

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

|  |  |
| --- | --- |
| **Module Title:** | *Programming 2* |
| **Module Number:** | *KV4001* |
| **Module Tutor Name(s):** | *Alan Maughan* |
| **Academic Year:** | *2018/19* |
| **% Weighting (to overall module):** | *7.5%* |
| **Coursework Title:** | *Homework 4* |
| **Average Study Time Required by Student:** | *5 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 Lab 11 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 11. |
| **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. Demonstrate understanding of the principles underlying the object-oriented paradigm to design and provide solutions to problems.
2. Identify and apply predefined classes in the development of object-oriented programs

**Nature of the submission required:**

Paper copies of source code. Code execution in lab.

**Instructions to students:**

*This is an individual piece of work.*

**Referencing Style:**

*N/A*

**Expected size of the submission**:

Under 12 pages – mostly 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.

This work is due at the start of your lab in week 11. It counts for 7.5% of the overall module mark.

You must bring with you a printed copy of your source code (the .java fileS). These should be produced before you come to the lab. Do not come to the lab and attempt to print copies then. The tutor will collect these when they mark your homework – make sure that they have your name / id on them. 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.

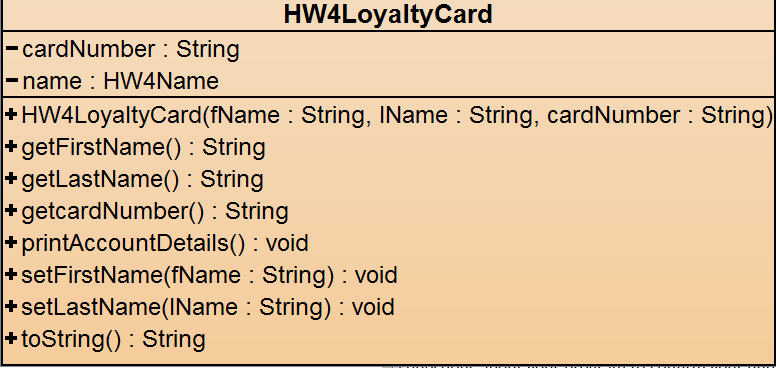
All code must:

* Have the class header and all methods commented to ‘Javadoc’ standards using @author, @version, @param and @return tags as appropriate. Check one of the classes supplied if you are not certain how these are used.
* Be coded to required layout (e.g. indentation) and naming standards.

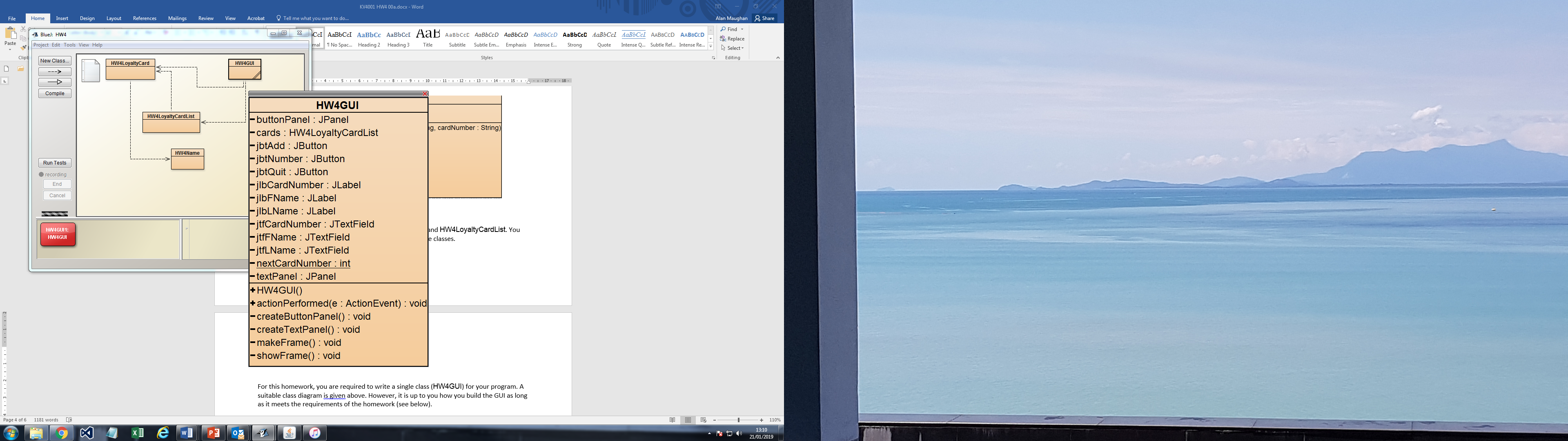
Failure to meet these standards will result in loss of a mark.

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.

This homework is intended to test your understanding of GUIs and is based upon a simplified loyalty card class.

