Lec-7
16/03/24 12:47 PM

Array → array
Class → int, float
data size
size → data

4 →
5x4 = 20b

5:
2x, 1+2x4
= 2x+9

arr[0]
2x+0x4
=2x

2x 1+4

arr[2]

```
public class Arrays_Demo {
    public static void main(String[] args) {
        // TODOAuto-generated method stub
        int[]arr= new int[5];
    }
}
```

base address
2x+1 2x+2=2x+2
2x+1+4x4=2x+9

arr = 2x
memory

1 2 5 4
0 1 2 3 4
10 20 30 40 50
2x 2x+4 2x+8

arr(0) =10
arr(1) =20
arr(2) =7

```
public static void main(String[] args) {
    // TODOAuto-generated method stub
//  int[] arr = new int[5];
    int[arr1] = new int[5];
    System.out.println(arr);
    // update
    arr[0] = 10;
    arr[1] = 20;
    arr[2] = 30;
    arr[3] = 40;
    arr[4] = 50;

    // print
    System.out.println(arr[0]);
    System.out.println(arr[1]);
    System.out.println(arr[2]);
    System.out.println(arr[3]);
    System.out.println(arr[4]);
    System.out.println(arr.length);

    for(int= 0; i< arr.length; i++) {
        System.out.print(arr[i] + " ");
    }
    int[] other= arr;
} → 5 ; 7
}
```

arr1=2x
memory
-9 20 30 40 50
0 1 2 3 4
2x
other 2x

number

```
Public class User_Input {
    public static void main(String[] args) {
        // TODOAuto-generated method stub
        Scanner sc= newScanner(System.in);
        int= sc.nextInt();
        int[] arr= newint[n];
        for(int= 0; i< arr.length; i++) {
            arr[i] = sc.nextInt();
        }
        Display(arr);
    }
    public static void Display(int[] arr) {
        for(int= 0; i< arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

5
10 2 4 9 7
2x 2x+4 2x+8

arr1=2x
memory
n=5
arr1=2x
Display(2x)

10 2 4 9 7
2x

7 9
temp
a=9 → 7
b=7 → 9
temp=a;
a=b
b=temp

```
public static void main(String[] args) {
    // TODOAuto-generated method stub
//  int[] arr = new int[] { 10, 20, 30, 40, 50 };
    int[] arr= { 10, 20, 30, 40, 50 };
    System.out.println(arr[0] + " "+ arr[1]);//10 20
    Swap(arr, 0, 1);
    System.out.println(arr[0] + " "+ arr[1]);//
}
public static void Swap(int a, int b) {
    int temp= a;
    a= b;
    b= temp;
}
```

10 20
20 10

10 20 30 40 50
2x
0 1 2 3 4

20 10
a=2x b=2x
temp=a
a=b
b=temp
swap
memory
arr1=2x
Swap(0,1)

```
public static void main(String[] arr) {
    // TODOAuto-generated method stub
    int[] arr= { 10, 20, 30, 40, 50 };
    System.out.println(arr[0] + " "+ arr[1]);// 10 20
    Swap(arr, 0, 1);
    System.out.println(arr[0] + " "+ arr[1]);//
}
public static void Swap(int[] arr, int i, int j) {
    int temp= arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}
```

10 20

10 20 30 40 50

arr1=2x i=0 j=1
temp= 10
arr(0)=arr(1)
arr(1)=temp
arr1=2x
Swap(2x,0,1)
Swap
main

```
public static void main(String[] arr) {
    // TODOAuto-generated method stub
    int[] arr= { 10, 20, 30, 40, 50 };
    int[] other= { 100, 200, 300, 400, 9 };
    System.out.println(arr[0] + " "+ other[0]);// 10 100
    Swap(arr, other);
    System.out.println(arr[0] + " "+ other[0]);//
}
public static void Swap(int[] arr, int[] other) {
    // TODOAuto-generated method stub
    int[] temp= arr;
    arr = other;
    other = temp;
}
```

arr1=2x, other=4x
10

10 20 30 40 50
2x

100 200 300 400 9
4x

arr1=2x
other=4x
temp=2x
arr=4x
other=temp
arr1=2x, other=4x
Swap(2x,4x)
Swap
main

2 1 2 4 7
0 1 2 3

```
public static int Search(int[] arr, int item) {
    for(int= 0; i< arr.length; i++) {
        if(arr[i] == item) {
            return i;
        }
    }
    return-1;
}
```

max=5

max=arr(0)
for(i=1; i<arr.length; i++)
if(arr(i)>max)
max=arr(i)

i=1 arr(1)>max
3 > 2
i=2 arr(2)>max
1 > 3
i=3 arr(3)>max
5 > 3
i=4 arr(4)>max
4 > 5

```
int[] arr= { 1, 34, 56, 78, 11, -7, 90 };
```

90
1
0 1 2 3 4 5 6

i=0 1 > max
i=2 56 > max
i=3 arr(3) > 78 > max

i=4 arr(4) > max
i=5 -7 > 78 ✗
i=6 90 > max

max = 90

max = -∞
-∞ > 25

2, 3, 1, 5, 7
0 1 2 3 4

i=0 arr(0) > max
2 > -∞
i=1 arr(1) > max
3 > 2
i=2 arr(2) > max
1 > 3
i=3 arr(3) > max
5 > 3
i=4 arr(4) > max
7 > 5

[2, 3, 1, 5, 7]
7 5 4 3 2

2 3 4 6 11 13
13 11 6 4 3 2

2 1 3 4 9 7 8 9 11
0 1 2 3 4 5 6

i = 0
j = arr.length-1 → 6

while k<(i>j)
swap(i,j)
i++
j--

Two-P

(5-1)

9 7 5 4 3 2
9 2 3 4 7 9 8
0 1 2 3 5

j i