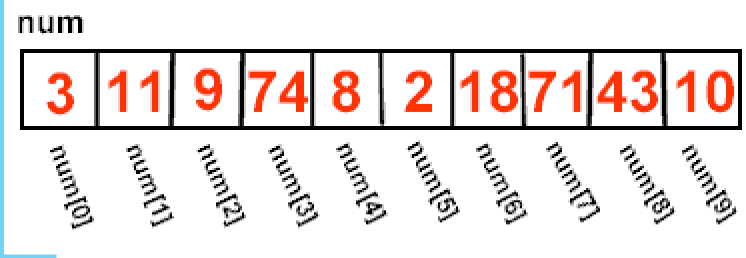
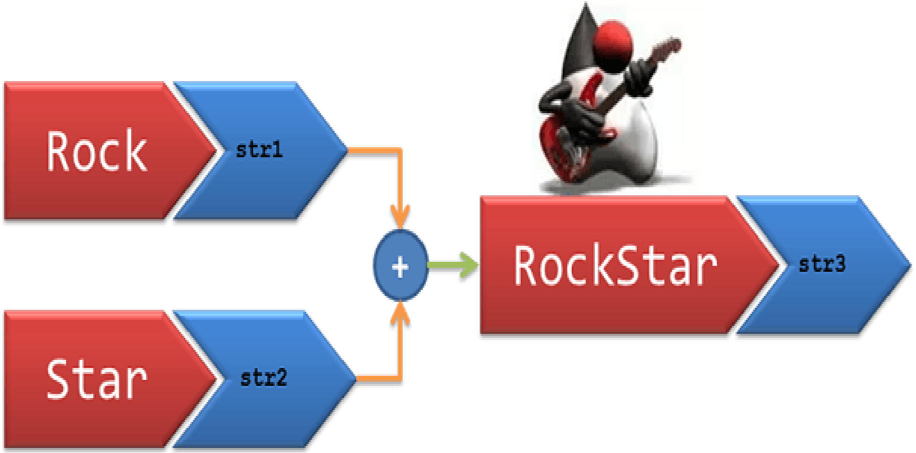
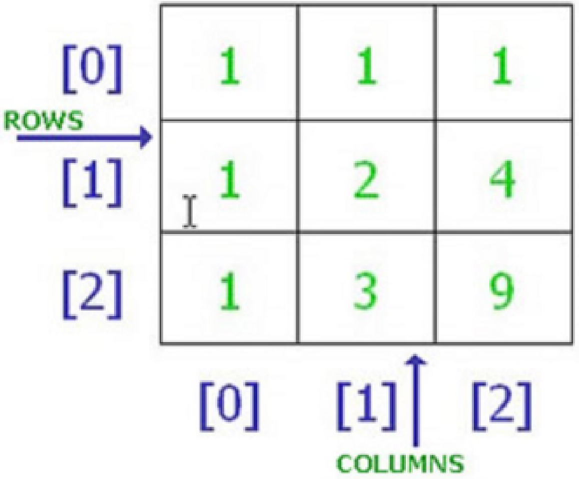


num

num[0]	num[1]	num[2]	num[3]	num[4]	num[5]	num[6]	num[7]	num[8]	num[9]
3	11	9	74	8	2	18	71	43	10

