

|  |
| --- |
| **COURSEWORK ASSESSMENT SPECIFICATION** |

|  |  |
| --- | --- |
| **Module Title:** | *Programming 1* |
| **Module Number:** | *KV4000* |
| **Module Tutor Name(s):** | *Alan Maughan* |
| **Academic Year:** | *2018/19* |
| **% Weighting (to overall module):** | *30%* |
| **Coursework Title:** | *Homework 5* |
| **Average Study Time Required by Student:** | *8 hours* |

**Dates and Mechanisms for Assessment Submission and Feedback**

|  |
| --- |
| **Date of Handout to Students:**  Week 10 |
| **Mechanism for Handout to Students:**  *via elp* |
| **Date and Time of Submission by Student:**  During Week 12 Lab Class |
| **Mechanism for Submission of Work by Student:**  Papers collected in lab |
| **Date by which Work, Feedback and Marks will be returned to Students:**  Marks & feedback will be given as the assessment is marked in the lab week 12. |
| **Mechanism for return of assignment work, feedback and marks to students:**  Marks & feedback will be given as the assessment is marked in the lab |

**Further Information**

*(Please ensure the assessment specification includes the following items)*

**Learning Outcomes tested in this assessment (from the Module Descriptor):**

1. Design a program from a specification;
2. Formulate solutions to a number of basic programming problems using an appropriate design notation;
3. Make effective use of basic data types arrays and structured programming control constructs: sequence, selection and iteration.
4. Understand and make basic use of functions/procedures.

**Assessment Criteria/Mark Scheme:**

One mark for each of the 12 tests.

**Nature of the submission required:**

Paper copies of source code & simple test plan. Code execution in lab.

**Instructions to students:**

*This is an individual piece of work.*

**Referencing Style:**

*N/A*

**Expected size of the submission**:

Under 10 pages – printed copies of source code

**Academic Conduct:**

You must adhere to the university regulations on academic conduct. Formal inquiry proceedings will be instigated if there is any suspicion of misconduct or plagiarism in your work. Refer to the University’s regulations on assessment if you are unclear as to the meaning of these terms. The latest copy is available on the university website.

# Homework 5

This work is due for marking **at the start of your lab** in week 12. It counts for 30% of the overall module mark.

You must bring with you a printed copy of your source code (the .java files) for the classes **LoyaltyCardArrayClass** and for **HW5Test** (see below). These should be produced before you come to the lab. Do not come to the lab and attempt to print copies then. This will be collected by the tutor when they mark your homework – make sure that they have your name / id on it. These will be retained for audit and internal moderation.

If these files are not submitted (and printed before the lab) then you will score 0 (zero)!

You must work on the program on your own, outside any formal classes and it must be ready to execute at the start of the scheduled laboratory class. All code must be completed using the BlueJ IDE. Any work utilising other IDEs will score zero.

All code must:

* Have the class header and all methods commented to ‘Javadoc’ standards using @author, @version, @param and @return tags as appropriate.
* Be coded to required layout (e.g. indentation) and naming standards.

Notes on the above were supplied in week 1 and there are numerous examples in the code you will have seen. Failure to meet these standards will result in loss of marks.

You may be asked questions about your program to confirm your understanding and that it is your own work. Failure to answer the questions may result in a deduction or total loss of marks.

***The work must be wholly your own. Collusion counts as academic misconduct and will be punished according to the University’s regulations detailed in “Assessment Regulations for Northumbria Awards” (ARTA) a copy of which is available on the University website.***

## The Task

Create a project which should be called “HW5”.

