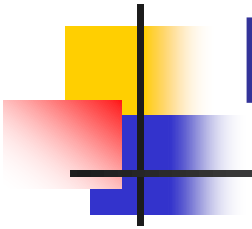
3. Java Language Structure

Data types

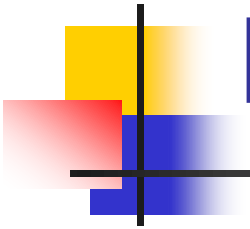
Operators

Control Statements

Data Type : Questions

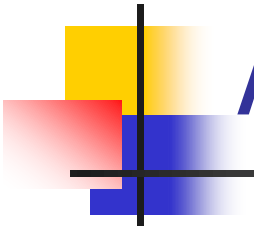


- The term "instance variable" is another name for _____.
- The term "class variable" is another name for _____.
- A local variable stores temporary state; it is declared inside a _____.
- A variable declared within the opening and closing parenthesis of a method is called a _____.



Data Type : Questions

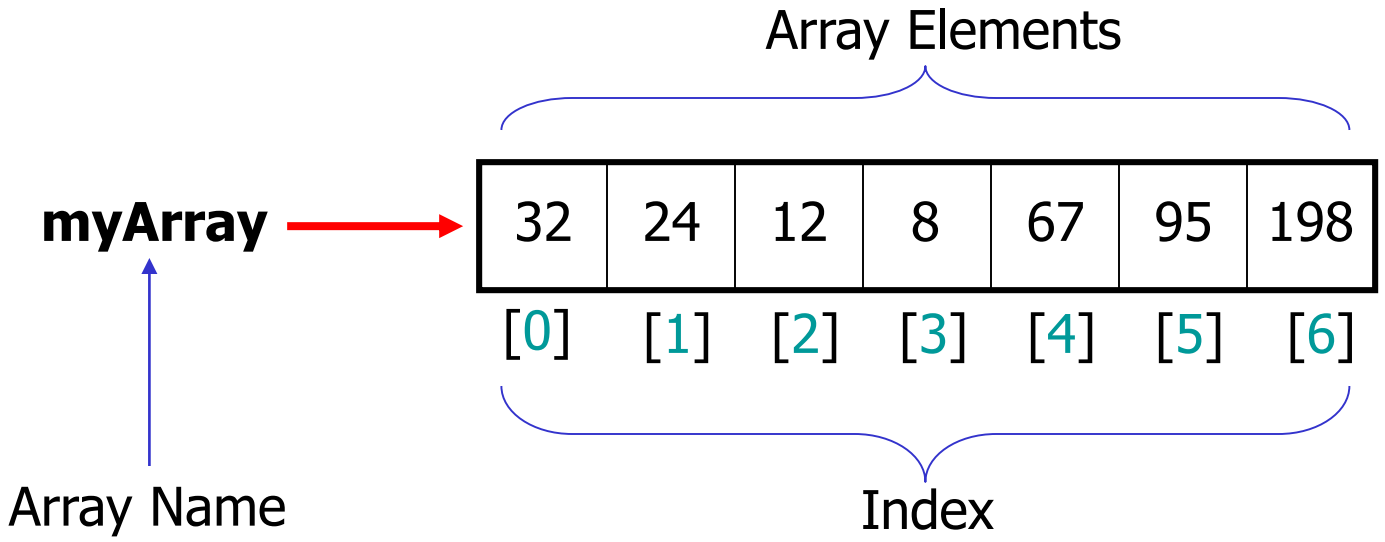
- What are the eight primitive data types supported by the Java programming language?
- Character strings are represented by the class _____.



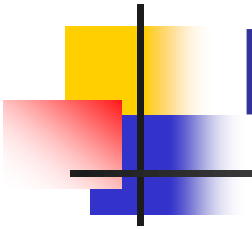
Arrays

- Array is a data structure that holds a collection of data of the same type.
- Conceptually, it is a numbered list of data elements; size is fixed.
- Elements can be accessed by their position, called the index, in the array (starts at 0).

Array as a Collection



We can only store values of the same type in an array.
Array name is a reference to the array.