Arrays, Strings, Vectors & Wrapper Classes

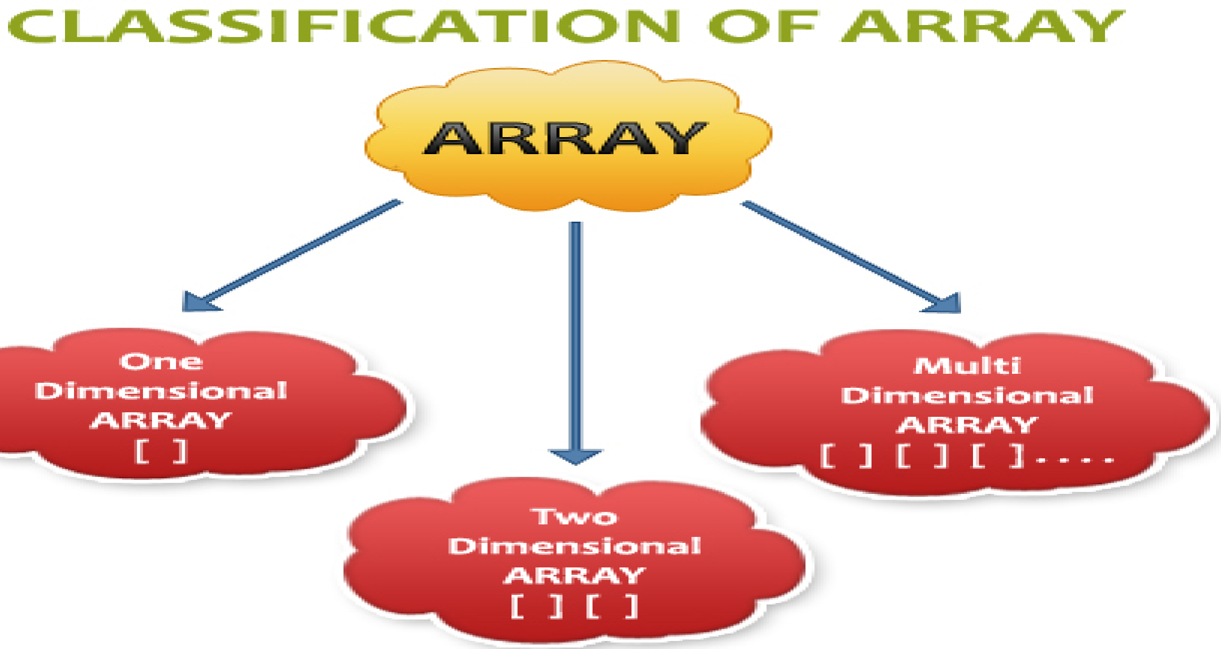




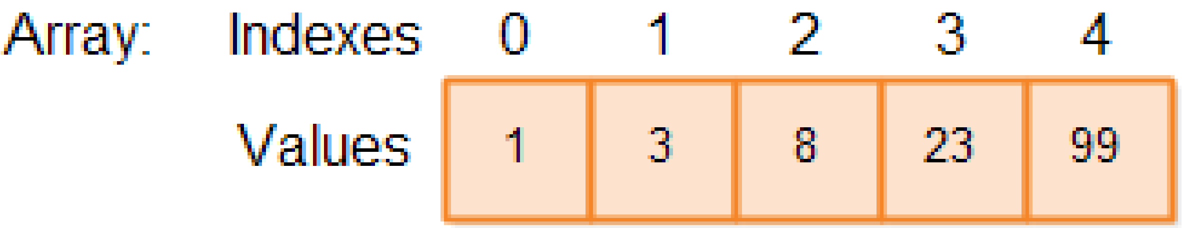
	[0]	[1]	[2]
[0]	1	1	1
[1]	1	2	4
[2]	1	3	9

Introduction

- Array as a collection of variables of the same type.
- It is fixed in size means that you can't increase the size of array at run time.
- It is a collection of homogeneous data elements.
- It stores the value on the basis of the index value.


CLASSIFICATION OF ARRAY
ARRAY
One Dimensional ARRAY [ ]
Two Dimensional ARRAY [ ] [ ]
Multi Dimensional ARRAY [ ] [ ] [ ]....

One Dimensional Array



Array:

Indexes	0	1	2	3	4
Values	1	3	8	23	99

Declaring Array Variables

Syntax:

```
dataType[] arrayRefVar;
        or
dataType arrayRefVar[];
```

Example :

```
    int[] number;
        or
    int number[];
```

Creating Arrays

You can create an array by using the new operator with the following syntax:

```
arrayRefVar = new dataType[arraySize];
```

It assigns the reference of the newly created array to the variable arrayRefVar.

e.g.

```
number=new int[5];
```

Initialization of Array

Syntax:

```
arrayname[subscript]=value;
```

e.g.

```
number[0]=35;
```

Alternatively you can initialize arrays as follows:

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

e.g.

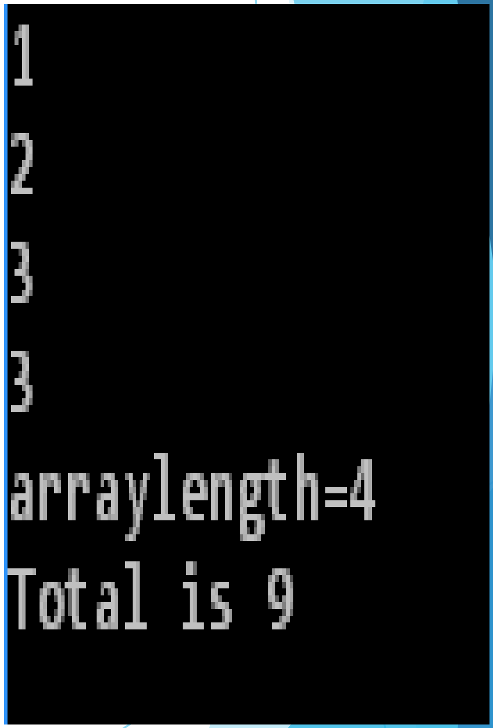
```
int number[]={2,3,4,5,6,7};
```

Example: To print of array elements

```
class TestArray
{
   public static void main(String[] args)
   {
      int myList[] = {1, 2, 3, 3};
      for (int i = 0; i < myList.length; i++)
      {
           System.out.println(myList[i] + " ");
     }
      System.out.println("arraylength="+myList.
length);
```

length: It is a final variable and only applicable for array. It represent size of array.

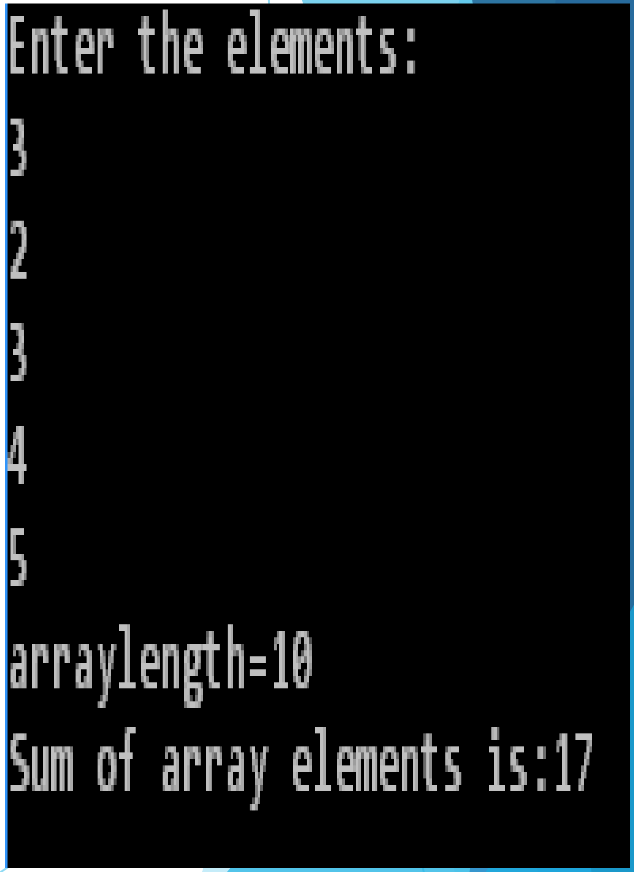
```
// Summing all elements
     int total = 0;
     for (int i = 0; i < myList.length; i++)
    {
          total += myList[i];
       }
     System.out.println("Total is " + total);
   }
}
```



```
1
2
3
3
arraylength=4
Total is 9
```

