



Start a new project in your IDE called **ArraysIntro**. Paste the starter code and answer the following questions:

1. How many **global variables** are there in the program?

2. How many times will the message **"Welcome"** be displayed?

3. What range of numbers are produced in the **generateData()** function?

4. What type of data will **getMax()** return?

The Problem

getMax() will determine the maximum value in the set of 5 temperature values. The logic for this is:

```
if temp1>temp2 && temp1>temp3 &&temp1>temp4&&temp1>temp5
    max = temp1
else if temp2>temp1 && temp2>temp3 &&temp2>temp2&&temp2>temp5
    max = temp2
etc
```

This is not a huge problem because there are only 5 variables to process. But what if there were 100 variables? Or 1000? It would be very awkward to determine the maximum value when using so many variables.

Arrays

All of the data items are the same - **integers**. So, a great strategy is to create a collection and store the data in the collection. A basic kind of collection is an **array**.

To **declare** an array in Java, we need to use [] eg:

```
int[] tempData; //declare the array
```

We also need to tell Java how many items of data will be stored in the array:

```
tempData = new int[5]; //initialise the array
```

We can do all of this in 1 line of code, too:

```
int[] tempData = new int[5];
```

In computer memory, 5 locations are now available to store integers. Each location has its own unique **index**, starting at 0:

tempData				
0	1	2	3	4

So, how can we put data into the array? We can use the indexes to help out!

```
tempData[0] = (int) (Math.random() * 31) + 10;  
tempData[1] = (int) (Math.random() * 31) + 10;  
tempData[2] = (int) (Math.random() * 31) + 10;  
tempData[3] = (int) (Math.random() * 31) + 10;  
tempData[4] = (int) (Math.random() * 31) + 10;
```

Oh! Look at those **index** numbers!!! Let's do this instead:

```
for(int i = 0; i < tempData.length; i++) {  
    tempData[i] = (int) (Math.random() * 31) + 10;  
}
```

Notes

Arrays in Java are used to store data of the same type. The data is **homogenous**.

When we initialise an array like this:

```
int[ ] tempData = new int[5];
```

we set the array's **length**. It won't change. An array is a **static** data structure.

Coding Activity

Can you figure out how to **redesign** the program so that the 5 global variables are replaced by a single global array of 5 integers?

Use the following **algorithm** to code the `getMax()` function:

```
max = 0
```

```
loop i from 0 to 4
    if tempData[i] > max
        max = tempData[i]
end loop
```

```
return max
```

Complete the coding challenges on the unit page