FIT ICT Software Development

Lecture 7

Lecturer : Charles Tyner

This Week:

Arrays

- Introduce the array data structure
- Use of arrays to store tables of values
- Declare an array, initialise an array, and refer to individual array items
- Bounds of an Array

Arrays

- Up until now we have stored **values in variables**, eg if we wanted to store a temperature we create a variable of type double.
  double temperature = 28;

- What if we want to store 7 temperatures, do we create 7 variables with different temperatures.

Handling seven temperature readings

```
class TemperatureReadings
{
  public static void main(String [] args)
  {
     double temperature1, temperature2,
                temperature3,temperature4,

                temperature5,temperature6,
                temperature7;
       // more code will go here
  }
}
```

Code to enter all seven temperatures

Getting one temperature is easy:

```
System.out.println
   ("max temperature for day 1 ?");
temperature1 = input.nextDouble( );
```

BUT

How do you write code to get the 2nd temperature, 3rd temperature and so on….

Code to enter all seven temperatures

```
System.out.println
   ("max temperature for day 1 ?");
temperature1 = input.nextDouble( );
System.out.println
   ("max temperature for day 2 ?");
temperature2 = input.nextDouble( );
System.out.println
   ("max temperature for day 3 ?");
temperature3 = input.nextDouble( );
....
```

=> very inefficient

Declaring and Creating an Array

Arrays are used to hold a ***collection of items all of the same type***.

Array creation is a **two stage process:**

1. **Declare an array.**

   ```
   ArrayType arrayName[];
   ```

2. **Create memory to store the array.**

   ```
   arrayName = new ArrayType [sizeOfArray];
   ```

Creating an array of temperatures

Since each temperature is of type **`double`** an array of temperature readings is **declared as follows:**

```
double temperature[];
```

Since there will be seven temperature readings, **memory is reserved** for this array as follows:

```
temperature = new double[7];
```

Solution is:
Create an Array

Declaring and Creating an Array

Just like the creation of objects, we could have **combined** the two procedures of **declaring** and **creating** an array in **one step**:

```
double temperature[] = new double[7];
```

Array Name

Array Type

To allocate **12 elements** for integer array **c**, the declaration following is used:

```
int c[ ] = new int[ 12 ];
```

