Arrays

- Arrays are data structures consisting of related data items of the same type.
- Array is a group of elements or components containing values that all have the same type.

```plaintext
\* Sum = c[0] + c[1] + c[2]
→ Sum of the first three elements.
\* x = c[6] / 2
```

must be nonnegative

- Declare and creating:
  - specify the type of the array elements
  ```java
  int[] c = new int[12];
  int[] c, declare
  c = new int[12]; create
  public class InitArray {
      public static void main() {
          int[] array;
          array = new int[10];
          for (int counter = 0; counter < array.length; counter++)
              System.out.println(counter, array[counter]);
      }
  }
  ```
* using an array initializer

```c
int array = { 10, 20, 30, 40, 50 };
for (int counter = 0; counter < array.length; counter++)
    printf("%5d %8d\n", counter, array[counter]);
```

```c
for (int counter = 0; counter < array.length; counter++)
    array[counter] = 2 * (counter + 2);
```

```c
for (int counter = 0; counter < array.length; counter++)
    printf("%5d %8d\n", counter, array[counter]);
```

```c
int array = { 87, 68, 100 };,
int total = 0;
for (int counter = 0; counter < array.length; counter++)
    total += array[counter];
printf("total = %d\n", total);
```

### Finding the largest element:
```c
int
double max = array[0];
for (int i = 1; i < array.length; i++)
    if (array[i] > max)
        max = array[i];
```
Name: string
Grades []: int

constructor
+ Receive the name as a parameter for the student
+ Receive the grades list as a parameter for the student

Fill grades
+ ( )
+ Receive the student's name, fill the list of grades for the student

Display
+ ( )
+ Receive the student's name, display the grades list for the student

Average ( )
+ Receive the student's grades list, calculate the average grade

Max ( )
+ Receive the student's grades list, return the highest grade

Bonus ( )
+ Receive the student's grades list, return the bonus grade

Copy (student)
+ Receive the student's grades list, copy to another student's grades list

Location (int, x)
+ Receive the number of the student and their location

---

Import java.util.Scanner;
Public class Student {  
    private string name;
    private int [] grades;

    public Student (String sName, int size) {  
        name = new string (sName);
        grades = new int [size];
    }
public void bonus (int b) {
    for (int j = 0; j < grades.length; j++)
        grades[j] += b; //
}

public void copyTo (Student s) {
    s.name = name;
    for (int j = 0; j < grades.length; j++)
        s.grades[j] = grades[j]; //
}

public int location (int x) {
    for (int j = 0; j < grades.length; j++)
        if (grades[j] == x)
            return j;
    return -1; //}
}

import java.util;

public class TestStudent {
    public static void main () {
        Scanner scan = new Scanner (...);
        System.out.println ("How many grades per student?");
        int n = scan.nextInt ();//
        Student s1 = new Student ("Ahmad", n);
        Student s2 = new Student ("Jamil", n);
        Student s3 = new Student ("Salim", n);
}
S1. fillGrades();
S2. fillGrades();
S1. CopyTo(S3);
S0.p("Ahmad Grades:");
S1. display();
S0.p("Jamal Grades:");
S2. display();
S0.p("Salim Grades:");
S3. display();
S0.p("The Average of Ahmad = " + S1. average());
S3. bonus(20);
S0.p("Salim Grades after 20 bonus:");
S3. display();
int loc = S1. location(90);
if (loc == -1)
  S0.p("Ahmad did not score 90.");
else
  S0.p("Ahmad score 90 as the " + loc + " grade");
  S0.p("The best grade for Jamal = " + S2. max());
}}
```java
public void display() {
    public void fillGrades() {
        Scanner key = new scanner (System.in);
        s.o.p ("Enter" + grades.length + " grades:");
        for (int i = 0; i < grades.length; i++)
            s.o.p ("Grade " + i + ": ");
        grades[i] = key.nextInt();
    }

    public void display() {
        s.o.p ("student name:" + name);
        s.o.p ("Grades:");
        for (int i = 0; i < grades.length; i++)
            s.o.p (grades[i] + ",");
        s.o.p ("\n");
    }

    public double average() {
        double sum = 0.0;
        for (int i = 0; i < grades.length; i++)
            sum += grades[i];
        return sum / grades.length;
    }

    public int max() {
        int m = grades[0];
        for (int i = 1; i < grades.length; i++)
            if (grades[i] > m)
                m = grades[i];
        return m;
    }
}
Finding the smallest index for largest element.

```java
int max = array[0];
int index = 0;
for (int i = 1; i < array.length; i++)
    if (array[i] > max)
        max = array[i];
        index = i;
```

* shift elements left.

```java
int temp = array[0];
for (int i = 1; i < array.length; i++)
    array[i-1] = array[i];
array[array.length-1] = temp;
```