

Arrays

An *array* is an indexed sequence of contiguous slots in memory, all of the same type. The type of the elements is called the *base type* of the array. The indexes are numbers, starting with 0.

The type of an array of elements that have base type T is T[]. For example, we would declare A to be an array of ints with

```
int [ ] A;
```

All arrays with the same base type have the same type (the length is not part of the type).

We *construct* an array of N elements with base type T with

```
new T[N];
```

Altogether, a typical line of Java code that declares and constructs an array of 10 Strings is

```
String [ ] page = new String[10];
```

You can then refer to `page[0]`, `page[1]`, etc. up to `page[9]`.

If you have a small number of specific values you want to put into an array, there is another way to construct it.

```
int [ ] A = {23, 45, 67};
```

This creates an array of 3 elements and puts 23 into A[0], 45 into A[1], and 67 into A[2]. This is only practical for initializing small arrays, but it is sometimes useful.

You can find the length (number of entries) of any array `A` with `A.length`. Note that this is a variable, not a function. Remember that this is just the allocated size of the array; nothing says that all of those entries have useful data. Your program needs to manage the data in your arrays.

If you are trying to count the instances of each letter 'a' through 'z' you might declare an array

```
int[] Counts = new int[26];
```

initialize all of the entries of Counts to 0, and each time you see an instance of the *i*th letter increment Counts[*i*].

On the other hand, if you are using an array Primes to keep track of the prime numbers you have found, there is nothing particularly meaningful about indexes you use. Here you probably want to keep a large array Primes:

```
int[] Primes = new int[1000];
```

and also have a variable size to keep track of how many entries of this array you are currently using:

```
int size;
```

If (size == 0) you haven't yet found any primes; otherwise the entries of Primes from index 0 to index size-1 are the primes you have found.

When you find a new prime p you add it to the array and increment size:

```
Primes[size] = p;  
p += 1;
```

Your program will crash if you try to access an entry of the array beyond its length, so test for this:

```
if (size < Primes.length) {  
    Primes[size] = p;  
    size += 1;  
}
```

First Clicker Question: Suppose we have a class Person. How do we declare an array A of 1000 persons?

A) `A = Person[1000];`

B) `Person [1000] A;`

C) `Person[] A = new Person[1000];`

D) `Person[] A = Person[1000];`

What is the signature (header) of a function foo that takes an integer argument x and returns an array of 10,000 ints?

A) `int[10000] foo(int x)`

B) `int[] foo(int x)`

C) `int* foo(int x)`

D) A function can't return an array