# Java Built-in Arrays

Besides collection classes like *ArrayList*, Java also has a built-in array construct that is similar to a Python list.

## *Example*

```
int array[];  // declare an array with elements of type int
array  = new int[10];  // allocate space for 10 elements on heap
array[0]=3;  // set the 0th element to 3.
```

Note that you must both declare the array and allocate the elements for it. As with creating objects, you can do this on one line:

```
int array[]= new int[7];
```

## *Arrays are error-prone*

Once you set the size it's fixed. So

```
array[10]=4;
```

for the example above will give a run-time error.

There is no way to increase the size of an array once you've created it, no append.

## *Array Iteration*

To find the length of an array, use array data member 'length'. 'length' gives the number of elements allocated, not the number inserted.

So here's a loop:

```
int i=0;
while (i<array.length) {
   doSomethingWith(array[i]);
   i=i+1;
}
```

## *Instructor code sample*

Write a class with a main method that creates an array of 10 integers and totals them up.

## Arrays of Objects

The elements of an array can be of any type, including a programmer-defined class.

```
Student studentList[] = new Student[5];  // creates slots for five students
```

Note that the object creation statement above allocates space for five pointers, not five students. You still need to create the students:

```
int i=0;
while (i<studentList.length)
{
        studentList[i]= new Student();
        i=i+1;
}
```

So what does the heap look like after all this?

## null

0 is the default value for an integer.
null is the default value for an object (really, we should say 'object reference)

If we didn't run the loop above, the value of each object in studentList would be null.

## Iterating through list

Once you have a list of objects, you can iterate through it. Be careful, because 'length' just says how many slots have been created, not how many objects:

```
int i=0;
while (i<studentList.length)
{
        Student student= studentList[i];
        If (student!= null)
                System.out.println(student.name);
        i=i+1;
}
```

## Instructor In-Class Demo

Given a BankAccount class, write a Bank class that has a data member which is a built-in array of BankAccount, a totalAssets method that returns the total principal in all accounts, and a main method that creates two bank accounts, adds them to the bank's list, calls totalAssets and prints out the result.

Students: Modify the Mastermind sample so that it uses built-in array, not ArrayList, in the ColorCode class..