

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

|  |  |
| --- | --- |
| **Module Title:** | *Programming 1* |
| **Module Number:** | *KV4000* |
| **Module Tutor Name(s):** | *Alan Maughan* |
| **Academic Year:** | *2018/19* |
| **% Weighting (to overall module):** | *15%* |
| **Coursework Title:** | *Homework 1* |
| **Average Study Time Required by Student:** | *8 hours* |

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

|  |
| --- |
| **Date of Hand out to Students:**  Week 2 |
| **Mechanism for Hand out to Students:**  *via eLP* |
| **Date and Time of Submission by Student:**  During Week 3 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 3. |
| **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:**

See below.

**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 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 1

This work is due at the start of your lab in week 3. It counts for 15% towards the module mark.

This homework is due to be marked at the start of your lab class in week 3. You must bring with you a printed copy of your source code (the .java file). This should be produced before you come to the lab. Do not come to the lab and attempt to print a copy then. The code will be collected by a tutor when they mark your homework (make sure that it has your name / id on it - @author your name / your id). This will be retained for audit and internal moderation. If this file is not available (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 (in this and in all subsequent homeworks) 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.

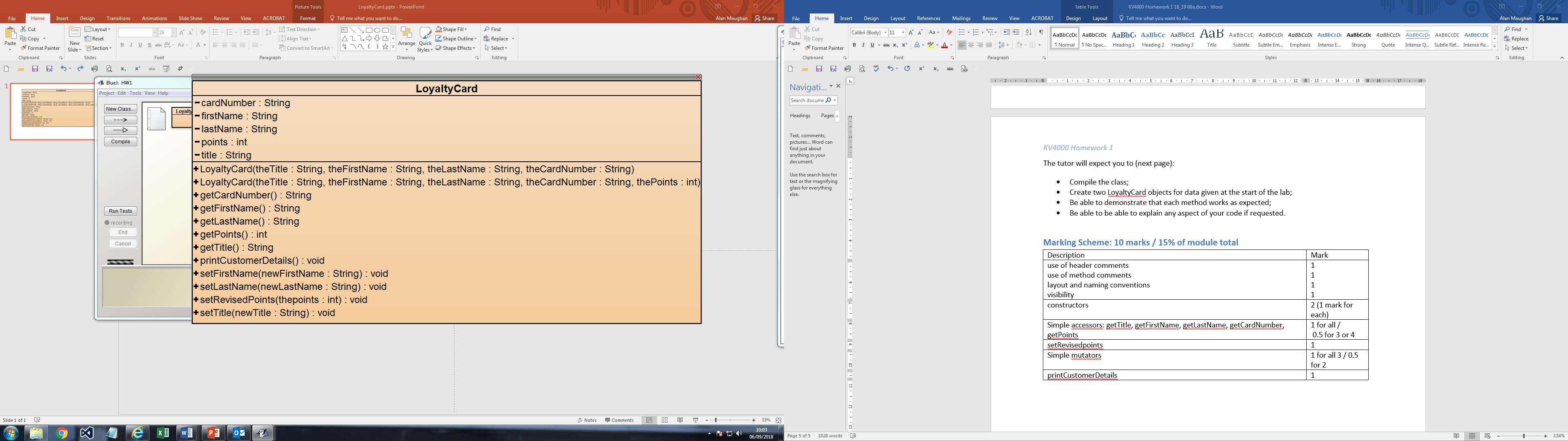
You may be asked questions about your program to confirm your understanding and that it is your own work. Failure to answer the questions may result in a deduction or total loss of marks.

***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

This homework and those that follow are based upon the requirements for a loyalty card for a fictional university bookshop. For this homework, you are expected to write simple class LoyaltyCard representing a rewards card. The BlueJ project should be called “HW1”.

The structure of the class is as per the class diagram on the next page (Note: The attributes / methods do not have to be in alphabetical order).



There are two constructors. Both will receive parameter data for title (Dr, Mr, Ms, etc.), first name, last name and card number (which you can assume for now will be a 3-digit string) and will assign these values to the relevant attributes.

* One constructor will set the spend to ‘0’ on the assumption that a new customer may obtain a card before making a purchase.
* The other constructor should receive a further parameter to specify an initial number of points obtained when the card was obtained. At the present, there is no need to validate the number supplied, which could be any positive integer value – including zero (0).

The class should be written as per the class diagram – do not use different names for attributes / methods; add additional code / functionality; etc. If you do then you will lose marks. If in doubt, ask a tutor.

**Methods**

getTitle, getFirstName, getLastName, getCardNumber and getPoints are simple accessor methods.

setTitle, setFirstName and setLastName are simple mutator methods

setRevisedPoints will update the total amount spent.

printCustomerDetails should return the customer’s details formatted as follows:

[title] [firstName] [lastName]

Card Number: [cardNumber]

Points avaiiable: [points]

i.e. Dr Fred Frog

Card Number: 1234

Points avalable: 23

NOTE: Don’t forget the spaces - “Dr Fred Frog” not “DrFredFrog”

NOTE: There is no mutator method for cardNumber. The program should not allow this to be changed.

The tutor will expect you to (next page):

* Compile the class;
* Create two LoyaltyCard objects for data given at the start of the lab;
* Be able to demonstrate that each method works as expected;
* Be able to be able to explain any aspect of your code if requested.

## Marking Scheme: 10 marks / 15% of module total

|  |  |
| --- | --- |
| Description | Mark |
| use of header comments  use of method comments  layout and naming conventions  visibility | 1  1  1  1 |
| constructors | 2 (1 mark for each) |
| Simple accessors: getTitle, getFirstName, getLastName, getCardNumber, getPoints | 1 for all /  0.5 for 3 or 4 |
| setRevisedPoints | 1 |
| Simple mutators: setTitle, setFirstName, setLastName | 1 for all 3 / 0.5 for 2 |
| printCustomerDetails | 1 |