Example: To find sum of array elements accepted by the user

```
import java.util.Scanner;
class ArraySumDemo  {
   public static void main(String args[])
  {
      Scanner sc= new Scanner(System.in);
      int a[] = new int[10];
      int sum = 0;
      System.out.println("Enter the elements:");
      for (int i=0; i<10; i++)
      {
           a[i] = sc.nextInt();
      }
System.out.println("arraylength="+array.length);
    for( int j=0;j<10;j++)
    {
         sum = sum+a[j];
    }
    System.out.println("Sum of array
elements is:"+sum);
  }
}
```



```
Enter the elements:
3
2
3
4
5
arraylength=10
Sum of array elements is:17
```

Question

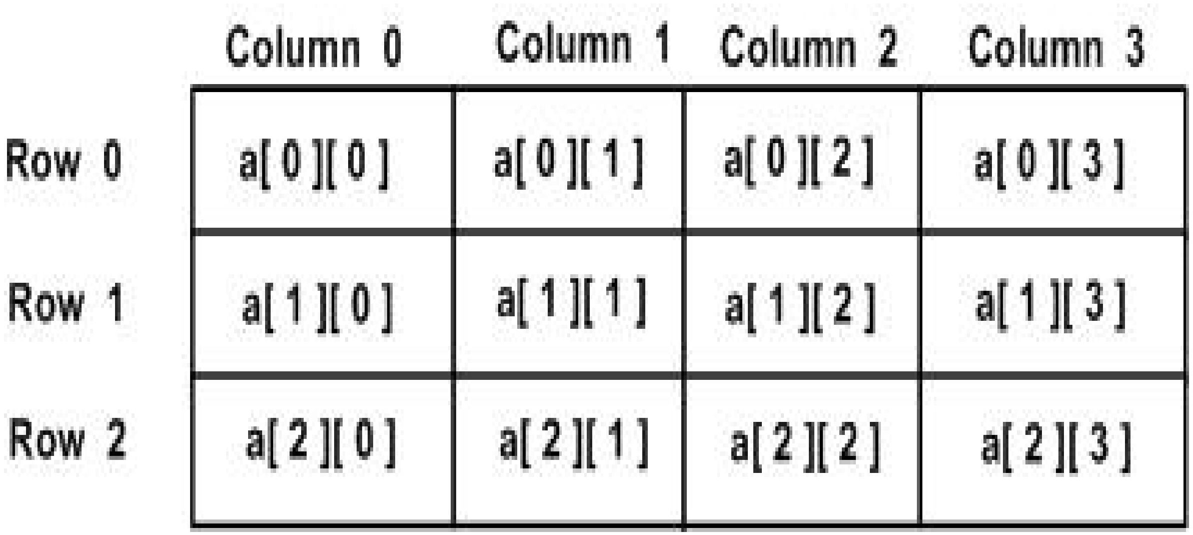
WAP in java to find Largest element of the array.

Solution

```
class TestArray
{
    public static void main(String[] args)
    {
        Scanner sc= new Scanner(System.in);
        int a[] = new int[10];
        int sum = 0;
        System.out.println("Enter the elements:");
        for (int i=0; i<10; i++)
        {
            a[i] = sc.nextInt();
        }
int max=a[0];
for (int i = 1; i < a.length; i++)
     {
       if (a[i] > max)
       max = a[i];
      }
    System.out.println("Max is " + max);
  }
}
```

Two Dimensional Array

A list of items with one variable name and two subscripts is called a Two dimensional array.



	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Declaration of Two dimensional Array

Syntax :

```
dataType[][] arrayRefVar;
        (or)
dataType arrayRefVar[][];
```

e.g.

```
int[][] arr=new int[3][3];
```

Initialization of Two dimensional Array

e.g.

```
        arr[0][0]=1;

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

      arr[][]={
                    {1,2,3},
                    {2,4,5},
                    {4,4,5}
                };
```

Example

```
class Testarray
{
   public static void main(String args[])
   {
       //declaring and initializing 2D array
        int arr[][]={{1,2,3},{2,4,5},{4,4,5}};
      //printing 2D array
      for(int i=0;i<3;i++)
      {
         for(int j=0;j<3;j++)
         {
              System.out.print(arr[i][j]+" ");
          }
          System.out.println();
       }
     }
}
```

Question

WAP in java to add two matrices. Accept input from user.

EXP2

- **To implement Arrays**
  - You have been given an array of positive integers A1, A2,..., An with length N and you have to print an array of same length (N) where the values in the new array are the sum of every number in the array, except the number at that index.
- i/p 1 2 3 4
- For the 0th index, the result will be 2+3+4= 9, similarly for the second, third and fourth index the corresponding results will be 8, 7 and 6 respectively.
- i/p 4 5 6
- o/p 11 10 9

- The annual examination results of 5 students are tabulated as follows:

Roll No	Subject1	Subject2	Subject3

- WAP to read the data and determine the following
- Total marks obtained by each student
- The student who obtained the highest total marks
  - WAP to display following pattern using irregular arrays (jagged arrays).
- 1
- 1 2
- 1 2 3 ……….

