Arrays

Declaring Arrays

- The **array, which stores a fixed-size sequential collection of** elements of the same type.

- **Declaring Array Variables:**

dataType[] arrayRefVar;

or

dataType arrayRefVar[];

Creating arrays

arrayRefVar = new dataType[arraySize];

- It creates an array using new dataType[arraySize];
- It assigns the reference of the newly created array to the variable arrayRefVar.
- Alternatively you can create arrays as follows:

dataType[] arrayRefVar = {value0, value1, …, valuek};

Processing Array

- Done using:
  - For loop
  - For each loop
- Sample code:

FOR EACH LOOP

```
public class TestArray
{
    public static void main(String[] args)
     {
    double[] myList = {1.9, 2.9, 3.4, 3.5};
    // Print all the array elements
    for (double element: myList)
     {
    System.out.println(element);
     }
     }
}
```

FOR LOOP

```
public class TestArray
{

  public static void main(String[] args)
   {
    double[] myList = {1.9, 2.9, 3.4, 3.5};

    // Print all the array elements
    for (int i = 0; i < myList.length; i++)
    {
      System.out.println(myList[i] + " ");
    }
    // Summing all elements
    double total = 0;
    for (int i = 0; i < myList.length; i++)
    {
      total += myList[i];
    }
    System.out.println("Total is " + total);
}
```

Processing Array

Passing Array

```
public static void printArray(int[] array)
{
    for (int i = 0; i < array.length; i++)
    {
    System.out.print(array[i] + " ");
    }
}
```

Returning an Array from a Method:

```
public static int[] reverse(int[] list)
{
  int[] result = new int[list.length];
 for (int i = 0, j = result.length - 1; i < list.length; i++, j--)
 {
     result[j] = list[i];
 }
   return result;
}
```

Array Class

- Present in java.util package
- Methods defined
  - public static int binarySearch(Object[] a, Object key)
  - public static boolean equals(long[] a, long[] a2)
  - public static void fill(int[] a, int val)
  - public static void sort(Object[] a)