Hands-On Activity 1

1. Write a static Java method that returns the **largest** element in a given integer array. Make sure that the method is efficient.

```
public class main {

	public static void main(String[] args) {

		int[] array = { 1, 9, 4, 8, 6, 7 };

		System.out.println(largestElement(array));


	}

	public static int largestElement(int[] arr) {

		int a = arr[0];

		for (int i = 0; i < arr.length; i++)

			if (a < arr[i]) {

				a = arr[i];
			}

		}

		return a;

	}

}
```

2. Write a static Java method that returns the **second** largest element in a given integer array. Make sure that the method is efficient.

```
public class main {

        public static void main(String[] args) {

                int[] array = { 1, 6, 12, 3, 4, 9 };

                System.out.println(secondLargestElement(array));

        }

        public static int secondLargestElement(int[] arr) {

                int largest = arr[0];
                int exLargest = 0;
                int secondLargest = 0;

        for (int i = 0; i < arr.length - 1; i++) {

        if (arr[i] < arr[i + 1]) {

                        if (arr[i + 1] > largest) {

                                        exLargest = largest;
                                          largest = arr[i + 1];

                                        if (arr[i] > secondLargest)
                                                secondLargest = arr[i];
                                        else if (exLargest > secondLargest)
                                                secondLargest = exLargest;

                                }

                 else if (arr[i + 1]>secondLargest && arr[i + 1]<largest)secondLargest =
                                                arr[i + 1];
                        }
         else {

                                if (arr[i + 1] > secondLargest)
                                   secondLargest = arr[i + 1];
                        }

                }

                return secondLargest;

        }
}
```

3. Write a static Java method that multiplies two given matrices (which have elements of type double) and returns the result as a 2D array.

```
public class main {

        public static void main(String[] args) {


                double[][] matrix1 ={
                                { 3.2,   5.7,    6.1   },
                                { 2.1,   1.9,    7.8   },
                                { 9.3,   8.6,    3.2   }
                        };

                double[][] matrix2 = {
                                { 1.8,   2.3   },
                                { 7.7,   6.1   },
                                { 4.9,   3.4   }
                        };

            multiply(matrix1, matrix2);


        }

       public static void multiply(double[][] matrix1, double[][] matrix2) {

                if (matrix1[0].length == matrix2.length) {

                        // 3 3 * 3 2
                        // 3 2 = matrix3

             double[][] matrix3 = new double[matrix1.length][matrix2[0].length];for (int i

                = 0; i < matrix1.length; i++) {

                        for (int k = 0; k < matrix2[0].length; k++) {

                                for (int j = 0; j < matrix1.length; j++) {

                    matrix3[i][k] += matrix1[i][j] * matrix2[j][k];

                                        }
                                }
                        }

                        for (int i = 0; i < matrix3.length; i++) {
                                for (int k = 0; k < matrix3[0].length; k++) {
                                        System.out.print(matrix3[i][k] + " ");
                                }
                                System.out.println();
                        }

                } else {
                        System.out.println("Cannot be multiplied");
                  }

        }
}
```

4. Write a static Java method to reverse the elements of a given integer array, using a temporary array within the method.

```
public class main {

        public static void main(String[] args) {

                int[] array1 = { 1, 2, 3, 4, 5 };

                reverseArray(array1);

        }

        public static void reverseArray(int[] arr) {int[]

                array2 = new int[arr.length]; int k = 0;

                for (int i = arr.length - 1; i >= 0; i--) {

                        array2[k] = arr[i];
                        k++;

                }

                System.out.println(Arrays.toString(array2));

        }

}
```

5. Write a static Java method to reverse the elements of a given array, without using the temporary array.

```
public class main {

        public static void main(String[] args) {

                int[] array1 = { 1, 2, 3, 4, 5 };

                reverseArray(array1);

        }

        public static void reverseArray(int[] arr) {

                int k = arr.length - 1;

                for (int i = 0; i < arr.length / 2; i++) {

                        int temp = 0;
                        temp = arr[i];
                        arr[i] = arr[k];
                        arr[k] = temp;
                        k--;

                }

                System.out.println(Arrays.toString(arr));

        }

}
```

6. Write a static method to randomly shuffle the elements in an array of double values.

```
public class main {

	public static void main(String[] args) {


			double[] array1 = { 1.2, 7.5, 9.2, 6.3, 3.4 };


	System.out.println(Arrays.toString(shuffledArray(array1)));


	}

	public static double[] shuffledArray(double[] arr) {

		double[] array2 = arr;

		for (int i = 0; i < arr.length; i++) {

			int index = (int) (Math.random() * array2.length);

			double temp = array2[i];
			array2[i] = array2[index];
			array2[index] = temp;

		}

		return array2;

	}

}
```

7. Write a static method that checks whether the given array of double values is sorted in ascending order.

```
public class main {

	public static void main(String[] args) {

		double[] array1 = { 1.2, 7.5, 9.2, 3.8, 6.5 };

		System.out.println(isSorted(array1));

	}

	public static boolean isSorted(double[] arr) {

		for (int i = 0; i < arr.length - 1; i++) {

			if (arr[i] > arr[i + 1]) {
				return false;
			}

		}

		return true;

	}

}
```