Solution

```
import java.util.Scanner;
class AddTwoMatrix
{
   public static void main(String args[])
   {
      int m, n, i, j;
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the number of rows and 
columns of matrix");
      m = sc.nextInt();
      n  = sc.nextInt();
```

contd…

contd…

```
      int first[][] = new int[m][n];
      int second[][] = new int[m][n];
      int sum[][] = new int[m][n];
     System.out.println("Enter the elements of first
matrix");
       for (  i = 0 ; i < m ; i++ )
         {
         for ( j = 0 ; j < n ; j++ )
           {
            first[i][j] = sc.nextInt();
           }
         }
```

```
System.out.println("Enter the elements of second
matrix");
 {
     for (  i = 0 ; i < m ; i++ )
        {
          for ( j = 0 ; j < n ; j++ )
         {
            second[i][j] = sc.nextInt();
         }
   }
```

contd…

```
for (  i = 0 ; i < m ; i++ )
    {
      for ( j = 0 ; j < n ; j++ )
      {
         sum[i][j] = first[i][j] + second[i][j];
     }
  }
```

contd…

```
System.out.println("Sum of entered matrices:-");
       for ( i = 0 ; i < m ; i++ )
      {
         for ( j = 0 ; j < n ; j++ )
          {
            System.out.print(sum[i][j]+"\t");
           }
          System.out.println();
      }
   }
}
```

System.arraycopy()

- The java.lang.System.arraycopy() method copies an array from the specified source array, beginning at the specified position, to the specified position of the destination array.
- A subsequence of array components are copied from the source array referenced by src to the destination array referenced by dest.
- The number of components copied is equal to the length argument.
- This method does not return any value.

System.arraycopy()

- Syntax :

System.arraycopy(srcArray, srcPos, destArray, destPos, length)

Where,

srcArray is Source Arrayname

srcPos index of source array from which array is to be copied

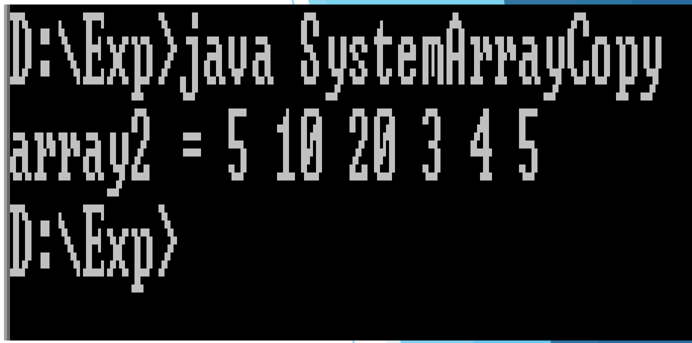
destArray is Destination Arrayname

destPos is index of destination array where the copied array is to put.

Length is the number of elements to be copied

Example

```
public class SystemArrayCopy
{
   public static void main(String[] args)
   {
         int arr1[] = { 0, 1, 2, 3, 4, 5 };
         int arr2[] = { 5, 10, 20, 30, 40, 50 };
        // copies an array from the specified source array
         System.arraycopy(arr1, 3, arr2, 3, 3);
         System.out.print("array2 = ");
         for(int i=0;i<6;i++)
         {            System.out.print(arr2[i] + " ");
         }
   }
}
```



```
D:\Exp>java SystemArrayCopy
array2 = 5 10 20 3 4 5
D:\Exp>
```

Question

WAP a java program to copy one array elements into second using system.arraycopy().

- Create and initialize array A and array B.
- Copy 4 elements from third element of array B into array A
- Place these 4 elements from the index value 3 of the array A.

Solution

```
import java.util.Scanner;
class SysArrayCopyex   {
  public static void main(String args[])
  {
    Scanner scanner = new Scanner(System.in);
    int[] A = new int[10];
    int[] B = new int[10];
    int sum = 0,count=0;
    System.out.println("Enter the  elements of A:");
    for (int i=0; i<10; i++)
    {        A[i] = scanner.nextInt();              }
    System.out.println("Enter the  elements of B:");
    for (int i=0; i<10; i++)
    {           B[i] = scanner.nextInt();            }
     System.arraycopy(B, 2, A, 3, 4);
      System.out.print("new array A = ");
      for(int i=0;i<10;i++)
      {        System.out.print(A[i] + " ");        }
   }  }
```

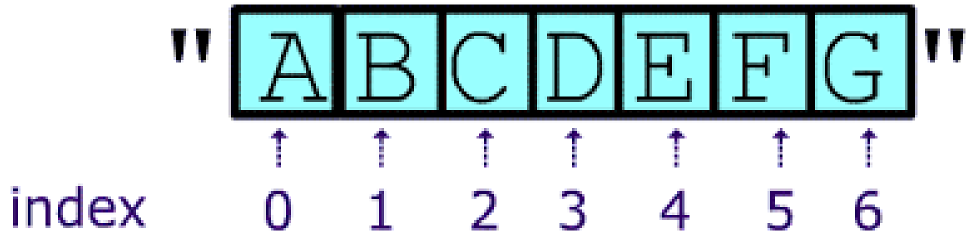
String

Character
Express
R
O
S
E

Introduction

- **String represent a sequence of characters.**
- In JAVA, strings are class objects and implemented using two classes :
  - ❑ String
  - ❑ StringBuffer

- A java string is not a character array and is not NULL terminated.