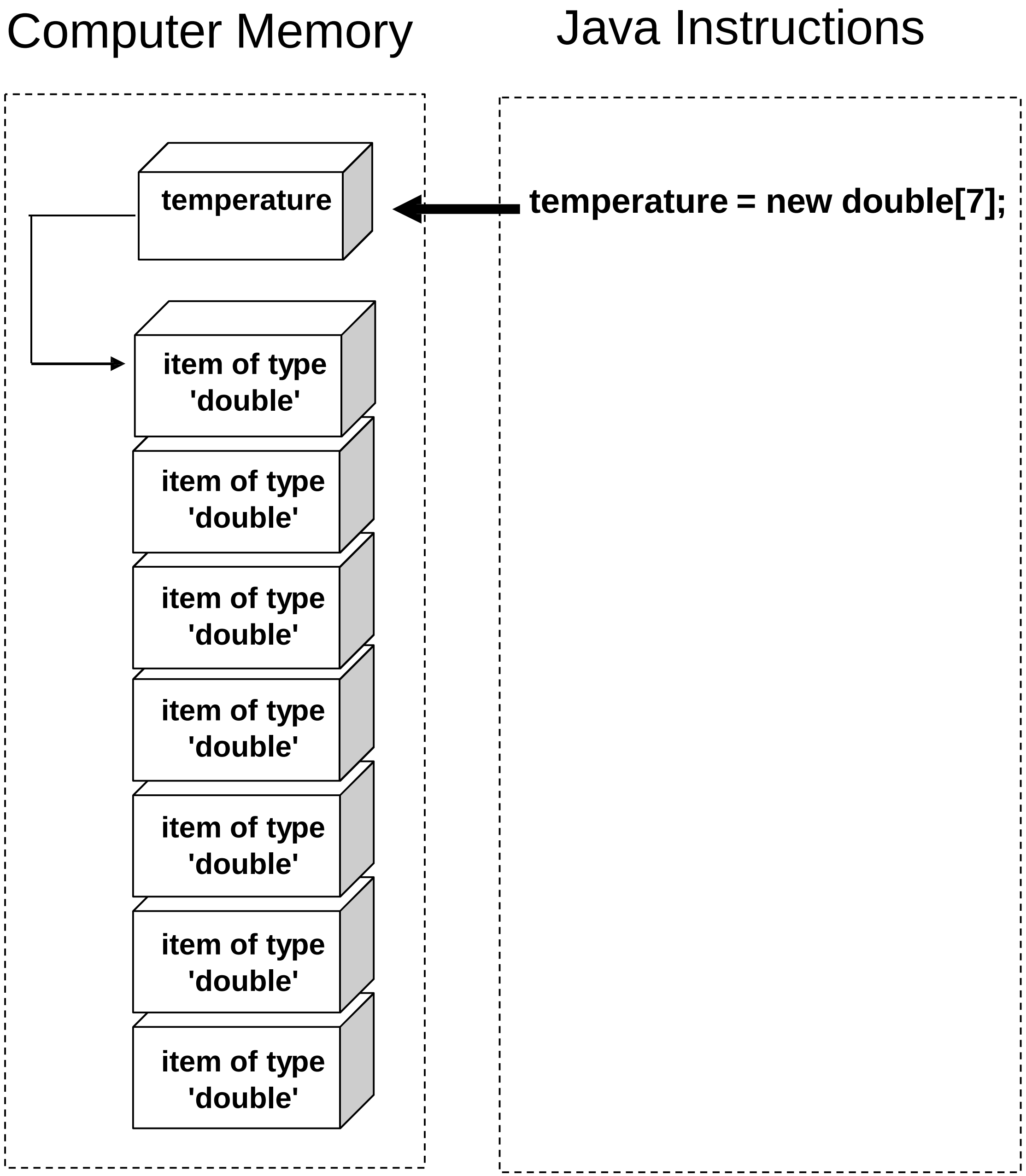
Array Name

Array Allocation

When arrays are allocated, ie declared and created, the elements are automatically initialised to:

zero for the numeric primitive data type variables
false for boolean variables
null for references (any non primitive type)

