SENG1110/SENG6110
Object Oriented Programming

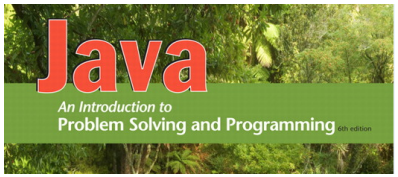
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

FACULTY OF ENGINEERING AND BUILT ENVIRONMENT

www.newcastle.edu.au

Lecture 8

Arrays – part I



Outline

- Previously…
  - Java basics/input/output
  - Conditional statements– if/switch
  - Loop statements – while/do-while/for
  - Classes and methods
- Now…
  - Array Basics
  - Arrays in Classes and Methods
    - Sales report example
    - Arrays as parameter/Return array
    - Compare arrays
  - More examples – arrays and methods
  - Multi dimensional array

Creating and Accessing Arrays

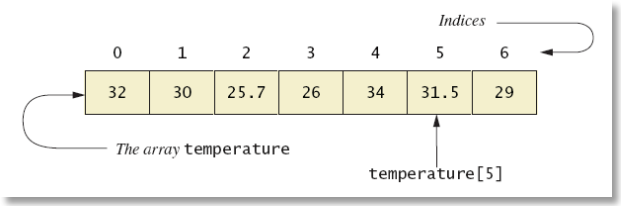
- An array is a special kind of object
- Think of as collection of variables of same type
- Creating an array with 7 variables of type double

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

- To access an element use
  - The name of the array
  - An index number enclosed in braces
- Array indices begin at zero

Creating and Accessing Arrays

- Figure 7.1 A common way to visualize an array



- Note CodeSamplesWeek8,
  `class ArrayOfTemperatures`

Creating and Accessing Arrays

```
Enter 7 temperatures:
32
30
25.7
26
34
31.5
29
The average temperature is 29.7428
The temperatures are
32.0 above average
30.0 above average
25.7 below average
26.0 below average
34.0 above average
31.5 above average
29.0 below average
Have a nice week.
```

Sample screen output

Square Brackets with Arrays

- With a data type when declaring an array

```
int [ ] pressure;
```

- To enclose an integer expression to declare the length of the array

```
pressure = new int [100];
```

- To name an indexed value of the array

```
pressure[3] = keyboard.nextInt();
```


Array Details

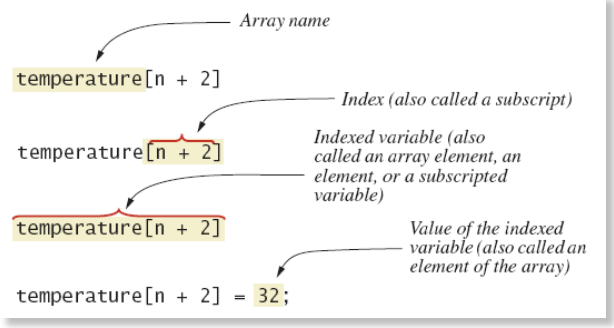
- Syntax for declaring an array with **new**

```
Base_Type[] Array_Name = new Base_Type[Length];
```

- The number of elements in an array is its length
- The type of the array elements is the array's base type

Array Details

- Figure 7.2 Array terminology



The Instance Variable `length`

- As an object an array has only one public instance variable
  - Variable `length`
  - Contains number of elements in the array
  - It is final, value cannot be changed

- Note CodeSamplesWeek8
  `class ArrayOfTemperatures2`

More About Array Indices

- Index of first array element is 0
- Last valid Index is `arrayName.length - 1`
- Array indices must be within bounds to be valid
  - When program tries to access outside bounds, run time error occurs

The Instance Variable `length`

```
How many temperatures do you have?
3
Enter 3 temperatures:
32
26.5
27
The average temperature is 28.5
The temperatures are
32.0 above average
26.5 below average
27.0 below average
Have a nice week.
```

Sample screen output

Initializing Arrays

- Possible to initialize at declaration time

```
double[] reading = {3.3, 15.8, 9.7};
```

- Also may use normal assignment statements
  - One at a time
  - In a loop

```
int[] count = new int[100];
for (int i = 0; i < 100; i++)
    count[i] = 0;
```


