**7. Study and report.**

**a. java Arrays class their methods**

**Arrays class in Java**

The Arrays class in java util package is a part of the Java Collection Framework. This class provides static methods to dynamically create and access Java arrays. It consists of only static methods and the methods of Object class. The methods of this class can be used by the class name itself.

**Methods in Java Arrays with examples**

**1**. static <T> List<T> asList (T… a): asList method is used to return the fixed-size list that mentioned Arrays back.

**Code:**

import java.util.Arrays;

public class Array {

public static void main(String[] args)

{

int Arr[] = { 10, 30, 35, 52, 75 };

System.out.println("The Integer Array as a List = "+ Arrays.asList(Arr));

}

}

**Output:**

integer array

**2.** static int binary Search (item To Search): This method would search for a mentioned element in the array through the Binary Search algorithm.

**Code:**

import java.util.Arrays;

public class Main {

public static void main(String[] args)

{

int Arr[] = { 10, 30, 35, 52, 75 };

Arrays.sort(Arr);

int ele = 35;

System.out.println (ele + " is found at index = "+ Arrays .binary Search (Arr, ele));

} }

**Output:**

index

**3.** static <T> int binary Search (T [] an int from Index, int to Index, T key, Comparator<T> c): This method would search the range of mentioned array for a specified object making use of binary search algorithm.

**Code:**

import java.util.Arrays;

public class Main {

public static void main (String[] args)

{

int Arr[] = { 10, 30, 35, 52, 75 };

Arrays.sort(Arr);

int ele = 35;

System.out.println ( ele+ " is found at index = "+ Arrays. Binary Search(Arr, 1, 3, ele));

}

}

**Output:**

Array system

**4**. compare Using ned(arr 1, arr 2): compare Using ned method would compare two arrays that are passed as parameters in a lexicographical style and treating them as unsigned. This method of Integer class would compare two integer values treating them as unsigned and then returning zero in case x is equal to y.

**Code:**

import java.lang.Integer;

class Arrays {

public static void main(String args[])

{

int m = 10;

int n = 20;

System. out . print ln (Integer. compare Using ned(m, n));

int x = 8;

int y = 8;

System. out. Print ln(Integer .compare Using ned(x, y));

int e = 25;

int f = 8;

System .out .print ln (Integer. Compare Using ned(e, f));

int o = 15;

int p = -7;

System .out .print ln (Integer. Compare Using ned(o, p));

}

}

**Output:**

system output

**5.** copy Of (original array, new length): copy method copies the mentioned array, truncates it or pads it with a default value but only if necessary so that copy has got the mentioned length.

**Code:**

import java.util.Arrays;

public class Example {

public static void main(String[] args)

{

int Arr[] = { 10, 25, 55, 22, 35};

System .out. print ln("The Integer Array is: "+ Arrays. To String(Arr));

System. out .print ln("\n The new Arrays fetched by copyOf is :\n");

System. out .print ln("Integer Array is: "

+ Arrays.to String(Arrays .copy Of(Arr, 10)));

}

}

**Output:**

Array Methods in Java

**6**. copy Of Range(the previous array, start Index, finish Index): copy Of Range method would copy the mentioned range of the mentioned array into a new Array.

**Code:**

import java.util.Arrays;

public class Array{

public static void main(String[] args)

{

int Arr[] = {20, 30, 15, 22, 35 };

System.out.println("Integer Array is: "+ Arrays. to String (Arr));

System.out.println("\nThe new Arrays through copy Of Range is :\n");

System.out.println("Integer Array: "+ Arrays.to String(Arrays. copy Of Range(Arr, 1, 3)));

}

}

**Output:**

Array Methods in Java

**7**. static boolean deep Equals(Object[] m1, Object[] m2): deep Equals method would return true in case the two mentioned arrays are deeply equal to the other array or not.