The effect on computer memory of declaring an array of values of type 'double'



Computer Memory

Java Instructions

temperature

item of type 'double'

item of type 'double'

item of type 'double'

item of type 'double'

item of type 'double'

item of type 'double'

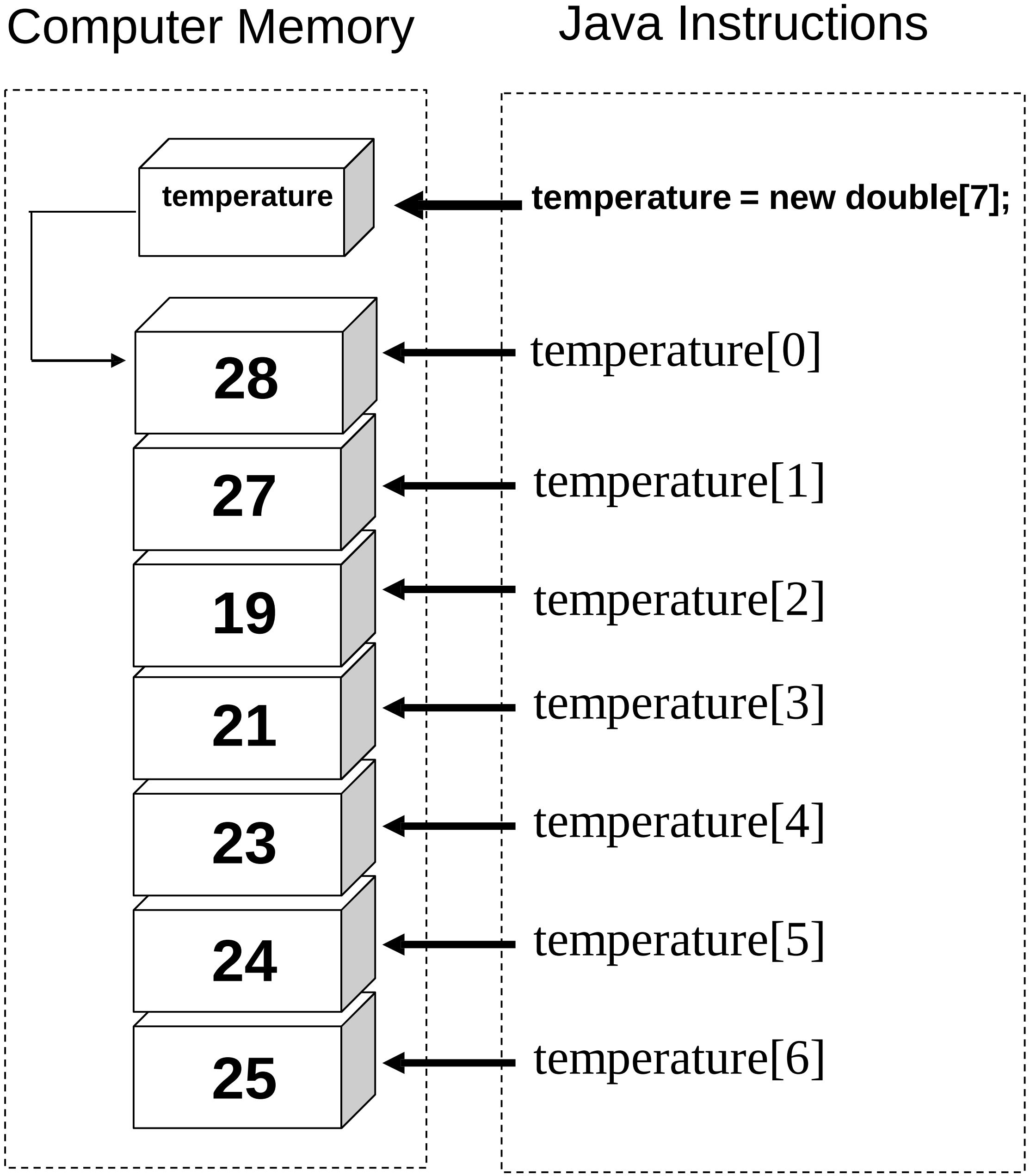
item of type 'double'

```
temperature = new double[7];
```