Arrays in Classes and Methods
Case Study: Sales Report

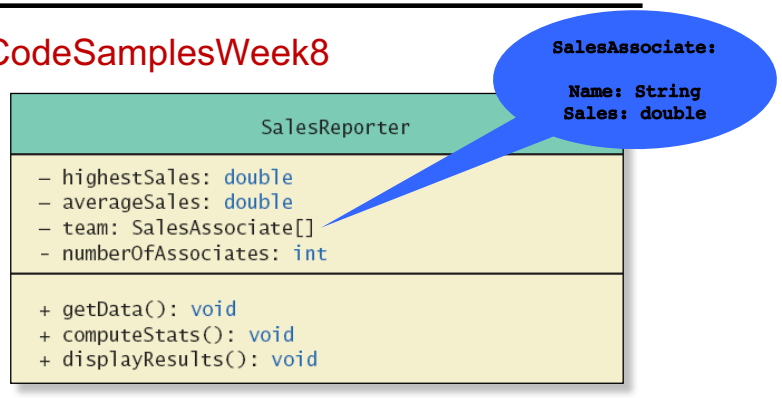
- Program to generate a sales report.
- It contains 2 classes:
  - **SalesAssociate** class will contain
    - Name
    - Sales figure
  - **SalesReporter** class will
    - Array of SalesAssociate
    - numberOfAssociates
    - highest sale
    - averageSale

Indexed Variables as Method Arguments

- Indexed variable of an array
  - Example … `a[i]`
  - Can be used anywhere variable of array base type can be used
- View CodeSamplesWeek8 using indexed variable as an argument,
  `class ArgumentDemo`

Case Study: Sales Report

- CodeSamplesWeek8



Note how the example does not use index 0.
Not a good practice.

Entire Arrays as Arguments

- Declaration of array parameter similar to how an array is declared
- Example:

```
public class SampleClass
{
    public static void incrementArrayBy2(double[] anArray)
    {
        for (int i = 0; i < anArray.length; i++)
            anArray[i] = anArray[i] + 2;
    }
    <The rest of the class definition goes here.>
}
```

Entire Arrays as Arguments

- Note – array parameter in a method heading does not specify the length
  - An array of any length can be passed to the method
  - Inside the method, elements of the array can be changed
- When you pass the entire array, do not use square brackets in the actual parameter

Array Assignment and Equality

- Arrays are objects
  - Assignment and equality operators behave (misbehave) as specified in previous chapter
- Variable for the array object contains memory address of the object
  - Assignment operator `=` copies this address
  - Equality operator `==` tests whether two arrays are stored in same place in memory

Arguments for Method main

- Recall heading of method `main`
  `public static void main (String[] args)`
- This declares an array
  - Formal parameter named `args`
  - Its base type is `String`
- Thus possible to pass to the run of a program multiple strings
  - These can then be used by the program

Array Assignment and Equality

- Two kinds of equality
- View CodeSamplesWeek8
  `class TestEquals`

Sample screen output

```
Not equal by ==.
Equal by the equals method.
```

Array Assignment and Equality

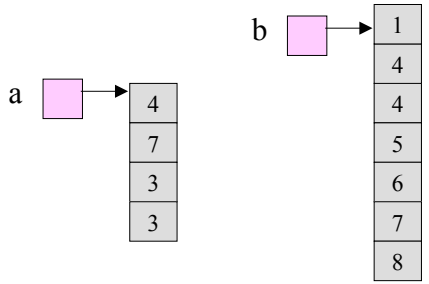
- Note results of **==**
- Note definition and use of method **equals**
  - Receives two array parameters
  - Checks length and each individual pair of array elements
- Remember array types are reference types

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

More about arrays and methods

- Suppose we have two arrays and suppose we receive some input

```
...
int[] a = new int[4];
int[] b = new int[7];
...
```



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Methods that Return Arrays

- A Java method may return an array
- View CodeSamplesWeek8
  class ReturnArrayDemo
- Note definition of return type as an array
- To return the array value
  - Declare a local array
  - Use that identifier in the **return** statement

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

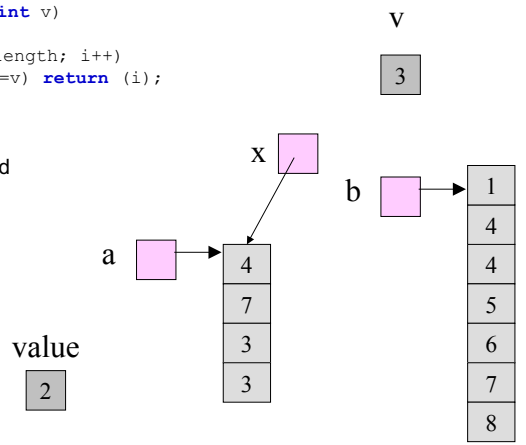
More about arrays and methods

- Suppose the method below (array is parameter)

```
public int meth1(int[] x, int v)
{
        for(int i=0; i<x.length; i++)
                if (x[i]==v) return (i);
        return(-1);
}
```

