

# Arrays (Chapter 6)

Hsuan-Tien Lin

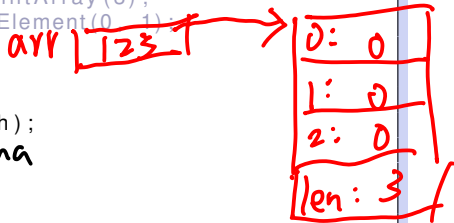
Department of CSIE, NTU

OOP Class, March 22-23, 2010

# Primitive Array (1/2)

```
1 public class ArrayDemo{
2     public static void main(String [] arg){
3         int[] arr = new int[3];
4         //think: intArray arr = new intArray(3);
5         arr[0] = 1; //think: arr.setElement(0, 1);
6         arr[1] = 3;
7         arr[2] = 5;
8         arr[3] = 9;
9         System.out.println(arr.length);
10        arr.length = 5;
11        arr = null;
12    }
13 }
```

ohshoh  
hahaha



- array is a reference by itself
- new, null like usual reference instances
- primitive array: new initialize element to default
- length: read-only
- index out of bound: run time error

## Primitive Array (2/2)

```
1 public class ArrayDemo{
2     public static void main(String [] arg){
3         int [] arr = {1, 3, 5};
4         //compare String s = "HTLin";
5         System.out.println(arr.length);
6     }
7 }
```

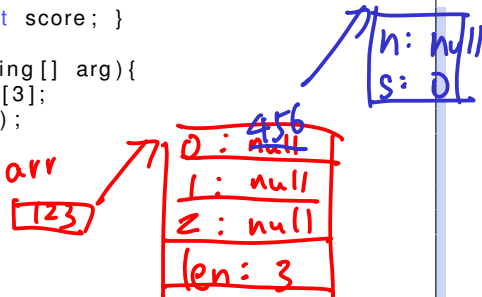
- construct an array instance (with automatic length calculation), then assign its address to the reference variable

# Primitive Array: Key Point

primitive array: reference to “a batch of values”

# Reference Array (1/1)

```
1  class Record{ String name; int score; }
2  public class ArrayDemo{
3      public static void main(String[] arg){
4          Record[] arr = new Record[3];
5          System.out.println(arr[0]);
6          arr[0] = new Record();
7          arr[1] = new Record();
8          arr[2] = arr[0];
9          arr[1] = null;
10         arr = null;
11     }
12 }
```

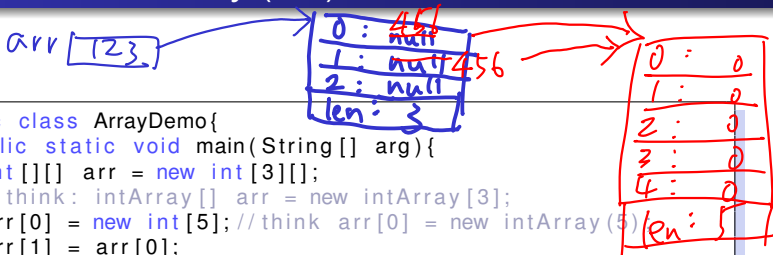


- array is a reference
- reference array: `new` initialize element to null

# Reference Array: Key Point

reference array: reference to “a batch of references”

# Multidimensional Array (1/3)



```
1 public class ArrayDemo{
2     public static void main(String[] arg){
3         int[][] arr = new int[3][];
4         //think: intArray[] arr = new intArray[3];
5         arr[0] = new int[5]; //think arr[0] = new intArray(5);
6         arr[1] = arr[0];
7         arr[2] = null;
8         System.out.println(arr.length);
9         System.out.println(arr[1].length);
10    }
11 }
```

- multidimensional: array of “array references”
- can be irregular

## Multidimensional Array (2/3)

```
1 public class ArrayDemo{
2     public static void main(String [] arg){
3         int [][] arr = new int [3][5];
4         System.out.println (arr.length);
5         System.out.println (arr [1].length);
6     }
7 }
```

- still array of “array references”
- regular, automatic construction



## Multidimensional Array (3/3)

```
1 public class ArrayDemo{
2     public static void main(String [] arg){
3         int [][] arr = {null, {0, 1}, {2, 3, 4}};
4         System.out.println(arr.length);
5         System.out.println(arr[1].length);
6     }
7 }
```

- construct an array, and assign its address to reference

## Multidimensional Array: Key Point

multidimensional array: a special reference array, reference to “a batch of (multidimensional) arrays”