Example of values stored in array temperature

temperature[0] = 28
temperature[1] = 27
…..
temperature[6] = 25

Note:
array indices start from 0 and not from 1



Naming of array elements

Each element in an array shares ***the same name as the array.***

The individual elements are then ***uniquely identified*** by an additional **index value**.

In Java, ***array indices start from 0 and not from 1*****.**

This index value is always enclosed in square brackets **:**

first temperature :	**`temperature[0]`**
second temperature:	**`temperature[1]`**

Summary
What is an Array?

An array is a group of contiguous memory locations that all have

- the same **name** and
- the same **type**.

To refer to a particular location or element in the array, we specify:

- the **name of the array** and
- the **position of the particular element in the array**.

Array Subscript or Index value

The position number in square brackets is more formally called a **subscript or index value**, eg temperature[0] – 0 is known as the subscript.

A subscript must be an integer or an integer expression.

If a program uses an expression as a subscript,
then the expression is evaluated to determine the subscript.

If we assume that variable **a** is equal to 5 and variable **b** is equal to 6
then the statement

```
c[ a + b ] += 2;
```

adds 2 to array element c[11].

Initializing an array

- It is possible to initialise an array with values

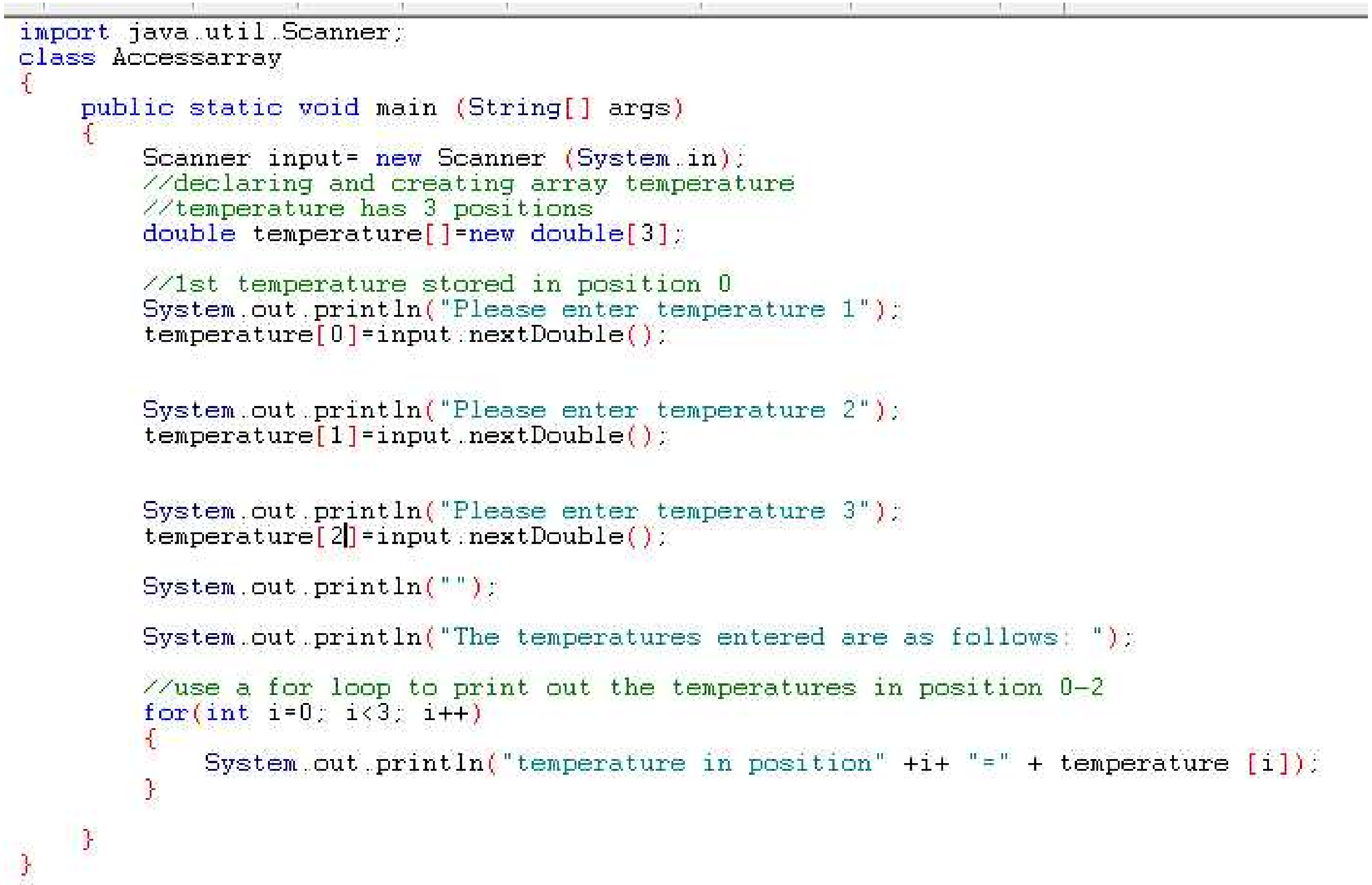
- The following example **initialises array temperature** with 7 values.

```
double temperature[] = {9, 11.5, 11, 8.5, 7, 9, 8.5};
```

=>	temperature[0] = 9	1st element in array
	temperature[1] = 11.5	2nd element in array
….		
	temperature[6] = 8.5	7th element in array

