Assignment: 5

Page No.:
Date:

1] Explain Arraycopy() with example.

=> This static method is available in this class system of the pacakage Java.lang.
- This method is used to copy an array from another.
- The speciality of this method is that you can select the starting location of source and destination array and also the numbes of elements can be selected.
- Syntax:

System.arraycopy(source_of_copy, starting, destination, ending_index, length)

For, e.g.

System.arraycopy(a, 0, b, 1, 3);

- It will copy a total n-1 elements whese n is the sizes of array 'a'.
- The elements will be copied from begining of element numbes '0' of array 'a', into the array 'b' begining from Indes '1' in the array is 'b'.

2] Explain methods of string class.

=> (1) .length()

-> Return an int value equal to length of the string.

(2) .toLowerCase()

-> It returns the string object through which it is called with the string converted into lower case.

(3) .toUpperCase()

-> It returns the string object through which it is called with the converted into upper case.

(4) .trim()

-> This method trims the blank space at the end of a string

(5) .equals(string)

-> It returns 0 for false and 1 for True after the two string's are compared.

(6) .replace(char, char)

-> It replaces the character in the string with the given character in the 2nd position.

(7) .equalsIgnoreCase(string)

-> It compares the string ignoring whether string is in upper or lower case and returns 0 or 1 accordingly

(8) .compareTo(string)

-> It compare's two string's and returns 0, 1, or -1 accordingle. If both are same then it will return 0.

(9) substring(int)

-> It returns the string from the given index.
e.g. for Youtube
substring(3) => tube

(10) substring(int, int)

-> It will give output from starting to ending index.
e.g. for Youtube
substring(1, 4) => outu

(11) .concate(string)

-> It combines two string with the given string in the brackets.

(12) charAt(int)
-> It return a particular characters from the given index
e.g. You
charAt(2) => u

(13) index of (char)
-> It return a particular index of given character.

(14) Index of (char, int)
-> It return the index of a character from a particular index.

15) toCharArray():
-> This method converts the string into a character array.

3] Explain methods of string Buffer Class

=> (1) setcharAt (int, char)
-> It sets a character at a particular given index

(2) charAt (int)
-> It returns the character at a given index

(3) to string()
-> It converts the string buffer data into a string.

(4) Insert (int, char)
-> Inserts the character passed at the index passed.

(5) Delete (int, int)
-> Deletes the substring from the starting to ending indices passed to the method.

(6) Replace (int, int, string)
-> Replace the string passed into the original string from the starting to ending indices passed to the method.

(7) append (string)
-> Appends the string passed to the original string.

4] Explain Dynamic method Dispath with example.

=> In case of method overriding when a method is called by an object of the class the method of the corresponding class is called
-> A base class refrence object can be used to refer either a base class object or a derived class object.
-> In this case of overridden method to be called depends on the object

to which the reference object is referring that moment.

-> This can be realized only during the runtime that the reference object is pointing to which class object.

Example:

```
class Parent {
    public static void main1(int x) {
    System.out.println ("x = " +x);
    }
}
class Child extends Parent {
    public static display (int y) {
        System.out.println("y = " +y);
    }
}
class main {
    public static void main (String args[]){
        child c = new child();
        parent p = new parent();
        parent ref;
        ref=c;
        ref display(5)
        ref = p;
        ref .display(10);
    }
}
```

OUTPUT:

y=5
x=10

5) Explain Finalize method and Wrappers Class.

=> * Finalize Method,

- In Java, it is a call back method which is called before any object is garbage collected.
- It is present in the java object class so it is available to all the classes in java.
- Though the finilize method of class object performs no special action; It simply returns normally.
- You will have to override the required implementation for cleaning up.
- It's object class is as protected void finalize() throws Throwable { }

* Wrappers Class

- A wrapper class is a class whose object wraps or contains a primitive data types.
- when we create an object to a wrappers class, it contains a field and in this field, we can store a primitive data types.
- In other words, we can wrap a primitive value into a wrapper class object.

Page No.
Date

:- Need

- An object is needed to support synchronization in multithreading.
- They convert primitive data types into objects. Objects are needed if we wish to modify the arguments passed into a method.
- The classes in java.util package handles only objects and hence wrapper classes.

6] WAP to accept n integers from user into array and display average of three numbers.

=>
```
import java.util.*;
class Average
{
  public static void main(String args[])
  {
    int i, n, sum=0
    float avg;
    Scanner sc = new Scanner(System.in)
    System.out.println("Enter no. of ele.");
    n = sc.nextInt();
    int a[] = new Int[n];
    for(i=0; i<=n-1; i++)
    {
     System.out.println("Enter a no.");
     a[i] = sc.nextInt();
    }
```

Page No.
Date

```
    for(i=0; i<=n-1; i++)
    {
      Sum(i=0; i<=n-1; i++)
      sum = sum + a[i];
    }
    avg = (float) sum;
    System.out.println("Avg =" + avg);
  }
}
```

OUTPUT:

```
Enter no. of ele.
4
Enter a no.
2
Enter a no.
3
Enter a no.
1
Enter a no.
4
Avg = 2.5
```

7] WAP to accept m×n matrix and display it in matrix form.

=>
```
import java.util.Scanner;
class Matrix
{
  public static void main(String args[])
  {
    int[][] a = new int[3][3];
    int i, j;
```

```
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter values for matrix");
        for (i=0; i<3; i++)
        {
            for (j=0; j<3; j++)
            {
                a[i][j] = sc.nextInt();
            }
        }
        System.out.println("Given matrix is:");
        for (i=0; i<3; i++)
        {
            for (j=0; j<3; j++)
            {
                System.out.print(a[i][j] + " ");
            }
            System.out.println(" ");
        }
    }
}
```

OUTPUT:

```
Enter values for matrix
1
2
3
4
5
6
7
8
9
Given matrix is:
1 2 3
4 5 6
7 8 9
```