Assignment-10

Part -A

Que-1)Difference between deep copy and shallow copy.

Shallow Copy:

Shallow copy stores references of objects to the original memory address. Any changes made to the copied object are reflected in the original object. It copies the original object but points to the same referenced objects. Shallow copy is faster and uses less memory.

Deep Copy:

Deep copy stores separate copies of the object’s values in new memory locations. Changes made to the copied object do not affect the original object. It copies the original object and recursively copies all referenced objects. Deep copy is slower and uses more memory.

Que-2)What are varargs in Java?

Varargs in Java allow a method to accept variable number of arguments.

method with varargs-

```
public void main(int... a){
}
```

internally, it use array to hold any no. of parametes.

Only one varargs parameter is allowed in a method.

The varargs parameter must be the last parameter in the method signature.

Que-3)What is autoboxing and unboxing?

AutoBoxing - When a primitive data type is automatically converted to their corresponding reference types, then it is called autoboxing.   For example, when a int value is automatically converted into the Integer object, then it is called autoboxing.

Unboxing - When a wrapper class object is automatically converted into the primitive type, then it is called unboxing.
         For example, when an Integer object is converted into the int, it is called unboxing.

Autoboxing and unboxing handle by compiler, but while unboxing, it throw nullpointerException if object point to null.

Que-4)What are the differences between == and equals() for Strings?

== is oparator that used to compare the references of objects. If it returns true, both references point to the same object in memory. If it returns false, the references point to different objects.

The equals() method is used to compare the content of objects. If the equals() method returns true, it means both objects have the same content. If it returns false, it means the contents of the objects are different.

Que-5)Explain String concatenation and interning.

String concatenation means joining two or more Strings using + or concat() method.

Strings are immutable, concatenation always creates a new String object.

String interning is a memory-saving mechanism where Java stores String literals in the String Constant Pool and reuses existing Strings instead of creating new ones.

Part - B

Que-1)Write a program to find second largest element in an array.

```
import java.util.Scanner;

public class SecondMax {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter N: ");
        int n = sc.nextInt();

        int[] arr = new int[n];
        int max = Integer.MIN_VALUE;
        int secMax = Integer.MIN_VALUE;

        System.out.println("Enter Array Elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        for (int i = 0; i < n; i++) {
            if (arr[i] > max) {
                secMax = max;
                max = arr[i];
            }
            else if (arr[i] > secMax && arr[i] != max) {
                secMax = arr[i];
            }
        }

        if (secMax == Integer.MIN_VALUE) {
            System.out.println("Second maximum does not exist");
        } else {
            System.out.println("Second Maximum Number: " + secMax);
        }
    }
}
```

```
Output -
Enter N: 4
Enter Array Elements:
12
32
32
43
Second Maximum Number: 32
```

Que-2)Write a program to sort an array in ascending order.

```
import java.util.Scanner;

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

            Scanner sc = new Scanner(System.in);

            System.out.print("Enter a N : ");
            int n = sc.nextInt();

            int[] arr = new int[n];

            System.out.println("Enter array elements:");
            for (int i = 0; i < n; i++) {
                  arr[i] = sc.nextInt();
            }

            for (int i = 0; i < n - 1; i++) {
                  for (int j = i + 1; j < n; j++) {
                        if (arr[i] > arr[j]) {
                              int temp = arr[i];
                              arr[i] = arr[j];
                              arr[j] = temp;
                        }
                  }
            }

            System.out.println("Array in Ascending Order:");
            for (int i = 0; i < n; i++) {
                  System.out.print(arr[i] + " ");
            }
      }
}
```

```
Output -
Enter a N : 4
Enter array elements:
12
31
2
45
Array in Ascending Order:
2 12 31 45
```

Que-3)Write a program to sort an array in descending order.

```
import java.util.Scanner;

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

            Scanner sc = new Scanner(System.in);

            System.out.print("Enter a N : ");
            int n = sc.nextInt();

            int[] arr = new int[n];

            System.out.println("Enter array elements:");
            for (int i = 0; i < n; i++) {
                  arr[i] = sc.nextInt();
            }

            for (int i = 0; i < n - 1; i++) {
                  for (int j = i + 1; j < n; j++) {
                        if (arr[i] < arr[j]) {
                              int temp = arr[i];
                              arr[i] = arr[j];
                              arr[j] = temp;
                        }
                  }
            }

            System.out.println("Array in Descending Order:");
            for (int i = 0; i < n; i++) {
                  System.out.print(arr[i] + " ");
            }
      }
}
```

```
Output -
Enter a N : 4
Enter array elements:
123
532
123
3455
Array in Descending Order:
3455 532 123 123
```

Que-4)Write a program to rotate an array left by n positions.

```
import java.util.Scanner;

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

            Scanner sc = new Scanner(System.in);

            System.out.print("Enter a N : ");
            int n = sc.nextInt();

            int[] arr = new int[n];

            System.out.println("Enter array elements:");
            for (int i = 0; i < n; i++) {
                  arr[i] = sc.nextInt();
            }

            System.out.println("Enter A Position : ");
            int key = sc.nextInt();

            key = key%n;

            for (int i = 0; i < key; i++) {
                  int first = arr[0],j;
                  for (j = 0; j < arr.length-1; j++) {
                        arr[j] = arr[j+1];
                  }
                  arr[j] = first;
            }

            System.out.println("After Rotating array by " + key + " : ");
            for (int i : arr) {
                  System.out.print(i + " ");
            }
      }
}
```

Output -

```
Enter a N : 4
Enter array elements:
12
31
4321
12
Enter A Position :
2
After Rotating array by 2 :
4321 12 12 31
```

Que-5)Write a program to rotate an array right by n positions.

```
import java.util.Scanner;

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

            Scanner sc = new Scanner(System.in);

            System.out.print("Enter a N : ");
            int n = sc.nextInt();

            int[] arr = new int[n];

            System.out.println("Enter array elements:");
            for (int i = 0; i < n; i++) {
                  arr[i] = sc.nextInt();
            }

            System.out.println("Enter A Position : ");
            int key = sc.nextInt();
            key = key%n;

            for (int i = 0; i < key; i++) {
                  int last = arr[n - 1];
                  for (int j = n - 1; j > 0; j--)
                        arr[j] = arr[j - 1];
                  arr[0] = last;
            }

            System.out.println("After Rotating array by " + key + " : ");
            for (int i : arr) {
                  System.out.print(i + " ");
            }
      }
}
```

Output -

```
Enter a N : 4
Enter array elements:
23
2
33
2
Enter A Position :
2
After Rotating array by 2 :
33 2 23 2
```

Github Link - https://github.com/YogeshPathade01/Java_Coding_Questtions.git