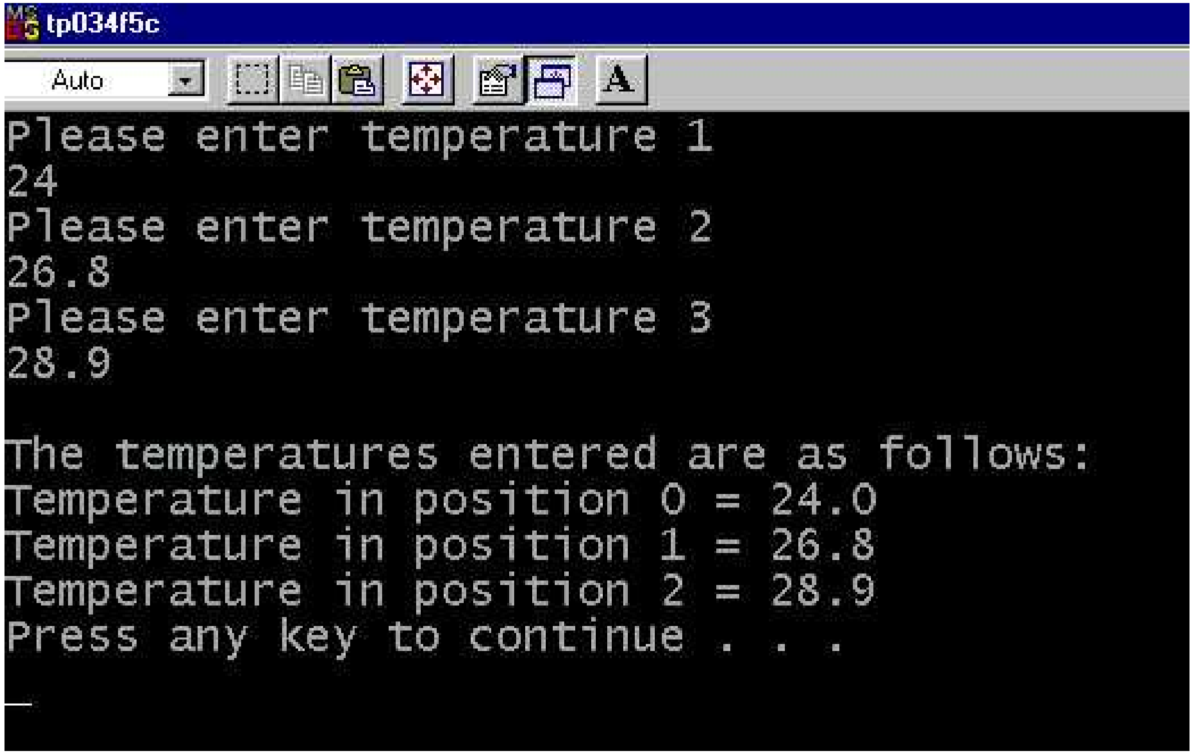
Accessing array elements

Program Example



```
import java.util.Scanner;
class Accessarray
{
    public static void main (String[] args)
    {
        Scanner input= new Scanner (System.in);
        //declaring and creating array temperature
        //temperature has 3 positions
        double temperature[]=new double[3];

        //1st temperature stored in position 0
        System.out.println("Please enter temperature 1");
        temperature[0]=input.nextDouble();


        System.out.println("Please enter temperature 2");
        temperature[1]=input.nextDouble();


        System.out.println("Please enter temperature 3");
        temperature[2]=input.nextDouble();

        System.out.println("");

        System.out.println("The temperatures entered are as follows: ");

        //use a for loop to print out the temperatures in position 0-2
        for(int i=0; i<3; i++)
        {
            System.out.println("temperature in position" +i+ "=" + temperature [i]);
        }

    }
}
```

Sample Test Run



```
Please enter temperature 1
24
Please enter temperature 2
26.8
Please enter temperature 3
28.9

The temperatures entered are as follows:
Temperature in position 0 = 24.0
Temperature in position 1 = 26.8
Temperature in position 2 = 28.9
Press any key to continue . . .
```

Accessing array elements

```
temperature[0] = input.nextDouble();

temperature[0] = temperature[0] * 2;

if (temperature[0] >= 18)
{
      System.out.println("it was hot today");
}
```

- In example above, if I entered 10 for temperature[0], temperature[0] * 2 => 20 is now stored in temperature[0].

Using a loop to enter values into an array

```
for(int i = 0; i < 7; i++)
{
  System.out.println("max temperature for day ”
                       +(i+1));
  temperature[i] = input.nextDouble();
}
```

- In the above example, the user is asked to **enter 7 temperatures** which will be stored in **array temperature**.

The 'length' attribute

```
double temperature[] = new double[7];

for (int i = 0; i < temperature.length, i++)
{
     // code for loop goes here
}
```

- The **length attribute** of an array holds the **number of elements** that can be placed in the array.
- In the above example temperature.length = 7

Program Example to enter and display temperatures

```
 class TemperatureReadings
 {
   public static void main(String[] args)
   {
      double temperature[] = new double[7];
      // code to enter temperatures

      // code to display temperatures

}
 }
```

