

return ← function سے اُطرح ہونے والا

بعض صورتوں میں break سے loop کو ختم کیا جاتا ہے

overloading

ایک ہی نام کے function کو مختلف data type کے arguments کے ساتھ لکھا جاتا ہے

```
public static void main (String[] args) {  
    test (1, 2);  
}
```

```
public static void test (int x, int y) {  
    System.out.println("one");  
}
```

```
public static void test (double x, int y) {  
    System.out.println("two");  
}
```

```
public static void test (int x, double y) {  
    System.out.println("three");  
}
```

```
public static void test (double x, double y) {  
    System.out.println("four");  
}
```

```
public static void test (int x, int y, int z) {  
    System.out.println("five");  
}
```

Write the output

1) test (3, 4);
→ one ⇒ int, int ⇒ 1

2) test (3.2, 2);
→ two ⇒ { int, int, double }
int is not a double
double is 2nd choice

3) test (3.2, 4, 1);
→ four
~~double, int, int, double~~
~~double, double, int, double~~

ex Suppose the method of signature (int x, int y) was ~~not~~ is not in the program

then: test (1, 2); and we have those

- ① test (double x, double y);
- ② test (double x, int y);
- ③ test (int x, double y);

الكل
هنا في 3 2 2 في نص واحد
there is an error named ←
"ambiguous"
لأننا في النص 3 2 2 في نص واحد

the overloading is an example of Encapsulation

double l1 name = { 1.2, 3.4, 5.6 };

char l2 name = { 'a', 'b', 'c' };

Arrays

How to declare an array??

① `double [] name = new double [size]`

or

② `double [] name;
name = new double [Size]`

Subscripted Value:

ex `runs[3] = 5.3;` فإنه في 5.3

Note: We can get array's size by this syntax
`name.length`

* Default Values in Arrays:

For int or double arrays $\rightarrow 0$ by default
For char arrays $\rightarrow '\backslash u 0000'$ by default
For boolean array $\rightarrow$ false by default

Array initialization

`double [] name = { 1, 2.3, 4 };`

`char [] name = { 'c', 'a' };`

// loops in Array:

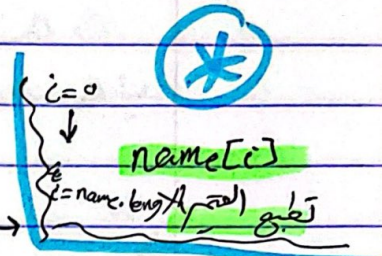
type I:

for (int i = 0; i < name.length; i++)
 ← increment i يعني يغير i بالزيادة

type II: for each

ex. for (int i : name)

System.out.println(i)



* 1. يطبع اعداد على القيمة
 * 2. لا يغير المتغير على الخلايا بالمتسلسلة

Note //

لو سببت داخل for each اخطأ تكتب على القيمة مثل (x=10;)
 ← لن يعمل error ولأنه لن يغير شيئاً.

How to pass an anonymous Array to a method?

print Array (int [] {60, 70});

لا نكتب العنصر لها مرة اخرى بعد ارسال القيمة.

// Methods return (array) type :

الفكرة هو

انه اعمل

واخترت في

two

دونه تغير

مقدار one

القيمة

Main {

int [] one = {1, 3, 5, 4};

int [] two;

two = addThree(one);

for (int i : two)

System.out.println(i);

لنرم الحجم size

public static int [] addThree (int [] x) {

int [] result = new int[x.length];

for (int i = 0; i < result.length; i++)

result[i] = x[i] + 3;

Note

```
int [] one = {1, 2, 3};  
int [] two;
```

⇒ two = one;

• هنا، لم نعمل Copy، المسمى الكاذب هو two
• يشير على نفس المكان الذي يشير إليه one

∴ 2 arrays' pointers are directed to the same elements

جرب عدد 1 ← ثم اطلع two

// How to copy an array to another

1) Using for loop // not recommended

```
int [] one = {1, 3, 5};  
int [] two = new int [one.length];
```

```
for (int i=0; i < one.length; i++)  
    two[i] = one[i]
```

2) Using Built-in method

```
int [] one = {1, 3, 2};  
int [] two = {4, 5};
```

~~System~~ System.arraycopy (Source, ^{Source}position, destination, ^{dest.}position, length)

① ② ③ ④ ⑤

System.arraycopy(two, 0, one, 1, 2);

output

1
4
5

How to sort an Array?

