++++++++++++++++++
Arrays and Strings
++++++++++++++++++
 1. Need of arrays
 2. Array declaration
 3. Array construction
 4. Array initialization
 5. Array declaration,construction,intialization in single line.
 6. length vs length() method
 7. Anaonmyous arrays
 8. Array element assignments
 9. Array variable assignments.

Why do we need arrays?
 => Single variable can hold only value at a time, if we want to store many values then we need to store them
    in mulitple variables.
 => Mulitple variable will hold mulitple values, but remembering the variable names to get the data back becomes headache for
    the programmer.
 => To resovle this problem we need to go for "Arrays".

+++++++++++++++++
Array declaration
+++++++++++++++++
 Single dimension array
      int[] a; //prefered
      int a[];

 Two dimension array
      int[][] a;//prefered
      int [][]a;
      int a[][];
      int[] [] a;
      int[] a[];
      int []a[];

 Three dimension array
        int[][][] a;
      int [][][]a;
      int a[][][];
      int [] [][]a;
      int [] a[][];
      int [] []a[];
      int [][] a[];
      int []a[] [];

Which of the following are valid and mention the dimension?
 int[] a,b; ===> valid[a-1d,b-1d]
 int[] a[],b; ===> valid[a-2d,b-1d]
 int[] []a,b; ===> valid[a-2d,b-2d]
 int[] a,[]b; ===> invalid
 int[3] a; ====> invalid

Note: if we want to specify the dimension before the variable, that rule is applicable only for 1st variable, from second variable
      onwards it is not applicable, if we try to it would result in "CompileTime" Error.

While declaring an array, we can't specify the size, if we try to do so it would result in "CommpileTime" Error.

```
+++++++++++++++++
ArrayConstruction
+++++++++++++++++
 int a[];
 int a[] =new int[3];
```

=> In java arrays are treated as objects, so using new keyword we need to construct and array.
=> For every array type corresponding classes are available.
=> These classes are part of java language and it would not be available to "programmer"
=> These classes would be avaiable at the run time so we say these classes as "Proxy classes".

ArrayType	Corresponding class
int[]	[I
byte[]	[B
boolean[]	[Z

Note::
class names are treated as an "identifers".
Rules for naming an identifer
a. symbols allowed(a to z, A to Z, 0 to 9, _ and $]

=> Whenever we print reference variable,internally jvm calls toString() on that variable and prints the "hashcode" value.
=> int a= 10; //primitive variable
=> Student s= new Student();
System.out.println(s);//s.toString()

```
++++++++
Snippets
++++++++

=> int[] a = new int[]; //CE
=> int[] a = new int[0];
    System.out.println(a[0]);//RE:: ArrayIndexOutOfBoundsException
    System.out.println(a.length);//0

=> int[] a = new int[-3]; //RE:NegativeArraySizeException
    System.out.println(a[0]);
    System.out.println(a.length);

=> int[] a =new int['a']; //valid

   byte b= 10;
   int[] a =new int[b]; //valid

   short s = 20;
   int [] a =new int[s]; //valid

   int[] a = new int[10L];//CE
   int[] a = new int[10.5];//CE

-> Integer a =new Integer(10);
   int[] b = new int[a.MAX_VALUE];
```

```
   int[] c = new int[a.MAX_VALUE + 1];//RE: OutOfMemoryError[heapmemory is full]

How multiDimension Arrays in java are implemented?
  => Array of arrays,it is not in matirx form.
  => Advantage is memory utilization

++++++++
example
++++++++
  2-D => 1-D + 1-D
  3-D => 2-D + 1-D

eg#1.
 int[][] a =new int[2][];//JaggedArray
    a[0] = new int[2];
    a[1] = new int[3];

eg#2.
 int[][] a =new int[2][3];//Homogeneous array

++++++++++++++++++++++
usecase of JaggedArray
++++++++++++++++++++++
=> Store the information of students belonging to 8th sem which has 3 sections, A->
50, B->35, C -> 51
      int[][] a = new int[3][];
       a[0] =new int[50];
         a[1] =new int[35];
       a[2] =new int[51];


Which of the following declarations are valid?
 int[] a = new int[]; //invalid
 int[][] a = new int[3][];//valid
 int[][] a = new int[][4];//invalid
 int[][][] a =new int[3][][];//valid
 int[][][] a =new int[][4][];//invalid

Q>
public class Sample
{
      public static void main(String[] args)
      {
            int[][] a= new int[3][4];
            System.out.println(a);//[[I@...
            System.out.println(a[0]);//[I@..
            System.out.println(a[0][0]);//0
      }
}

Q>
public class Sample
{
      public static void main(String[] args)
      {
            int[][] a= new int[3][];
            System.out.println(a);//[[I@...
            System.out.println(a[0]);//null
            System.out.println(a[0][0]);//RE: NullPointerException
```

```
     }
}
```

Note:
 => if we try to access the array element with out of range index, then it would result in "ArrayIndexOutOfBoundsException".
 => if we try to work with the null values, then it would result in "NullPointerException".

Q>
Which among the following declaration is valid?
 1. int[] a;
 2. int a[];
 3. int []a;
 4. int[6] a;
Predict the answer.
A. 1,2,3
B. 1,2,4
C. 2,3,4
D. None of the above

Answer : A

Q>
```
1.   int[]  a;
       a=new int[5];

2.  int[] a =new int[5];
```

Both the declarations are they same?
A. yes
B. no

Answer: A

```
3. int n[][] = {{1,3},{2,4}};
     for(int i=n.length-1;i>=0;i--){
            for(int y:n[i])
                  System.out.print(y);
     }
```

A. 1234
B. 2313
C. 3142
D. 4231
E. 2413

Answer: E

```
Q> int nums1[]  = {1,2,3};
       int nums2[]  ={1,2,3,4,5};
       nums2 = nums1;
       for(int x:nums2)
            System.out.print(x+":");
```
What is the result?
A. 1:2:3:4:5

B. 1:2:3:
C. Compilation fails
D. ArrayIndexOutOfBoundsException

Answer:: B

Q> int data[] = {2010,2013,2014,2015,2014};
```
       int key = 2014;
       int count=0;
       for(int e:data){
            if(e!=key){
                  continue;
                  count++;
            }
      }
      System.out.println(count+" found");
```
What is the result?
A. Compilation fails
B. 0 found
C. 1 found
D. 3 found

Answer:: A

Q>
```
class Test{
      public static void main(String[] args){
                  int numbers[];
                  numbers =new int[2];

                  numbers[0] = 10;
                  numbers[1] = 20;

                  numbers = new int[4];
                  numbers[2] = 30;
                  numbers[3] = 40;

                  for(int x: numbers)
                        System.out.print(" " + x);//0  0 30 40
      }
}
```
What is the result?
A. 10 20 30 40
B. 0 0 30 40
C. Compilation fails
D. An exception is thrown at runtime

Answer: B

Q>
```
  int wd = 0;
  String days[] = {"sun","mon","wed","sat"};
  for(String s:days){
      switch(s){
            case "sat":
            case "sun":
                        wd-=1;
                        break;
```

```
            case "mon":
                        wd++;
            case "wed":
                           wd+=2;
      }
  }
 System.out.println(wd);// wd = -1, 0, 2, 4 , 3
```

What is the result?

A. 3
B. 4
C. -1
D. compilation fails

Answer: A