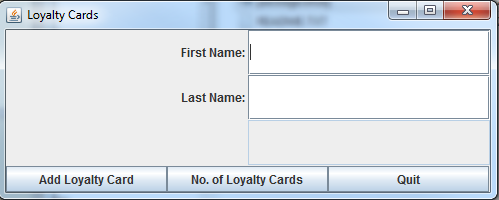


You are given the classes HW4Name, HW4LoyaltyCard and HW4LoyaltyCardList. You should firstly obtain and check your understanding of these classes.



For this homework, you are required to write a single class (HW4GUI) for your program. A suitable class diagram is given above. However, it is up to you how you build the GUI as long as it meets the requirements of the homework (see below).

When the program opens, it should produce the following GUI of size 500 x 200 pixels and should be centred in the middle of the screen:



The frame should allow the entry of a new cardholder’s first and last names and generate a card number number for that member. There should be three text fields to hold first name, last name and the card number. The third text field should not be editable, nor should the window be resizable. The labels should be right aligned. The third label should begin blank and will be amended as required by the program.

The ‘X’ at the top right-hand corner should not allow the window to close or the program to close.

**Add Loyalty Card**

Assuming that there is text in both name text fields – the button should:

* Set the text of the third JLabel to “Loyalty Card Number:”
* Generate a card number and display it in the loyalty card number text field. The first number generated should be ‘10001’ and each subsequent number should be one greater (i.e. the second card number should be 10002, the third 10003, etc.). You should use a static variable to help with this process.
* Change the text of the ‘Add Loyalty Card’ button to ‘Confirm’.
* Disable the ‘No. of Loyalty Cards’ button

**NOTE 1:**

If there is no entry in:

* either name text field, clicking ‘Add Loyalty Card’ should place the text “No names” in the loyalty card number text field;
* the first name text field only clicking ‘Add Loyalty Card’ should place the text “No first name” in the loyalty card number text field;
* the last name text field only clicking ‘Add Loyalty Card’ should place the text “No last name” in the loyalty card number text field;

In each of these cases, the GUI should wait for the user to add the missing data and, if it is now complete, clicking the ‘Add Loyalty Card’ button will proceed as above. If it is still incorrect, then it should continue to inform the user as per NOTE 1.

**NOTE 2**: There is no need to carry out any validation of the name data entered (other than it is there). You may assume that we will test your work with valid names.

**Confirm**

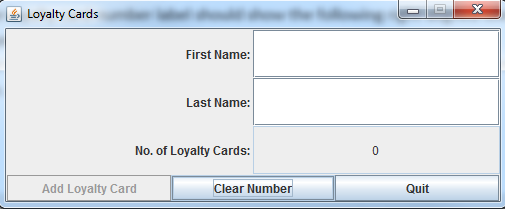
When the ‘Confirm’ button is clicked:

* The loyalty card details should be written to the ArrayList object;
* The text fields should be cleared and the text of the third label removed;
* The button text ‘Confirm’ returned to ‘Add Loyalty Card’ready for the next loyalty card;
* Enable the ‘No. of Loyalty Cards’ button

**No. of Loyalty Cards**

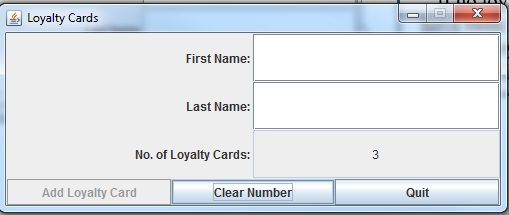
When the ‘No. of Loyalty Cards’ button is clicked:

* Disable the ‘Add Loyalty Card’button
* The text on the ‘No. of Loyalty Cards’ button should be set to ‘Clear Number’;
* The loyalty card number label should show the following right-aligned text – ‘No. of Loyalty Cards:’
* The loyalty card number text field should show the number of cards as a numeric (i.e. 6 not six)



If no loyalty cards

have been created and ‘No. of Loyalty Cards’ is clicked.



If 3 loyalty cards

have been created

**Clear Number**

When the ‘Clear Number’ button is clicked:

* The button text should be set back to ‘No. of Loyalty Cards;
* The loyalty card number label text should be removed:’;
* The loyalty card number text field should be cleared.

Marking Scheme: (6 Marks / 7.5% of module total)

|  |  |
| --- | --- |
| Mark | Criteria |
| 0 | No serious attempt / program does not compile |
| **OR** | |
| 1 | Appearance (0.5) / location on screen (0.5) |
| 1 | JLabels - correct & right-aligned |
| 1 | JTextFields – correct /cannot enter text directly into the membership number text field |
| 3 | actionPerformed method:   * changes button text (e.g. Add Loyalty Card -> Confirm -> Add Loyalty Card) * generates sequential card numbers if neither name field empty * Shows number of card holders |