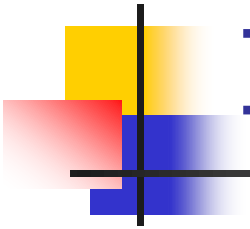
Declaring Arrays

- An array creation has 2 steps:
  - Array declaration – *int myArray[ ];*
  - Physical memory allocation – *myArray = new int[7];*
- These 2 steps can be clubbed together as:

int myArray[ ] = new int[7];

or,

int[ ] myArray = new int[7];

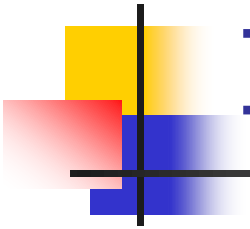


Initializing Arrays

- If all values are known beforehand:

```
int[] myArray = {32,24,12,8,67,95,198};
double[] d = { 0.5, 1.2, 500.201 };
char[] vowels = {'a','e','i','o','u'};
```

- Need not specify the array size - inferred from the assignment.



Initializing Arrays

- Typically, loops are used.

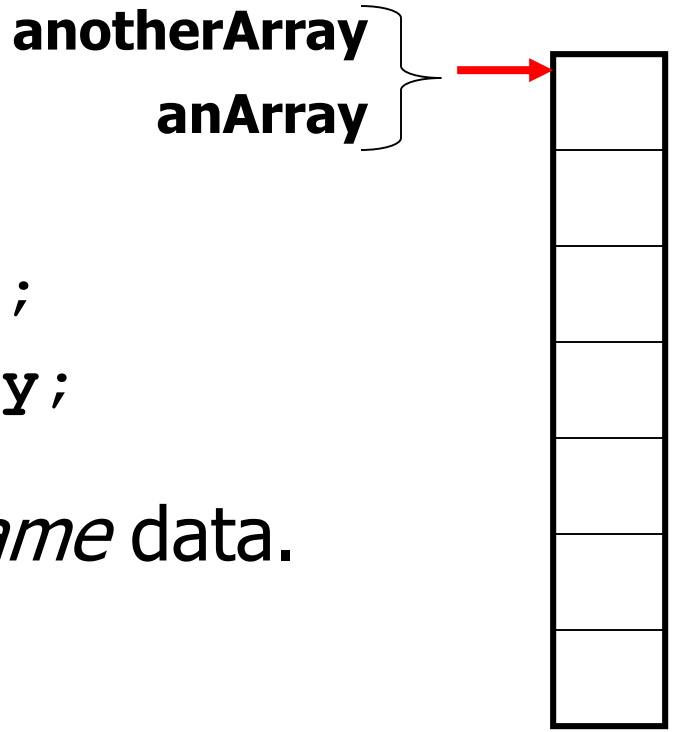
```
int[] array = new int[10];
array[0] = 10;
array[1] = 1;
for( int i=2; i<10; i++ ) {
        array[i] = -1;
}
```

- Accessing a[10], a[11], etc. would cause an error to occur.

Declaring vs. Allocating

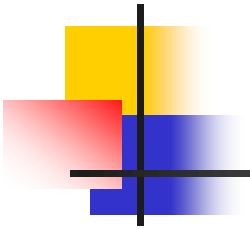
- It is possible to declare an array variable, and use it without explicitly allocating memory for it.
- E.g.,

```
int[] anArray;
int[] anotherArray;
anArray = new int[20];
anotherArray = anArray;
```



- Both variables access the *same* data.

Accessing Array Elements

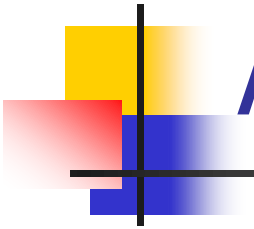


- To access an array element we need:
  - Array variable
  - Position of the desired element in the array - *index*.
- Usage:

  <variable_name>[<index>]

- That is, to access the value of the 6th element of array **myArray** we write:

```
int element = myArray[5];
```



Array Size

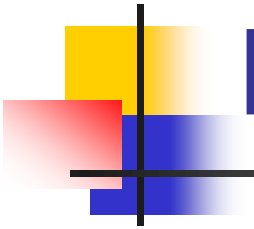
- Array size is the number of elements an array holds, and is fixed when the array is allocated.
- The member variable ***length*** stores the array's size (we cannot modify it!)
- Example:

```
int[] array = new int[12];
System.out.println(array.length);
// Prints out '12'
```

```
public class ArrayTest {
    public static void main(String args[]) {
        int myArray[] = new int[10];
        // Array initializing....
        for(int i=0; i<myArray.length; i++) {
                myArray[i] = i+1;
        }
        System.out.println("\n Array Elements are:\n");
        for(int i=0;  i<myArray.length; i++) {
                System.out.print("   " + myArray[i]);
        }
        System.out.println(" \n ");
    }
}
```

Array Elements are:

1 2 3 4 5 6 7 8 9 10



Exercise

- Write a Java program to find the duplicate values of an array of integer values