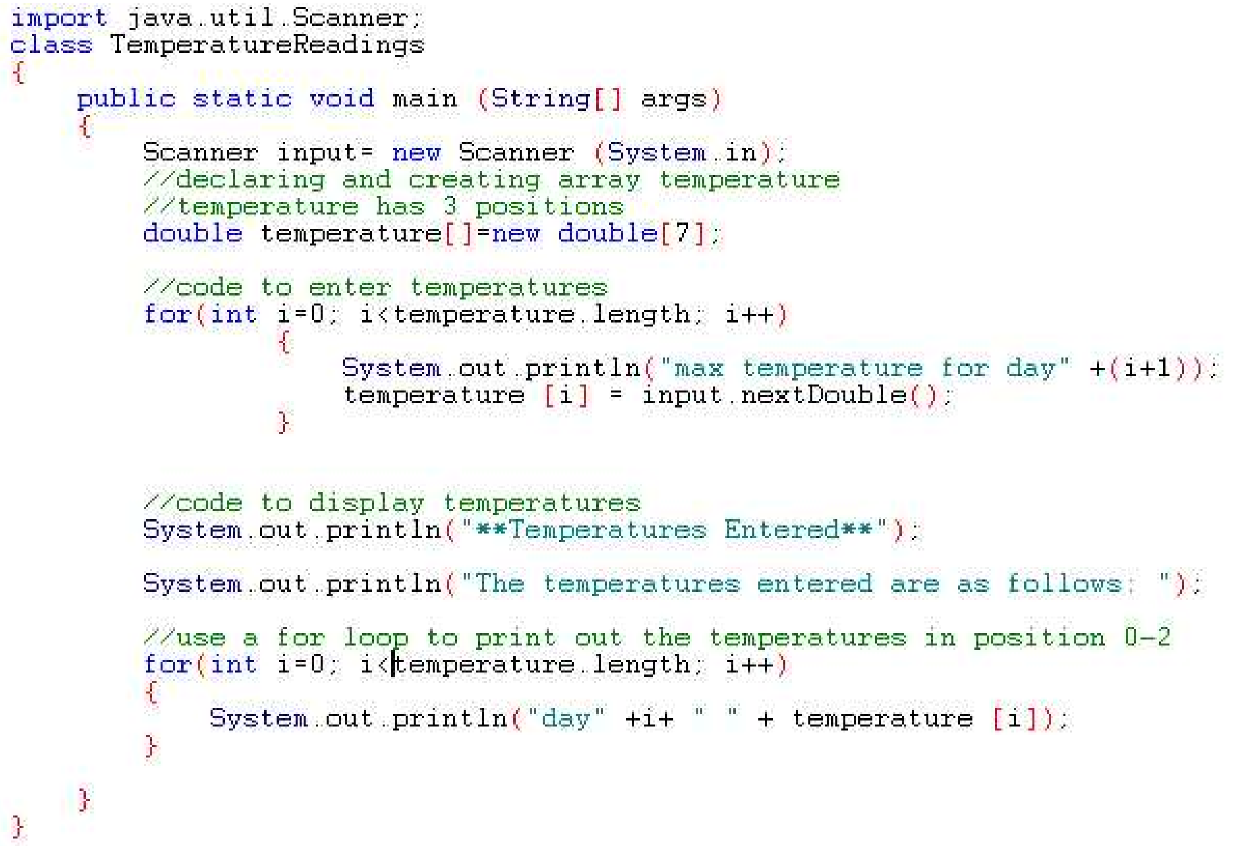
Code to enter temperatures

```
for (int i = 0; i < temperature.length; i++)
{
    System.out.println
      ("max temperature for day " + (i+1));
    temperature[i] = input.nextDouble();
}
```

Code to display temperatures

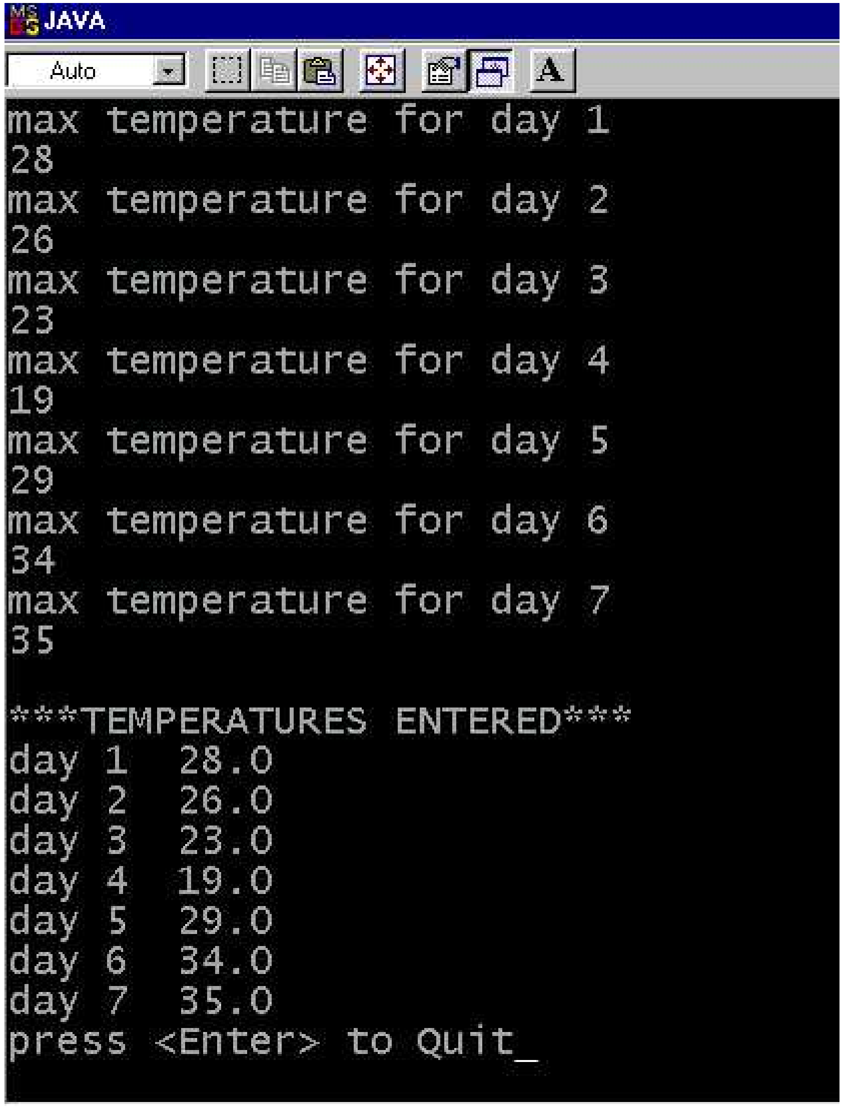
```
System.out.println();
System.out.println("***TEMPERATURES ENTERED***");
for (int i = 0; i < temperature.length; i++)
{
   System.out.println("day "+(i+1)
                       + "  ” +   temperature[i]);
}
```