- Suppose we call the method

```
value = meth1(a,3)
value = meth1(b,value*2+1)
```



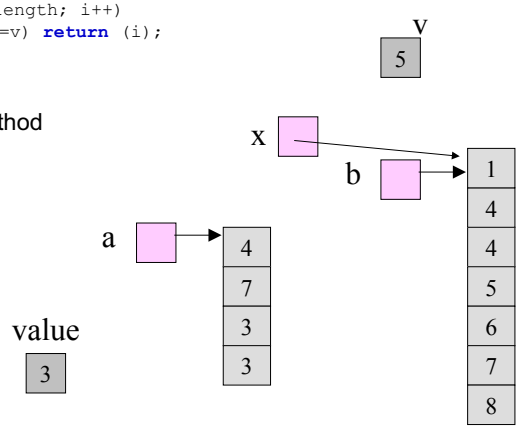
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

More about arrays and methods


```
public int meth1(int[] x, int v)
{
        for(int i=0; i<x.length; i++)
                if (x[i]==v) return (i);
        return(-1);
}
```

- Suppose we call the method

```
value = meth1(a,3)
value = meth1(b,value*2+1)
```

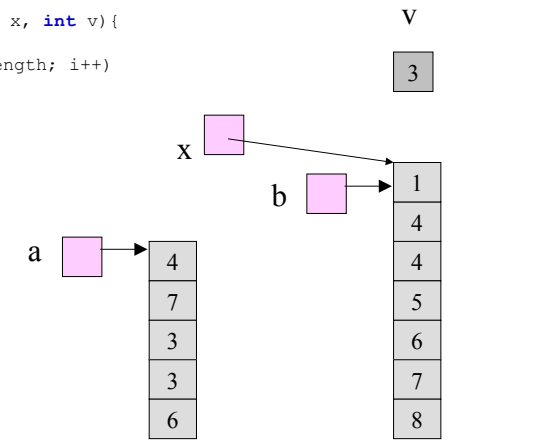


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

More about arrays and methods

- Now, suppose the method below (array is parameter and return)

```
public int[] meth1(int[] x, int v){
{
      for(int i=0; i<x.length; i++)
              x[i]+=v;
      return(x);
}
...
a = meth1(b,3)
```

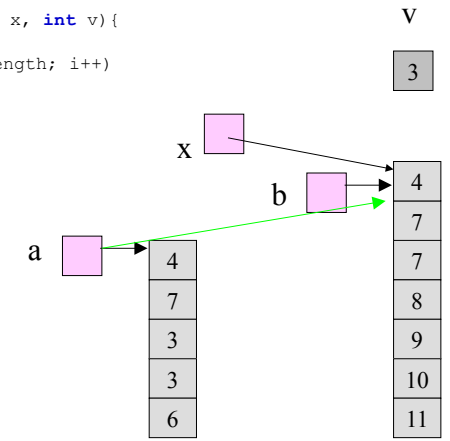


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

More about arrays and methods

- Now, suppose the method below (array is parameter and return)

```
public int[] meth1(int[] x, int v){
{
      for(int i=0; i<x.length; i++)
              x[i]+=v;
      return(x);
}
...
a = meth1(b,3)
```



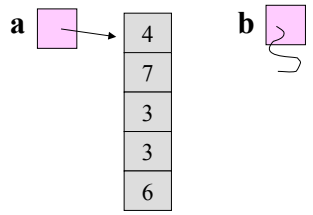
a and **b** will be reference to the same array

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying an array

- Suppose we have the arrays:

```
...
int[] a = new int[4];
int[] b;
...
```

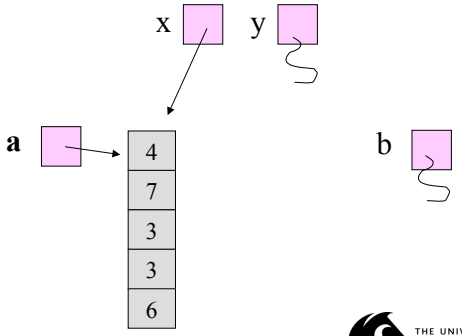


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying an array – copy1


```
public void copy1(int[] x, int[] y)
    {
          y = new int[x.length];
          for(int i=0; i< x.length; i++)
                    y[i] = x[i];
    }

...
copy(a,b)
```