Arrays.sort (name);

لصناديق

* اذا بدى تنازلي ، التنازل في array اخرى بشكل عكسي
او اطلع بشكل عكسي
(او بترتيب reverse - فنتيجة حاجز)

// Variable length , Argument list

ex Mainf

anonymous

→ (new int[] {1,2,3})

int [] one = {1, 2, 3};

do It (4, 5); ✓

numbers ✓

do It (new int[] {4, 5}); ✓

anonymous array ✓

do It (one); ✓

array ✓

}

public static void doIt (int... X) {

~~for~~
for (int i=0; i < X.length; i++)

System.out.println (X[i]);

}

* أي شيء عباره لـ ، يعتبره array

ex

do It (4.5, new int[] {2, 3});

public static void (double y, int... X);

أخذ الاول ك double والثاني ك array

لـ 4.5 بطبع X ← بطبع 2 (لأنه 4.5 عن 2 في X)

Rules ① $(int \dots x) \Rightarrow$ argument آخر

② لازم واحدة فقط في ال Parameters الواحد

Strings Rule 1: How to scan string
Rule 2: $\sim \sim \sim$ Char

```
string name;
```

```
char gender;
```

```
Scanner input = new Scanner(System.in);
```

Rule ①

```
name = input.next();
```

Rule ②

```
gender = input.next().charAt(0);
```

```
or ( string x = input.next();  
      gender = x.charAt(0);
```

Note in factorial, use (long).

Note to exit ~~the~~ ^{from the} program (like return 0 in C)

use `System.exit;`

Array ^{مассив}

Note in for $\Rightarrow$ `println(array[i])`

in for-each $\Rightarrow$ `println(i)`

Q1 output?

```
int [] one = { 1, 2, 3 };
```

```
int [] two;
```

```
two = one;
```

```
for (int i = 0; i < one.length; i++)
```

```
one[i] += 10;
```

```
for (int j : two)
```

```
System.out.println(j);
```

→

11

12

13

```
for (int j : one)
```

```
System.out.println(j);
```

11

12

13

السبب

لأن one و two مؤشران على نفس المكان.

ف عندما نغير one نغير القيمة التي مؤشر على.

← ننغير القيمة التي يفرع عليها two

So :- this is not copying Arrays.

Q.

int [] one = { 1, 2, 3, 4, 5, 6 };

int [] two = { 10, 20, 30, 40 };

suppose

System.arraycopy(two, 1, one, 0, 2);

for (int i : one)
System.out.println(i);

output →

20
30
3
4
5
6

Suppose

System.arraycopy(one, 3, two, 2, 3);

for (int i : two)
System.out.println(i);

output →

error ; the size of two is 4

and while copying → I need
more one element

Q

int [] one = { 10, 20 };

~~int []~~ two = one;

for (int i = 0; i < two.length; i++)

System.out.println(i);

→ output is $\begin{matrix} 0 \\ 1 \end{matrix}$ (known size, unknown values)

