6      {
7          System.out.println("meth1() called\n");
8
9          String names[]=new String[5];
10
11         names[0]="Sujatha";
12         names[1]="Anjum";
13         names[2]="Lakshmi";
14         names[3]="Cristine";
15         names[4]="Athena";
16
17         //names[5]="Mounica";// It generates AIOB Exception
18
19     /*  System.out.println(names[0]);
20         System.out.println(names[1]);
21         System.out.println(names[2]);
22         System.out.println(names[3]);
23         System.out.println(names[4]);
24     */
25         System.out.println("Reteriving the elements by using for loop");
```

```
26          for(int i=0;i<=4;i++)
27          {
28              System.out.println(names[i]);
29          }
30          System.out.println("\nReteriving the elements in reverse order by using for loop");
31          for(int i=4;i>=0;i--)
32          {
33              System.out.println(names[i]);
34          }
35          System.out.println("\nReteriving the data by using for-each loop");
36          for(String data:names)
37          {
38              System.out.println(data);
39          }
40      }
41      public static void main(String[] args)
42      {
43          ClassA aobj=new ClassA();
44          aobj.meth1();
45      }
46  }
```

```
/*
 for loop uses index positions of an Array to reterive the data.
 So we will be able to reterive the data in both forward & backward
 directions from an Array by using for loop.

 foreach loop will be working on STREAMING data so we will NOT be
 able to reterive the data in a backward direction from an Array.
 THe ONLY use for for-each loop is to reterive the data from Array (or) from Collection classes
 */
```

Jump Statements

- Java jump statements enable transfer of control to other parts of program.
- Java provides three jump statements:
  1) break
  2) continue
  3) return
- In addition, Java supports exception handling that can also alter the control flow of a program.

Understanding break statement

- The break statement is used to jump out of a loop.
- When the break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.
- We can us break statement inside "switch", "loops" & "labeled blocks"., other than this if you are using any where you will be getting an compile time error.

Understanding continue statement

- Continue statement is used to skip current iteration and continue for the next iteration in the loop.
- We can us continue statement inside "loops" & "labeled blocks"., other than this if you are using any where you will be getting an compile time error.


```
2
3  public class ClassA
4  {
5      void meth1()
6      {
7          System.out.println("meth1() called");
8          for(int i=1;i<=10;i++)
9          {
10             if(i==5)
11             {
12                 break;
13             }
14             System.out.println("i value : "+i);
15         }
16         System.out.println("Compiler came out from for-loop");
17     }
18     public static void main(String[] args)
19     {
20         System.out.println("Start");
21         new ClassA().meth1();
22         System.out.println("End");
23     }
24 }
```

```
Start
meth1() called
i value : 1
i value : 2
i value : 3
i value : 4
Compiler came out from for-loop
End
```

```java
public class ClassA
{
    void meth1()
    {
        System.out.println("meth1() called");
        for(int i=1;i<=10;i++)
        {
            if(i==5)
            {
                //break;
                continue;
            }
            System.out.println("i value : "+i);
        }
        System.out.println("Compiler came out from for-loop");
    }
    public static void main(String[] args)
    {
        System.out.println("Start");
        new ClassA().meth1();
        System.out.println("End");
    }
```

```
Start
meth1() called
i value : 1
i value : 2
i value : 3
i value : 4
i value : 6
i value : 7
i value : 8
i value : 9
i value : 10
Compiler came out from for-loop
End
```

```java
    void meth1()
    {
        System.out.println("meth1() called");
        for(int i=1;i<=10;i++)
        {
            if(i==5)
            {
                //break;
                //continue;
                return;
            }
            System.out.println("i value : "+i);
        }
        System.out.println("Compiler came out from for-loop");
    }
    public static void main(String[] args)
    {
        System.out.println("Start");
        new ClassA().meth1();
        System.out.println("End");
    }
}
```