Program Example to enter and display temperatures



```
import java.util.Scanner;
class TemperatureReadings
{
    public static void main (String[] args)
    {
        Scanner input= new Scanner (System.in);
        //declaring and creating array temperature
        //temperature has 3 positions
        double temperature[]=new double[7];

        //code to enter temperatures
        for(int i=0; i<temperature.length; i++)
                {
                    System.out.println("max temperature for day" +(i+1));
                    temperature [i] = input.nextDouble();
                }


        //code to display temperatures
        System.out.println("**Temperatures Entered**");

        System.out.println("The temperatures entered are as follows: ");

        //use a for loop to print out the temperatures in position 0-2
        for(int i=0; i<temperature.length; i++)
        {
            System.out.println("day" +i+ " " + temperature [i]);
        }

    }
}
```

Sample Test Run



```
max temperature for day 1
28
max temperature for day 2
26
max temperature for day 3
23
max temperature for day 4
19
max temperature for day 5
29
max temperature for day 6
34
max temperature for day 7
35

***TEMPERATURES ENTERED***
day 1  28.0
day 2  26.0
day 3  23.0
day 4  19.0
day 5  29.0
day 6  34.0
day 7  35.0
press <Enter> to Quit_
```

The Bounds of the Array

When a Java program is executed, the Java interpreter checks array element references to make sure they are valid, ie that they are not **outside the bounds of the array.**

All subscripts must be **> = 0** and **< length of the array**.

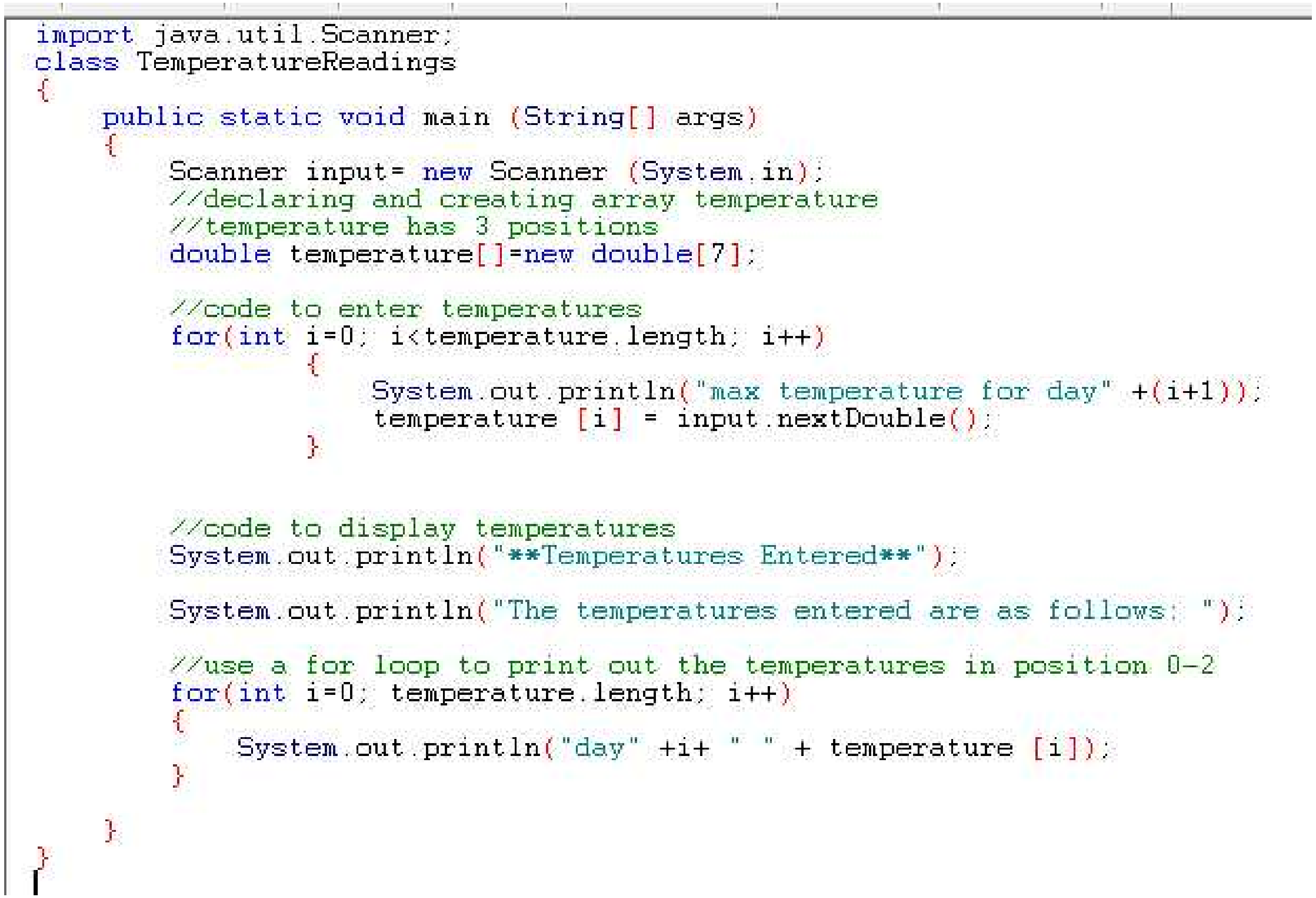
If there is an invalid reference Java generates an exception.

