**OOP2. Problem Set 5B: arrays of objects and the Bicycle system. Corresponds to**

**Section 5, Units 14 and 15.**

1. Copy the sample programs WelcomeWithArgs1 and WelcomeWithArgs2 into your own

folder, and compile – but don’t run for the moment – both. Now open a command

window, change to your own drive, and change to your current working folder within that

drive. You are now ready to run both programs from

the command-line, by typing java MyProgram firstArgument secondArgument

<enter>, where firstArgument should be your name, and secondArgument should be your

favourite place for relaxing. If you want either argument to include more than one word,

put the combined words in quotation marks, as in “at home”.

2. Write a program which runs from the command line and generates a letter inviting a

person to attend for interview in a specific place on a specific day at a specific time.

Your program should take in the person’s name, the place, the day and the time as 4

command-line arguments. The letter should be displayed in a dialog.

3. Write a program which sets up the names of the days of the week as an array of

Strings as follows: String [] days = {“Monday”, “Tuesday”, ……….}; The

program should allow the user to enter a day number between 1 and 7, and display the

corresponding day name in a dialog. The day number should be used as an index into the

days array (see the slide titled Arrays of String labels, slide 40, in your notes for

section 5). You do not need a switch or if-else construction in this program. When you

have it working, add a loop controlled by sentinel so that it repeats until you enter a day

number of 0.

4. Declare the array of day names as for the previous exercise, then use the enhanced

for loop to display the 7 day names as a list within a message dialog. See your lecture

notes slides 42 and 43 for examples of how to do this.

5. Copy the 4 versions of FriendsArray.java and look at them. Make sure you have a copy

of Person in the same folder. In FriendsArray1,

(i) What is the name of the array?

(ii) How many Persons will it hold?

(iii) Add a block of code which will add up the ages of all the persons in your

array, and calculate and output their average age.

(iv) Add a single line of code to FriendsArray2 which will display the name of

the second person you entered.

(v) FriendsArray2 has code which finds the youngest and oldest persons.

Change this to find and display the person with the longest name.

(vi) FriendsArray2 has code which searches for a person by name. Alter this so

that, instead, it searches for the first person with a particular age.

(vii) FriendsArray3 works a bit differently: it doesn’t fill the array, but instead

keeps track of the number of persons you have added. Notice that the output

contains some ‘null’ entries. Modify the output loop – you will have to use a

regular ‘for’ instead of the enhanced version – so that the null entries are not

displayed. Add a block which searches for a particular person by name,

similar to the block in FriendsArray2.  
(viii) FriendsArray4 contains a series of methods to process an array. Add a method to find and display the oldest person, and code within main() to test it.

(ix) (Advanced and optional): add a block of code to FriendsArray3 which

allows you to delete a particular person. For this, first the program must

allow the user to enter a name, then it must find that entry, then it must move

all the items further along the array up one slot, then set the last entry to null.

(x) (Advanced and optional): look ahead to the notes for unit 18, then rewrite

your solution to the previous part so that it uses an ArrayList instead. You

should find it much easier.

6. Copy the Bicycle class and BicycleFrame0, BicycleFrame1 and BicycleFrame2 into your

own folder. Open them in JCreator and make sure you have ‘class view’ selected. Draw

a UML class diagram for Bicycle. How many attributes does a Bicycle have? How

many methods? Draw a VOPC diagram for BicycleFrame2.

7. In BicycleFrame0:

(i) how many menus appear on the menu bar?

(ii) How many menu items on each one, and where are they created?

(iii) What is the purpose of the showMessage() method?

(iv) Why are there two versions of it?

(v) How many other methods are there?

8. In BicycleFrame1:

(i) What identifier is being used for the array of Bicycles?

(ii) What is the maximum number of bicycles that can be stored in it?

(iii) Where is it being declared?

(iv) Where is it being created?

9. In BicycleFrame2:

(i) Find the method which adds a bicycle to the system. It doesn’t let you set the make or the

model of the bicycle. Add code to do this.

(ii) What variable stores the number of valid bicycles in the system? Where is it declared?

Where is it initialized?

(iii) Find the method which displays the details of the bicycles. The display isn’t well laid

out. Set the font for the text area to a monospaced font like Courier, and use ‘get’ methods

and String.format() to get a better layout.

(iv) Could you make the code for this method shorter by using the enhanced for loop?