```
Start
meth1() called
i value : 1
i value : 2
i value : 3
i value : 4
End
```

```java
package com.pack1;

public class ClassA
{
    void meth1(int num1, int num2)
    {
        for(;num1<=num2;num1++)
        {
            if(num1%2==0)
                System.out.println(num1+" is even");
        }
    }
    public static void main(String[] args)
    {
        new ClassA().meth1(1, 75);
    }
}
```

```
2 is even
4 is even
6 is even
8 is even
10 is even
12 is even
14 is even
16 is even
18 is even
20 is even
22 is even
24 is even
26 is even
28 is even
30 is even
32 is even
34 is even
36 is even
38 is even
40 is even
42 is even
44 is even
46 is even
```

Need to print even numbers in between

```java
package com.pack1;

public class ClassA
{
    void meth1(int num1, int num2)
    {
        int count =1;
        for(;num1<=num2;num1++)
        {
            if(num1%2==0)
                System.out.println(num1+" is even "+"("+count+++")");
        }
    }
    public static void main(String[] args)
    {
        new ClassA().meth1(1, 75);
    }
}
```

```
2 is even (1)
4 is even (2)
6 is even (3)
8 is even (4)
10 is even (5)
12 is even (6)
14 is even (7)
16 is even (8)
18 is even (9)
20 is even (10)
22 is even (11)
24 is even (12)
26 is even (13)
28 is even (14)
30 is even (15)
32 is even (16)
34 is even (17)
36 is even (18)
38 is even (19)
40 is even (20)
42 is even (21)
44 is even (22)
46 is even (23)
```

```java
package com.pack1;

public class ClassA
{
    void meth1(int num1, int num2)
    {
        int count =1;
        for(;num1<=num2;num1++)
        {
            if(num1%2==0)
                System.out.println(num1+" is even "+"("+count+++")");
        }
    }
    public static void main(String[] args)
    {
        new ClassA().meth1(1, 10);
    }
}
```

```
<terminated> ClassA [Java Applic
2 is even (1)
4 is even (2)
6 is even (3)
8 is even (4)
10 is even (5)
```

Print 2 table

```java
package com.pack1;

public class ClassA
{
    void meth1(int num1)
    {
        for(int i=1;i<=10;i++)
            System.out.println(num1+" * "+i+" = "+(num1*i));
    }
    public static void main(String[] args)
    {
        new ClassA().meth1(2);
    }
}
```

```
<terminated> ClassA [Java Appl
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
```

Print tables from 2 to 5

```java
package com.pack1;

public class ClassA
{
    void meth1(int num1)
    {
        for(int i=1;i<=10;i++)
            System.out.println(num1+" * "+i+" = "+(num1*i));
    }
    void meth2(int num1,int num2)
    {
        for(;num1<=num2;num1++)
        {
            for(int i=1;i<=10;i++)
                System.out.println(num1+" * "+i+" = "+(num1*i));
            System.out.println("------------");
        }
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        //aobj.meth1(2);
        aobj.meth2(2, 5);
```

```
<terminated> ClassA [Java App
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
------------
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30
------------
```

```java
package com.pack1;

public class ClassA
{
    void meth1()
    {
        System.out.println("hi");
        new ClassA().meth1(); // RECURSION
    }
    public static void main(String[] args)
    {
        new ClassA().meth1();
    }
}
```

```
hi
hi
hi
hi
hi
hi
hi
hi
hi
hi
hi
hi
hi
hi
hi
```

Assignment

print numbers from 1 to 100 numbers without using loops (while, do-while, for loop).

1) Write a Java program to print the numbers from 1 to 10 using a for loop.
2) Write a Java program to print the even numbers between 1 and 20 using a for loop
3) WAP with a parameterized method having a positive integer as a parameter. It should then print the multiplication table of that number.

2*1=2
2*2=4
.
.
.
2*10=20

4)Write a Java program to print the sum of the first 100 natural numbers using a for loop.
5)Write a Java program to find the factorial of a number using a for loop.

6)Write a Java program to find the prime numbers between 1 and 100 using a for loop

7)Write a Java program to print the reverse of a given number using a for loop.