Exceptions are used to indicate that an error occurred in a prog

They enable the programmer to recover from an error and continue the execution of the program instead of abnormally terminating the program.

When an invalid array reference is made an **ArrayIndexOutOfBoundsException** is generated.

The Bounds of the Array



```
import java.util.Scanner;
class TemperatureReadings
{
    public static void main (String[] args)
    {
        Scanner input= new Scanner (System.in);
        //declaring and creating array temperature
        //temperature has 3 positions
        double temperature[]=new double[7];

        //code to enter temperatures
        for(int i=0; i<temperature.length; i++)
            {
                System.out.println("max temperature for day" +(i+1));
                temperature [i] = input.nextDouble();
            }


        //code to display temperatures
        System.out.println("**Temperatures Entered**");

        System.out.println("The temperatures entered are as follows: ");

        //use a for loop to print out the temperatures in position 0-2
        for(int i=0; temperature.length; i++)
        {
            System.out.println("day" +i+ " " + temperature [i]);
        }

    }
}
```

Returning to our previous example, Temperatures, if I tried to access temperature[7], I would get an **ArrayIndexOutOfBoundsException Error**

Exercises

Written Exercises

1. Consider the following explicit creation of an array:

```
int someArray = {2,5,1,9,11};
```

a). What would be the value of someArray.length?

b) What is the value of someArray[2]?

c) Create the equivalent array by assigning the **new** operator and then assigning the value of each element individually.

d). Write a loop that will double the value of every item in **someArray.**

e). Write a loop that will display all the items in the array greater than or equal to 5.

Written Exercises

2. Answer the following questions regarding an array called **fractions**

1. Define a constant variable ARRAY_SIZE initialised to 10.
2. Declare an array with ARRAY_SIZE elements of type float and initialise the elements to 0.
3. Name the fourth element from the beginning of the array.
4. Refer to array element four.
5. Assign the value 1.667 to array element 9.
6. Sum the value 3.333 to the seventh element of the array.
7. Sum all the elements of the array using a for repetition control structure. Define the integer variable **x** as a control variable for the **for** the loop.

Programs

1. Write an **application** that asks the user to enter 5 numbers, these numbers are stored in an array called **numbers.** The numbers are then printed to the screen.

   Save this program as **L7numbersarray.java**

Programs

2. Return to program TemperatureReadings from lecture as shown below:

Modify this program so that only temperatures > 18 are printed to the screen.
Save this program as **L7TemperatureReadings.java**

Programs

3. Using the data list given below to initialise an array, write a **program** to solve each of the following problems:

Data List = 2,4,6,9,5,4,5,7,12,15,21,32,45,5,6,7,12

- Compute the sum of the elements in the array.
- Compute the product of the elements in the array.
- Print only the multiples of 3 in the array.
- Compute the frequency of the number 5 in the array.

Save this program as **L7ArraySum.java**

Programs

4. Write a program that reads a list of 15 numbers and prints only those values that are greater than the last number on the list.

Save this program as **L7GreatCheck.java**