**Code:**

import java.util.Arrays;

public class Array{

public static void main(String[] args)

{

int Arr[][] = { {10, 20, 35, 82, 95} };

int Arr2[][] = { { 10, 15, 22 } };

System.out.println("Arrays when compared: "+ Arrays.deep Equals(Arr, Arr2));

}

}

**Output:**

Array Methods in Java

**8.** static int deep Hash Code(Object[] a): deep Hash Code: method would return the hash code depending upon “deep contents” of the mentioned arrays.

**Code:**

import java.util.Arrays;

public class Array {

public static void main(String[] args)

{

int Arr[][] = { { 10, 20, 15, 22, 35} };

System. Out .print ln(Arrays .deep Hash Code(Arr));

}

}

**Output:**

Array Methods in Java

**b.java String class their method**

The java. lang. String class provides a lot of built-in methods that are used to manipulate string in Java. By the help of these methods, we can perform operations on String objects such as trimming, concatenating, converting, comparing, replacing strings etc.

Java String is a powerful concept because everything is treated as a String if you submit any form in window based, web based or mobile application.

1. **Java String to Upper Case() and to Lower Case() method**

The Java String toUpperCase() method converts this String into uppercase letter and String toLowerCase() method into lowercase letter.

**Code:**

public class Stringoperation1

{

public static void main(String ar[])

{

String s="Sachin";

System. Out .print ln(s.to Upper Case());

System. out. Print ln(s.to Lower Case());

System. out. println(s);//Sachin(no change in original)

}

}

**Output:**

SACHIN

sachin

Sachin

1. **Java String trim() method**

The String class trim() method eliminates white spaces before and after the String.

**Code:**

public class Stringoperation2

{

public static void main(String ar[])

{

String s=" Sachin ";

System.out.println(s);

System.out.println(s.trim());

}

}

**Output:**

Sachin

Sachin

1. **Java String starts With() and ends With() method**

The method starts With() checks whether the String starts with the letters passed as arguments and ends With() method checks whether the String ends with the letters passed as arguments.

**Code:**

public class Stringoperation3

{

public static void main(String ar[])

{

String s="Sachin";

System.out.println(s. starts With("Sa"));

System.out.println(s. ends With("n"));

}

}

**Output:**

true

true

1. **Java String charAt() Method**

The String class char At() method returns a character at specified index.

**Code:**

public class Stringoperation4

{

public static void main(String ar[])

{

String s="Sachin";

System.out.println(s.char At(0));

System.out.println(s.char At(3));

}

}

**Output:**

S

h

1. **Java String length() Method**

The String class length() method returns length of the specified String.

**Code:**

public class Stringoperation5

{

public static void main(String ar[])

{

String s="Sachin";

System.out.println(s. length ());//6

}

}

**Output:**

6

1. **Java String intern() Method**

A pool of strings, initially empty, is maintained privately by the class String.

When the intern method is invoked, if the pool already contains a String equal to this String object as determined by the equals(Object) method, then the String from the pool is returned. Otherwise, this String object is added to the pool and a reference to this String object is returned.

**Code:**

public class Stringoperation6

{

public static void main(String ar[])

{

String s=new String("Sachin");

String s2=s. intern ();

System.out.println(s2);

}

}

**Output:**

Sachin

1. **Java String value Of() Method**

The String class value Of() method coverts given type such as int, long, float, double, boolean, char and char array into String.

**Code:**

public class Stringoperation7

{

public static void main(String ar[])

{

int a=10;

String s=String. value Of(a);

System.out.println(s+10);

}

}

**Output:**

1010

1. **Java String replace() Method**

The String class replace() method replaces all occurrence of first sequence of character with second sequence of character.

**Code:**

public class Stringoperation8

{

public static void main(String ar[])

{

String s1="Java is a programming language. Java is a platform. Java is an Island.";

String replace String=s1.replace("Java", "Kava");//replaces all occurrences of "Java" to "Kava"

System.out.println(replace String);

}

}

**Output:**

Kava is a programming language. Kava is a platform. Kava is an Island.