© javaclass.info

Declaration and Creation of String

Syntax:

```
String stringname=new String(“String”);
                  or
 String s=“stringname”;
```

e.g.

```
String firstName=new String(“Sachin”);
                  or
 String s=“Sachin”;
```

Example

```
class StringExample
{
  public static void main(String args[])
  {
   String s1="java";            //creating string by java string literal
   char ch[]={'s','t','r','i','n','g','s'};
   String s2=new String(ch);       //converting char array to string
  String s3=new String("example");//creating java string by new keyword
   System.out.println(s1);
   System.out.println(s2);
   System.out.println(s3);
  }
}
```

String Arrays

Syntax:

String Arrayname[]=new String[size];

Example:

```
class StringArray
{
  public static void main(String[] args)
  {
      String[] s1 = {“abc", “xyz", “pqr"};
       int size = s1.length;
        for (int i=0; i<size; i++)
        {           System.out.println(s1[i]);       }
   }
}
```

String methods

Method call	Task performed
s2=s1.tolowerCase;	Convert the string to all lowercase
s2=s1.toupperCase;	Convert the string to all uppercase
s2=s1.replace(‘x’,’y’);	Replace all appearances of x with y
s2=s1.trim();	Remove white spaces at the beginning and end of the string s1
s1.equals(s2);	Return true if s1=s2
s1.equalsIgnoreCase(s2)	Return true if s1=s2, ignoring the case of characters
s1.length()	Gives the length of s1
s1.CharAt(n)	Gives n th character of s1
s1.compareTo(s2)	Returns negative if s1<s2,positive if s1>s2 and zero if s1=s2
s1.concat(s2)	Concatenates s1 and s2

Contd…

S1.substring(n)	Gives substring starting from n th character
S1.substring(n,m)	Gives substring starting from n th character up to m th(excluding m th character)
p.toString()	Creates a string representation of object p
S1.indexOf(‘x’)	Gives the position of the first occurrence of ‘x’ in the string s1
S1.indexOf(‘x’,n)	Gives the position of of ‘x’ that occurs after nth position in the string s1
String.ValueOf(variable)	Converts the parameter value to string representation(converts int, float etc into string)

Example

```
class String_method
{
public static void main(String args[])
{
   String s="Sachin";
   System.out.println(s.length());
   System.out.println(s.toUpperCase());
   System.out.println(s.toLowerCase());
   System.out.println(s);      //Sachin (no change in original)
   System.out.println(s.charAt(3));
   String ss=“  Virat Kohali ";
   System.out.println(ss.trim());
   System.out.println(ss.substring(3));
```

contd…

Contd…

```
System.out.println(ss.substring(2,6));
System.out.println(ss.indexOf('a'));
System.out.println(ss.indexOf('a',7));
int a=10;
String s2=String.valueOf(a);
System.out.println(s2+10);
String s1="Java is a programming language. Java is a platform.";
String replaceString=s1.replace(“Java",“Python");//replaces all occurrences of
                                                 "Java"  to " Python "
System.out.println(replaceString);
System.out.println("Hello".concat("World"));
System.out.println("Hello".compareTo("World"));
}
}
```

Ouput

```
6
SACHIN
sachin
Sachin
h
Virat Kohali
irat Kohali
Vira
5
11
1010
Python  is a programming language. Python is a platform.

HelloWorld
-15
```

Question 1

Write a program to find number of uppercase, lowercase characters, blank spaces, digits and special character from a string.

Solution

```
import java.util.*;
class Count
{
    public static void main(String args[])
 {
     Scanner sc=new Scanner(System.in);
   System.out.println("enter the String");
   String s=sc.nextLine();
    int i,digit=0,whitespace=0,sp=0,upperCase=0,lowerCase=0;
    char ch;
    int l=s.length();
   for(i=0;i<l;i++)
   {
     ch=s.charAt(i);
     if (Character.isUpperCase(ch))
    { upperCase++; }
```

contd…

Contd…

```
   else if (Character.isLowerCase(ch)){ lowerCase++; }
   else if(Character.isDigit(ch))
   ++digit;
   else if(ch==' ')
   ++whitespace;
    else
    ++sp;
}
System.out.println("no of Uppercase="+upperCase);
System.out.println("no of Lowercase="+lowerCase);
System.out.println("no of Digit="+digit);
System.out.println("no of Spaces="+whitespace);
System.out.println("no of Symbol="+sp);
}
}
```

Question 2

WAP to display string between round braces from the given string.

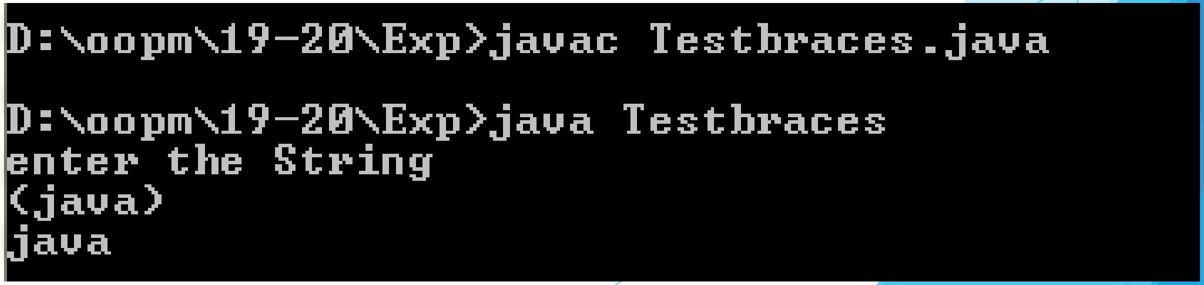
solution

```
import java.util.*;
class Testbraces
{
 public static void main(String args[])
{
  int index1=0,index2=0;
  String S1="(";
  String S2=")";
Scanner sc=new Scanner(System.in);
System.out.println("enter the String");
String a=sc.nextLine();
```

contd…

Contd…

```
if (a.contains(S1))
 {
        index1 = a.indexOf(S1);
  }
if (a.contains(S2))
{
        index2 = a.indexOf(S2);
}
if(index1<index2)
{
     System.out.println(a.substring(index1+1,index2));
}
}
}
```



```
D:\oopm\19-20\Exp>javac Testbraces.java

D:\oopm\19-20\Exp>java Testbraces
enter the String
(java)
java
```