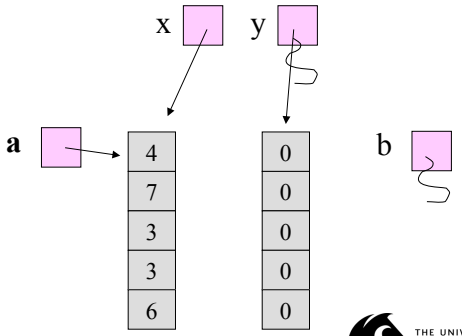


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying an array – copy1


```
public void copy1(int[] x, int[] y)
    {
          y = new int[x.length];
          for(int i=0; i< x.length; i++)
                    y[i] = x[i];
    }

...
copy(a,b)
```



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

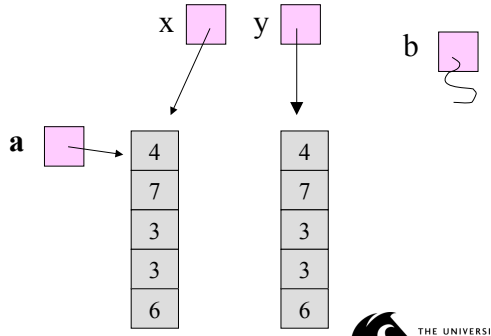
Copying an array – copy1


```
public void copy1(int[] x, int[] y)
{
      y = new int[x.length];
      for(int i=0; i< x.length; i++)
                y[i] = x[i];
}

copy(a,b)
```

It will not work

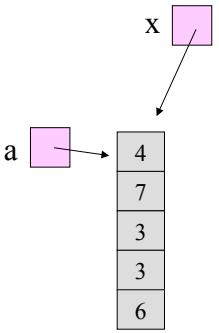
WHY?



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying an array – copy2

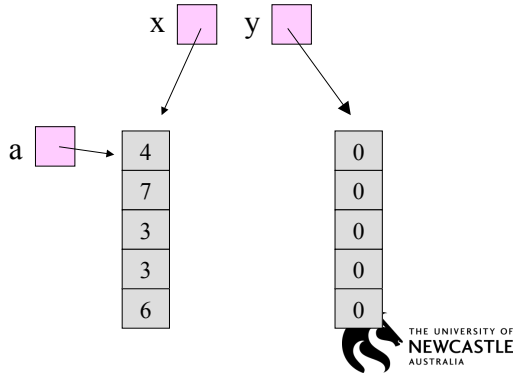

```
public int[] copy2(int[] x)
{
      int[] y = new int[x.length];
      for(int i=0; i< x.length; i++)
                y[i] = x[i];
      return(y);
}
...
b = copy2(a)
```



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying an array – copy2

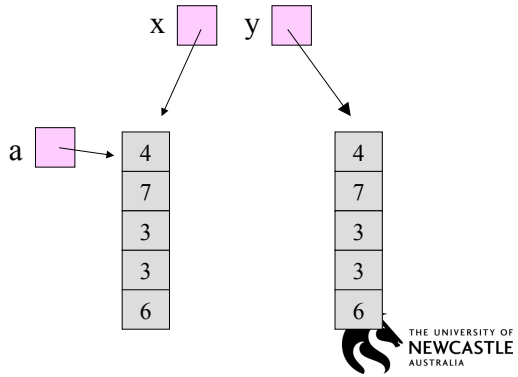

```
public int[] copy2(int[] x)
{
      int[] y = new int[x.length];
      for(int i=0; i< x.length; i++)
            y[i] = x[i];
      return(y);
}
...
b = copy2(a)
```



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying an array – copy2


```
public int[] copy2(int[] x)
{
      int[] y = new int[x.length];
      for(int i=0; i< x.length; i++)
            y[i] = x[i];
      return(y);
}
...
b = copy2(a)
```



