JAVA EXERCISES

Laboratory Exercise: 2

Learning Outcomes

After completing these exercises, you should be able to ...

- initialise, declare and iterate over arrays
- specify the classpath when compiling and executing applications
- access command line arguments
- write a basic class

...and understand what is meant by ...

- a nested loop

Source Code

The source code for this worksheet is available on Learning Central at:

CM1210 → Learning Materials → Exercises

Arrays and Strings

1. Two ways to declare and initialise an array are...

```
int[] nums = {3,5,7,9,10};
```

...and...

```
int[] nums = new int[5];
nums[0] = 3;
nums[1] = 5;
nums[2] = 7;
nums[3] = 9;
nums[4] = 10;
```

Write a program that creates an array containing the above values and then uses a for loop to calculate and print out their sum.
NOTE: When writing a loop that will iterate over the elements of an array use the expression **myArray.length** to get the number of elements in the array.

2. Download the source file **CountNames.java**. The file declares and initialises an array containing the list of actors' names.

(a) Modify the application so that it counts the number of actors whose names end in **Jones**. **HINT:** Use the Java API Documentation to investigate the **endsWith** method in the **String** class.

(b) One of the names has been written in UPPER CASE and might not be counted in your answer to 7a. Modify your application so that it is able to count all instances of **Jones** regardless of whether they're written in upper case, lower case or mixed case.

Command line arguments

3. Every Java application includes a method:

```
public static void main( String[] args )
```

The parameter **args** is an array of **String** objects containing the *command line arguments* passed to the method/application by the Operating System. For example, given a Java application **Foo**, executing the following at the command line:

```
% java Foo bar1 bar2 bar3
```

will pass the arguments **bar1**, **bar2** and **bar3** to the **main** method as an array of three elements names **args**.

Write an application **ShowArgs** that outputs (to standard outptut):

(a) The number of command line arguments provided.

(b) The contents of the arrays **args** (use an appropriate **for** loop).

HINT: To test the application you will have to pass it some command line arguments.

What happens when you surround a few command line arguments in quotation marks? Try executing the following:

```
% java ShowArgs "bar 1bar2" bar3
% java ShowArgs "bar 1bar2 bar3"
```

4. Arrays in Java support the *for-each* loop for iterating over elements without using an explicit index (see slides from session 1). The following is an example of a *for-each* loop that prints out each element of an array of **Strings**:

```
String[] arr = {"apple", "pear", "orange"};
for( String fruit : arr ) {
   System.out.println( fruit );
}
```

Update your program from Q2 to use a *for-each* loop to print each command line argument.

Writing classes

5. Download the file **BankAccount.java**. Complete the declarations of the data fields (instance variables) by adding the appropriate types. Add:

(a) A constructor that takes two arguments specifying the holder's name and account number

(b) Appropriate *accessor* and *mutator* methods (also called *setters* and *getters*). For example, for the **name** field you should create an accessor method called **getName()** and a mutator method called **setName(String newName)**.

(c) Methods to deposit to, withdraw from, and check the balance of the account.

Test the class by writing a separate application **BankTest** that creates an account with the name *M J W Morgan* and number *0012067* and calls each method (output the state of the object to standard output after each call - hint: override toString() method).

6. Modify your **BankTest** application so that the name and bank account number are taken from two command line arguments. For example:

```
% java BankTest "W White" 0033548
```

should create an account with name *W White* and number *0033548*.

7. Modify your answer to Q5 so that the application takes an **optional** third argument specifying the initial balance of the account. For example:

```
% java BankTest "W White" 0033548 300.00
```

will create an account with an initial deposit of £300.00.