StringBuffer Class

- StringBuffer is a peer class of String.
- While String creates strings of fixed length, StringBuffer creates string of flexible length that can be modified in terms of length and content.
- We can insert characters and substrings in the middle of a string or append another string to the end.
- **Syntax :**

```
StringBuffer stringname=new StringBuffer(“string");
```

- **E.g.**

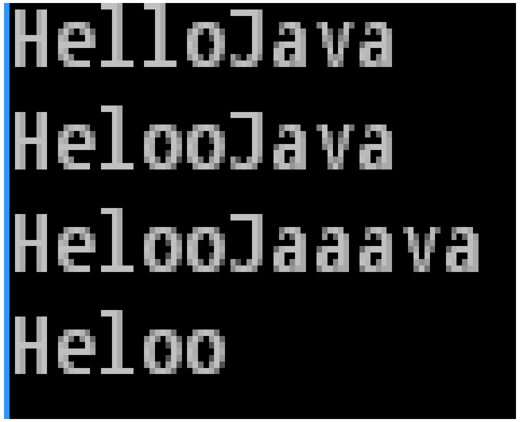
```
        StringBuffer sb=new StringBuffer("Hello");
```

StringBuffer Method

Method	Task
s1.setCharAt(n,’x’)	Modifies the nth character of x
s1.append(s2)	Appends the string s2 to s1 at the end
s1.insert(n,s2)	Insert the string s2 at the position n of the string s1
s1.setLength(n)	If n>s1.length() zeros are added to s1.

Example

```
class StringBufferexp
{
   public static void main(String args[])
  {
    StringBuffer sb=new StringBuffer("Hello");
    sb.append("Java");
    System.out.println(sb);
    sb.setCharAt(3,'o');
    System.out.println(sb);
    sb.insert(7,"aa");
    System.out.println(sb);
    sb.setLength(5);
    System.out.println(sb);
  }
}
```



```
HelloJava
HelooJava
HelooJaaava
Heloo
```

Vectors

- Vector is used to hold objects of any type and any number.
- A vector can be used to store a list of objects that may vary in size.
- We can add and delete objects from the list as and when required.
- **Syntax:**

```
Vector<type> vector_name=new Vector<type> ();
        or
Vector vector_name=new Vector();
```

e.g.

```
Vector<Integer> intVect=new Vector<Integer> ();  //declaring without size
Vector<String> list=new Vector<String> (3);    //declaring with size
```

Vector Methods

Method call	Task Performed
v.addElement(item)	Adds the item specified to the list at the end
v.elementAt(n)	Gives the name of the n th object
v.size()	Gives the number of objects present
v.removeElement(item)	Removes specific item from the vector
v.removeElementAt(n)	Removes the item stored in the n th position of the vector
v.removeAllElements()	Remove all elements in the vector
v.insertElementAt(item,n)	Insert the element at n th position

Example 1:vector of integers

```
import java.util.*;
public class VectorDemo
{
   public static void main(String[] args)
  {
   // create an empty Vector vec without size
   Vector<Integer> vec = new Vector<Integer>();
   // use add() method to add elements in the vector
   vec.addElement(4);
   vec.addElement(3);
   vec.addElement(2);
   vec.addElement(1);
   for(int i =0; i <= vec.size() - 1; i++)
   {      System.out.print(vec.elementAt(i) + " ");    }
  }
}
```

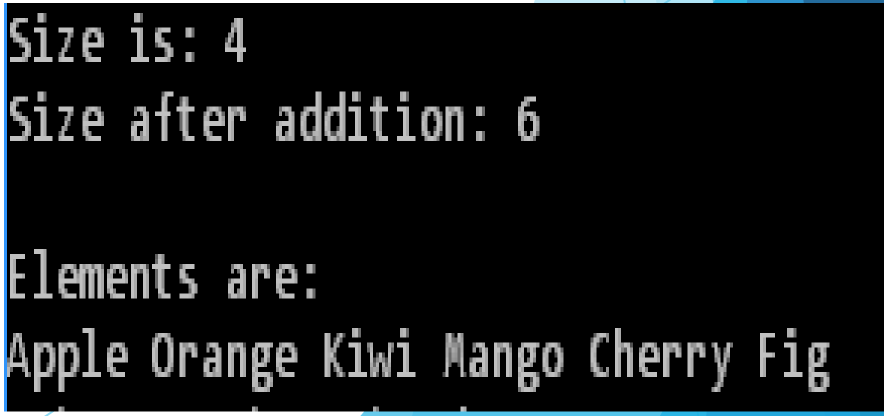
Example 2:vector of string

```
import java.util.*;
public class VectorExample
{
    public static void main(String args[])
    {
        /* Vector of initial capacity(size) of 2 */
        Vector<String> vec = new Vector<String>(2);
        /* Adding elements to a vector*/
        vec.addElement("Apple");
        vec.addElement("Orange");
        vec.addElement("Mango");
        vec.addElement("Fig");
        System.out.println("Size is: "+vec.size());
```

contd…

contd…

```
    vec.insertElementAt("Kiwi",2);
    vec.insertElementAt("Cherry",4);
    /*size and capacityIncrement after two insertions*/
    System.out.println("Size after addition: "+vec.size());
    /*Display Vector elements*/
    Enumeration en = vec.elements();
    System.out.println("\nElements are:");
    while(en.hasMoreElements())
      System.out.print(en.nextElement() + " ");
  }
}
```



```
Size is: 4
Size after addition: 6

Elements are:
Apple Orange Kiwi Mango Cherry Fig
```