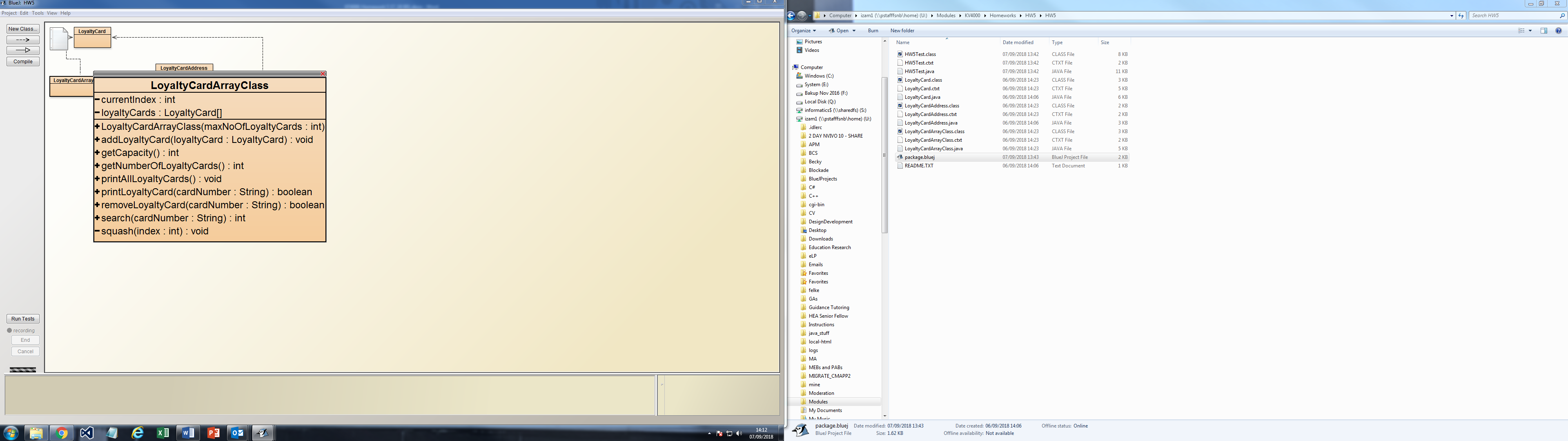
For this homework you are expected to write a program based upon the **LoyaltyCard** and **LoyaltyCardAddress** classes supplied on Blackboard. **NOTE:** These may be different to those used before. It is your responsibility to understand the code so that it can be correctly used in your class **LoyaltyCardArrayClass**.

You should not change the **LoyaltyCard** or **LoyaltyCard Address** classes. Your homework should work with the two classes as currently supplied

You are expected to write a class **LoyaltyCardArrayClass** which represents a collection of loyalty cards utilising a simple array.

**Task 1**

You are expected to write the **LoyaltyCardArrayClass** class as per the following class diagram:



Notes

|  |  |
| --- | --- |
| LoyaltyCardArrayClass | The constructor should set the size of the array at runtime and initialise the variable currentIndex. |
| addLoyaltyCard | Such check that there is space for a new loyalty card to be added to the array and, if so, add it utilising the LoyaltyCard class. Otherwise it should produce an error message “Membership full. Could not add LoyaltyCard!" |
| getCapacity / getNumberOfLoyaltyCard | These should return the size of the array and the number of loyalty cards. |
| printLoyaltyCard | This should print a single loyalty card given a card number utilising a method of the LoyaltyCard class. If this is invalid it should output "Loyalty card with card number: [cardNumber] not found." |
| printAllLoyaltyCards | This should print all loyalty cards utilising a method of the LoyaltyCard class. |
| removeLoyaltyCard | This should remove a loyalty card given a valid card number utilising a method of the LoyaltyCardclass and reoganise the remaining elements of the array. |
| search | Should search for a library card with the given card number and return the index utilising a method of the LoyaltyCardclass. If invalid it should return -1. |

**Testing and Marking**

You are given the class **HW5Test** which you should add to your project. You should put your name in the @author tag. This class will be executed to assess and mark your homework. The tutor will run this at your desk in the lab.

**You are not allowed to change this class in any way and if it is found that you have done so you will score zero!**

You should familiarise yourself with the code and use it to test your program. It has a method **runTests()** which calls methods to test your code. It is suggested that you implement your methods one method at a time. To do this you may wish to comment out the other method calls in runTests() until you have completed your coding / testing. **Do not forget to remove the commenting!** If you cannot get a particular method to work, move on to another. Please note that the code is designed to test valid and invalid situations – both score marks. Please ensure that, if you are having difficulty, that your classes compile and that you have implemented some methods.

## Marking Scheme: 12marks / 30% of module total

See HW5Test for marking scheme.