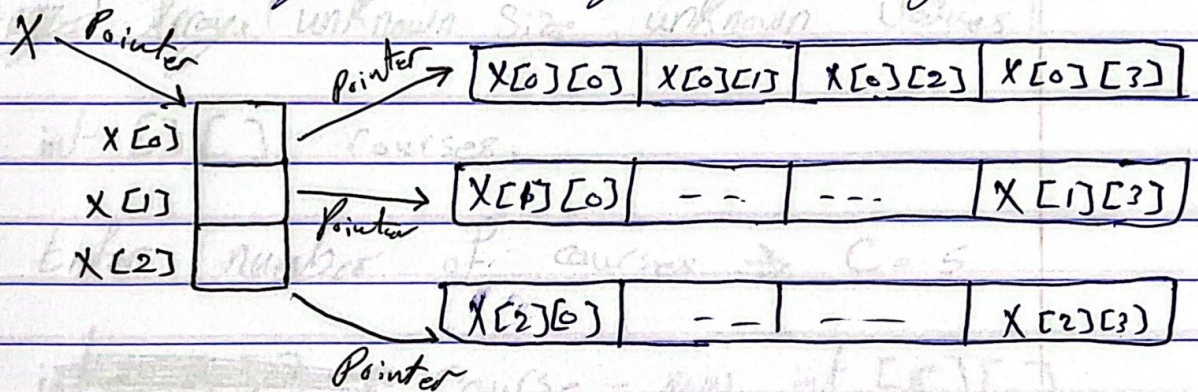
Since this is not for-each $\ddot{u}$

Two - 1) Arrays.

```
int [][] X = new int [3][4];
```

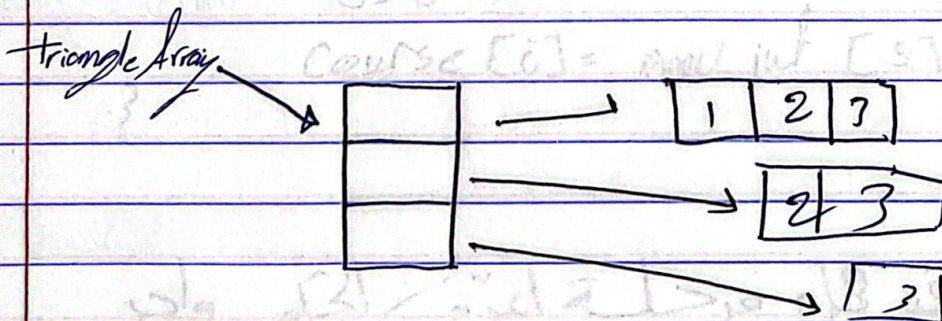
```
// X.length = 3
```

```
// X[0].length = X[1].length = X[2].length = 4
```



Ragged Arrays

```
int [][] triangleArray = { { 1, 2, 3 }, { 2, 3 }, { 3 } };
```



لا يكون ال arrays الاخلاق Size غير مساوي

Ragged Array (Known Size, unknown Values)

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

```
int rag[0] = new int [4];
```

من اجزاءه غير افترق

```
rag[1] = new int [5];
```

```
rag[2] = new int [3];
```

في array داخلية : افترقا في array

Ragged Arrays (unknown Size, unknown Values)

```
int [][] courses;
```

Enter number of courses $\Rightarrow C = 5$

```
int [][] course = new int [C] [];
```

```
for (int i = 0; i < course.length; i++) {
```

Enter number of students

S = 6

```
course[i] = new int [S];
```

كأنني افترقا
1-D array

في كل مرحلة افترقا اتجاه واحد

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

```
arr[0] = new int [4]
```

مفرد

Processing 2-D Arrays

```
for (int i=0; i < course Matrix.length; i++)
```

```
for (int j=0; j < Matrix[i].length; j++)
```

```
System.out.println (Matrix[i][j]);
```

Multi-Dimensional Array:

```
int [][][] nums = new int [5][3][2];
```

```
// nums.length = 5
```

```
// nums[1].length = 3
```

```
// nums[1][2].length = 2
```

17/10 Ques

Q use Math.random() to write integer between 0-100 inclusive

inclusive $\rightarrow [0, 100] \rightarrow [0, 101)$

لا ارفع ان احد ليا في حقله $\Rightarrow$

$(int) (Math.random() * 101)$

Q Same Q between 1-100 inclusive

$[0, 100) + [1, 1] = [1, 101)$

$(int) (Math.random() * 100) + 1$

$[1, 101)$

Q ——— between 5, 90 inclusive

$(int) (Math.random() * 86) + 5$

~~[0, 86)~~ $[0, 86)$

$[5, 91)$

Write a program which has one 2-D array named Courses, and it has array of courses contain array of students, Built on Ragged Array

تعريف 2-D
⇒ `int courses[][];` Enter number of courses
`courses = new int [sc.nextInt()][];` ← تعريف 2-D
`for (int i = 0; i < courses.length; i++) {`

تعريف 1-D
⇒ Enter number of students in course i
`Courses[i] = new int [sc.nextInt()];`
← تعريف 1-D

Scanning Values
→ `for (int j = 0; j < courses[i].length; j++) {`
Enter the grade;
`courses[i][j] = sc.nextInt();`
}

قراءة
→ `for (int i = 0; i < courses.length; i++)`
`for (int j = 0; j < courses[i].length; j++)`
→ `System.out.println(courses[i][j]);`