Enumeration Interface

- The Enumeration interface defines the methods by which you can enumerate (obtain one at a time) the elements in a collection of objects.
- The methods declared by Enumeration are summarized in the following table –

Sr. No.	Method & Description
1	**boolean hasMoreElements( )** When implemented, it must return true while there are still more elements to extract, and false when all the elements have been enumerated.
2	**Object nextElement( )** This returns the next object in the enumeration as a generic Object reference.

Wrapper Classes

- Wrapper class in java provides the mechanism to convert primitive into object and object into primitive.
- Wrapper classes in java as the name wraps or encapsulates the primitive data types such as int, char etc. and gives them an object like appearance which is mostly used in Collections because in Collections we can only add objects.
- They are also known as **Type Wrappers** because they convert the data types into a class type.

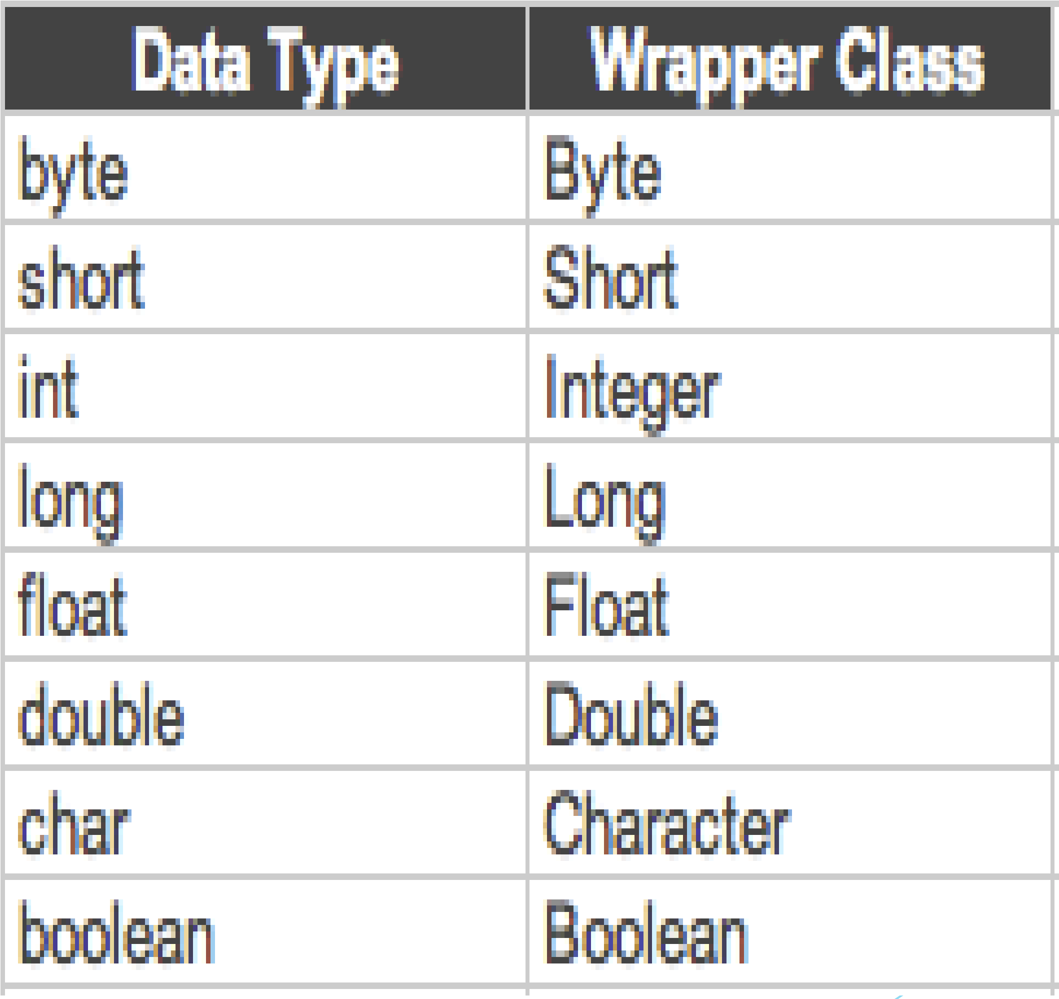
Why To Use Wrapper Classes

- Since java is object oriented language in which every single element should be treated as object whether it is a file, image or anything but it uses primitive data types which are not actual objects.

- we cannot pass primitive data types by reference, they are passed by value and also we cannot make two references which refer to same data.

- Java only uses these primitive data types for performance reasons and hence there should a way in which we can convert them into objects and for this designers create Wrapper Classes.