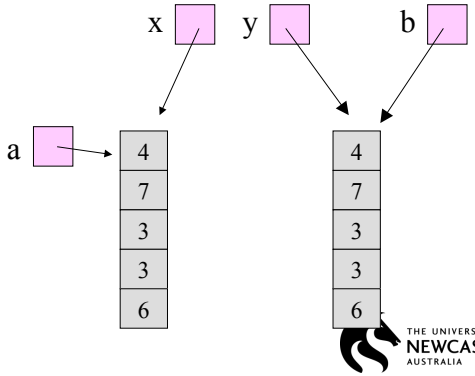
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying an array – copy2


```
public int[] copy2(int[] x)
{
      int[] y = new int[x.length];
      for(int i=0; i< x.length; i++)
            y[i] = x[i];
      return(y);
}
...
b = copy2(a)
```

a and b are reference to 2 different arrays

b is a copy of a

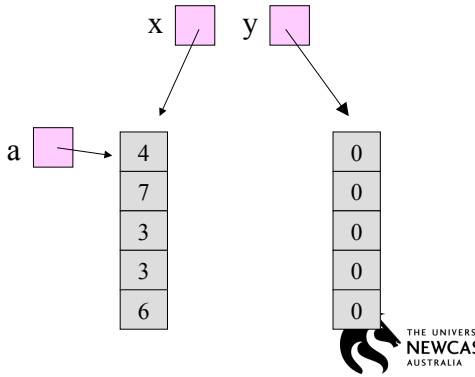


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying part of an array – copyPart

- Suppose we want to copy only some elements from an array. Example, we want to copy only elements that are greater than a number p.

```
public int[] copyPart(int[] x, int p)
{
      int[] y = new int[x.length];
      for(int i=0; i< x.length; i++)
            if (x[i]>p) y[i] = x[i];
      return(y);
}
...
b = copyPart(a,3)
```

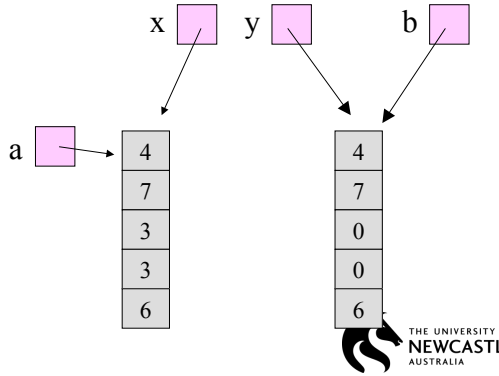


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying part of an array – copyPart

- Suppose we want to copy only some elements from an array. Example, we want to copy only elements that are greater than a number p.

```
public int[] copyPart(int[] x, int p)
{
        int[] y = new int[x.length];
        for(int i=0; i< x.length; i++)
                if (x[i]>p) y[i] = x[i];
        return(y);
}
...
b = copyPart(a,3)
```



What is the problem?

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Multidimensional-Array Basics

- Consider Figure 7.6, a table of values

Savings Account Balances for Various Interest Rates Compounded Annually (Rounded to Whole Dollar Amounts)

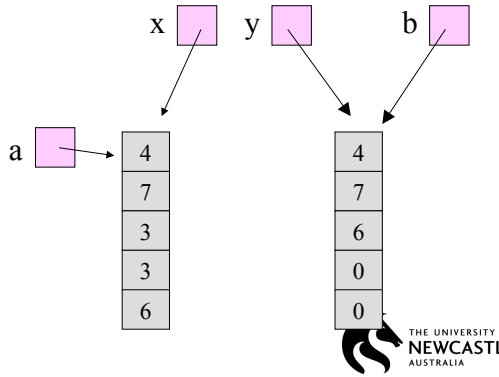
Year	5.00%	5.50%	6.00%	6.50%	7.00%	7.50%
1	$1050	$1055	$1060	$1065	$1070	$1075
2	$1103	$1113	$1124	$1134	$1145	$1156
3	$1158	$1174	$1191	$1208	$1225	$1242
4	$1216	$1239	$1262	$1286	$1311	$1335
5	$1276	$1307	$1338	$1370	$1403	$1436
6	$1340	$1379	$1419	$1459	$1501	$1543
7	$1407	$1455	$1504	$1554	$1606	$1659
8	$1477	$1535	$1594	$1655	$1718	$1783
9	$1551	$1619	$1689	$1763	$1838	$1917
10	$1629	$1708	$1791	$1877	$1967	$2061

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Copying part of an array – copyPart

- Suppose we want to copy only some elements from an array. Example, we want to copy only elements that are greater than a number p.

```
public int[] copyPart(int[] x, int p)
{
        int[] y = new int[x.length];
        for(int i=0,j=0; i< x.length; i++)
                if (x[i]>p) y[j++] = x[i];
        return(y);
}
...
b = copyPart(a,3)
```



Is it ok now?

