For Each Loop :
* It is an metaword for loop which is introduced from JDK 1.5V.
* It is an improvement over classical for loop, actually java compiler will automatically generate classical for loop for "for-each loop".

ForEach.java

```
public class ForEach
{
    public static void main(String [] args)
    {
        int arr[] = { 10,20,30};

        for(int x : arr)
        {
            System.out.println(x);
        }
    }
}
```

ForEach class

```
public class ForEach
{
    public static void main(String [] args)
    {
        int arr[] = { 10,20,30};

        for(int i=0; i<arr.length; i++)
        {
            System.out.println(arr[i]);
        }
    }
}
```

* The main purpose of For-each loop is to **retrieve/fetch the values from the collection like array one by one.**
//Program :-

```
package com.ravi.loop;

public class ForEachDemo1
{
    public static void main(String[] args)
    {
        int []arr = {23, 78, 45, 32};

        //Ordinary for loop
        System.out.println("By using ordinary for loop :");
        for(int i=0; i<arr.length; i++)
        {
            System.out.println(arr[i]);
        }

        System.out.println("By using for-Each loop :");

        for(int x : arr)
        {
            System.out.println(x);
        }
    }
}
```

How to sort array data :
* There is a predefined class called **Arrays available in java.util package.**
* Arrays class has provided predefined static method to sort the array data.

a) public static void sort(int []arr) : Used to sort an integer array.
b) public static void sort(Object []arr) : Used to sort Object type of homogeneous data.

```
package com.ravi.loop;

import java.util.Arrays;

public class ForEachDemo2
{
    public static void main(String[] args)
    {
        int []values = {78, 34, 12, 90, 10};
        System.out.println("Original Array Data :");
        for(int value : values)
        {
            System.out.println(value);
        }
        System.out.println("Sorting Array Data in Ascending Order :");
        Arrays.sort(values);
        for(int value : values)
        {
            System.out.println(value);
        }
    }
}
```

```
package com.ravi.loop;

import java.util.Arrays;

public class ForEachDemo3
{
    public static void main(String[] args)
    {
        String []cities = {"hyd", "kanra", "mumbai", "ajmer"};
        System.out.println("Original Array Data :");
        for(String city : cities)
        {
            System.out.println(city);
        }
        System.out.println("Sorted based on alphabetical order :");
        Arrays.sort(cities);
        for(String city : cities)
        {
            System.out.println(city);
        }
    }
}
```

Can we hold heterogeneous types of data by using array ?
* Yes, We can hold heterogeneous types of data by using Object[] becuase Object is the super class of all the classes in java.

```
public class ForEachDemo4
{
    public static void main(String[] args)
    {
        Object []arr = {12, 23.98, 'A', "RIT", false};

        System.out.println("Printing the array data :");

        for(Object obj : arr)
        {
            System.out.println(obj);
        }
    }
}
```

Can we sort heterogeneous types of data :
* No, We cannot sort hetrogeneous types of data, If we try to sort then It will generate an exception java.lang.ClassCastException

```
Object []arr = {12, 23.98, 'A', "RIT", false};

System.out.println("Printing the array data :");

Arrays.sort(arr); //java.lang.ClassCastException.
```

What is BLC and ELC Class ?

BLC :
* It stands for "Business Logic Class".
* It is used to write the business logic so we **should not write** any main method in the BLC class.
* All the project logic we should always write in the BLC class.

Example :
//BLC class
public class Calculate
{
    //Does not contain main method
}

ELC :
* It stands for "Executable Logic Class".
* It should contain **main method so**, the execution of the program will always starts from ELC class

Example :
//ELC class
public class Main
{
    public static void main(String[] args)
    {
    }
}

//Program on BLC and ELC :

Calculate.java [com.ravi.blc package]
```
package com.ravi.blc;

//BLC
public class Calculate
{
    public static void bonus(int x, int y)
    {
        System.out.println("Sum is :"+(x+y));
    }
}
```

ELC.java [Available in another package com.ravi.elc]
```
package com.ravi.elc;
import com.ravi.blc.Calculate;

//ELC
public class ELC
{
    public static void main(String[] args)
    {
        Calculate.bonus(1000, 2000);
    }
}
```

NOTE : IN THIS BLC CLASS WE ARE importing Calculate class (BLC class) for reusability purpose

How to reuse a class in java ?
The slogan of java is "WORA" write once run anywhere.

A public class created in one package can be reuse from different packages also by using import statement. [non public classes will be usable from the same package only]

In a single java file, we can declare only one public class that must be our .java file and that class can be reusable to all the packages.

*In a single java file, we can write only one public class and multiple non-public classes but it is not a recommended approach because the non public classes we can use within the same package only.

So the conclusion is, we should declare every java class in a separate file to enhance the reusability of the BLC classes.
[Total we have 10 classes -> 10 Java files]

Rectangle.java [Available in com.ravi.blc]
```
package com.ravi.blc;

//BLC
public class Rectangle
{
    public static void getAreaOfRect(int length, int breadth)
    {
        System.out.println("Area of Rectangle is :"+(length * breadth));
    }
}
```

```
package com.ravi.elc;

import com.ravi.blc.Rectangle;

public class Main
{
    public static void main(String[] args)
    {
        Rectangle.getAreaOfRect(10, 20);
    }
}
```

How many .class file will be created for the following program
----
class Alpha
{
}

class Beta
{
}

class Gamma
{
}

public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}

Note : Here compiler will generate 4 class files, But Alpha, Beta and Gamma.class will be reusable within the same package only.