Wrapper classes for converting simple types



Data Type	Wrapper Class
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

Converting primitive numbers to object numbers using constructor method

Constructor Calling	Conversion Action
Float FloatVal=new Float(f);	Primitive float to Float object
Integer IntVal=new Integer(i);	Primitive integer to Integer object
Double DoubleVal=new Double(d);	Primitive double to Double object
Long LongVal=new Long(l);	Primitive long to Long object

Converting object numbers to primitive number using typeValue() method

Method Calling	Conversion Action
int i=IntVal.intValue();	Object to primitive integer
float f=FloatVal. floatValue();	Object to primitive float
long l=LongVal.longValue();	Object to primitive long
double d=DoubleVal. doubleValue();	Object to primitive double

Converting numbers to string using toString() method

Method Calling	Conversion Action
str=Integer.toString(i);	Primitive integer to string
str=Float.toString(f);	Primitive float to string
str=Double.toString(d);	Primitive double to string
str=Long.toString(l);	Primitive long to string

Convert string object to numeric objects using the static method ValueOf()

Method Calling	Conversion Action
IntVal=Integer.ValueOf(str);	Converting string to Integer object
LongVal=Long.ValueOf(str);	Converting string to Long object
FloatVal=Float.ValueOf(str);	Converting string to Float object
DoubleVal=Double.ValueOf(str);	Converting string to Double object

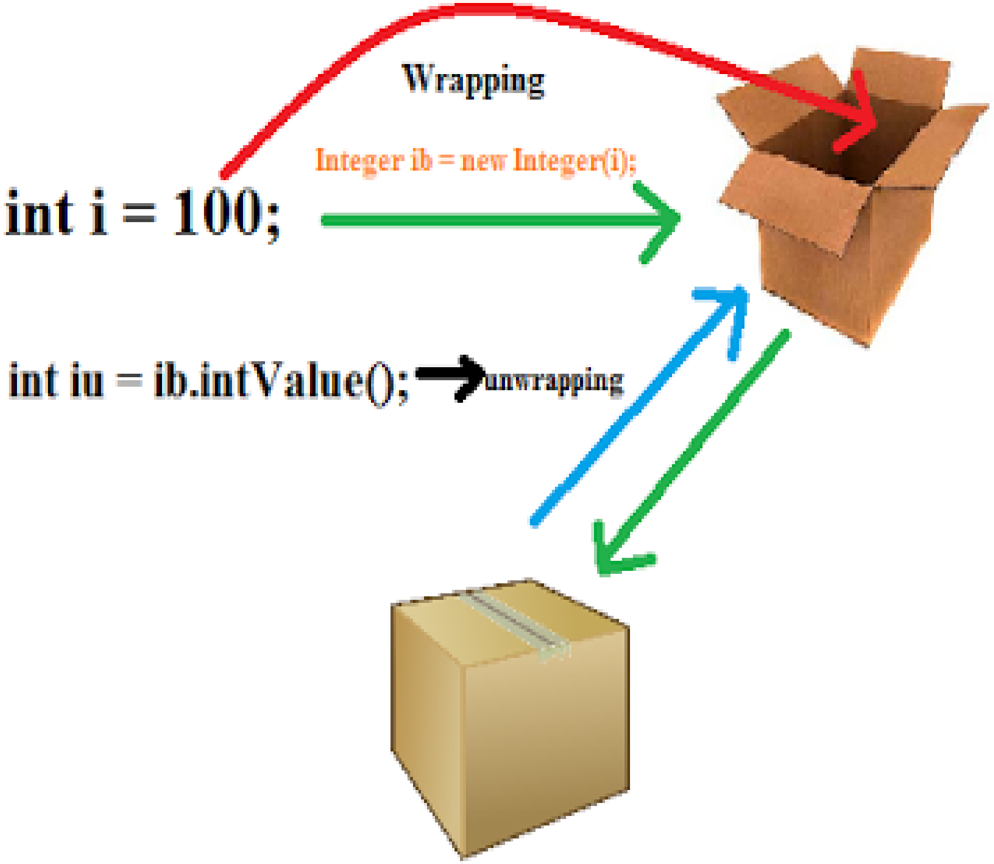
Converting numeric strings to primitive numbers using parsing methods

Method Calling	Conversion Method
int i=Integer.parseInt(str);	Converts string to primitive integer
long l=Long.parseLong(str);	Converts string to primitive long
float f=Float.parseFloat(str);	Converts string to primitive float
double d=Double. parseDouble(str);	Converts string to primitive double

Autoboxing and Unboxing

- Converting primitive data types to wrapper class types automatically is called Autoboxing.
- Converting wrapper class type into primitive types automatically is called Unboxing.
- The compiler generates a code implicitly to convert primitive type to the corresponding wrapper class type and vice-versa.
- E.g.

```
        Double d_object=90.6;
        double d_primitive=d_object; (instead of d_object.
doubleValue();)
```



```
Integer ib = new Integer(i);
int i = 100;
int iu = ib.intValue();
```

Program without toString()

```
class Student{
 int rollno;
 String name;
 String city;
 Student(int rollno, String name, String city){
 this.rollno=rollno;
 this.name=name;
 this.city=city;
 }
   public static void main(String args[]){
   Student s1=new Student(101,"Raj","lucknow");
   Student s2=new Student(102,"Vijay","ghaziabad");
   System.out.println(s1);
   System.out.println(s2);
 }
}
```

Output: **Student@1fee6fc** //s1 and s2 prints the hashcode values of the objects

```
Student@1fee6fc
Student@1eed786
```

Example of toString()

```
class Student {
    int rollno;
    String name;
    String city;
    Student(int rollno, String name, String city){
    this.rollno=rollno;
    this.name=name;
    this.city=city;
 }
 public String toString()       //overriding the toString() method
{     return rollno+" "+name+" "+city;   }
 public static void main(String args[]){
   Student s1=new Student(101,"Raj","lucknow");
   Student s2=new Student(102,"Vijay","ghaziabad");
   System.out.println(s1);       //compiler writes here s1.toString()
   System.out.println(s2);         //compiler writes here
s2.toString()
 }
}
```

```
Ouput :    101 Raj lucknow
            102 Vijay ghaziabad
```