The logical size of the array b **Think about that!**

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Multidimensional-Array Basics

- Figure 7.7 Row and column indices for an array named `table`

`table[3][2]` has a value of 1262

Row index 3

Column index 2

Indices	0	1	2	3	4	5
0	$1050	$1055	$1060	$1065	$1070	$1075
1	$1103	$1113	$1124	$1134	$1145	$1156
2	$1158	$1174	$1191	$1208	$1225	$1242
3	$1216	$1239	$1262	$1286	$1311	$1335
4	$1276	$1307	$1338	$1370	$1403	$1436
5	$1340	$1379	$1419	$1459	$1501	$1543
6	$1407	$1455	$1504	$1554	$1606	$1659
7	$1477	$1535	$1594	$1655	$1718	$1783
8	$1551	$1619	$1689	$1763	$1838	$1917
9	$1629	$1708	$1791	$1877	$1967	$2061

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Multidimensional-Array Basics

- We can access elements of the table with a nested for loop
- Example:

```
for (int row = 0; row < 10; row++)
   for (int column = 0; column < 6; column++)
      table[row][column] =
          balance(1000.00, row + 1, (5 + 0.5 * column));
```

- View CodeSamplesWeek8
  class InterestTable

Multidimensional-Array Parameters and Returned Values

- Methods can have
  - Parameters that are multidimensional-arrays
  - Return values that are multidimensional-arrays
- View CodeSamplesWeek8
  class InterestTable2

Multidimensional-Array Basics

```
Balances for Various Interest Rates Compounded Annually
(Rounded to Whole Dollar Amounts)

Years  5.00%  5.50%  6.00%  6.50%  7.00%  7.50%
1      $1050  $1055  $1060  $1065  $1070  $1075
2      $1103  $1113  $1124  $1134  $1145  $1156
3      $1158  $1174  $1191  $1208  $1225  $1242
4      $1216  $1239  $1262  $1286  $1311  $1335
5      $1276  $1307  $1338  $1370  $1403  $1436
6      $1340  $1379  $1419  $1459  $1501  $1543
7      $1407  $1455  $1504  $1554  $1606  $1659
8      $1477  $1535  $1594  $1655  $1718  $1783
9      $1551  $1619  $1689  $1763  $1838  $1917
10      $1629  $1708  $1791  $1877  $1967  $2061
```

Sample screen output

Java's Representation of Multidimensional Arrays

- Multidimensional array represented as several one-dimensional arrays
- Given
  int [][] table = new int [10][6];
- Array table is actually 1 dimensional of type **int[]**
  - It is an array of arrays
- Important when sequencing through multidimensional array

Ragged Arrays

- Not necessary for all rows to be of the same length
- Example:

```
int[][] b;
b = new int[3][];
b[0] = new int[5]; //First row,  5 elements
b[1] = new int[7]; //Second row, 7 elements
b[2] = new int[4]; //Third row,  4 elements
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Programming Example

```
Employee  1   2   3   Totals
Monday    8   0   9   17
Tuesday   8   0   9   17
Wednesday 8   8   8   24
Thursday  8   8   4   20
Friday    8   8   8   24
Total  =  40  24  38
```

Sample screen output

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

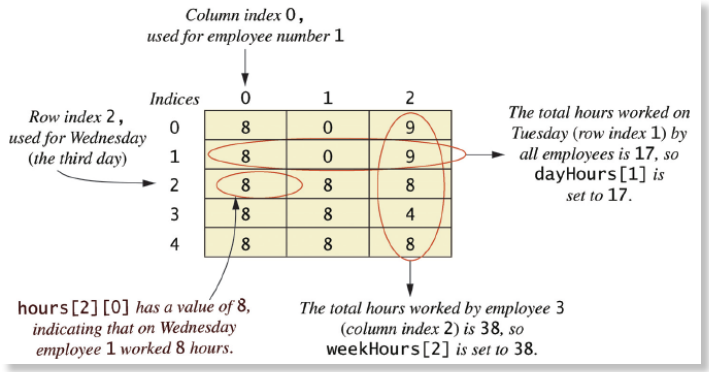
Programming Example

- Employee Time Records
  - Two-dimensional array stores hours worked
    - For each employee
    - For each of 5 days of work week
  - Array is private instance variable of class
- View CodeSamplesWeek8
  `class TimeBook`

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Programming Example

- Figure 7.8 Arrays for the class `TimeBook`



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Your task

- Read
  - Chapter 7 of the text book

- Exercises
  - MyProgrammingLab
  - Implement/compile/run the examples from lecture slides (copy from codeSamplesWeek8 – available in Blackboard)
  - Complete all the lab exercises
  - There are extra examples in chapter 7.