

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

|  |  |
| --- | --- |
| **Module Title:** | *Programming 1* |
| **Module Number:** | *CG0047* |
| **Module Tutor Name(s):** | *Alan Maughan* |
| **Academic Year:** | *2014/15* |
| **% Weighting (to overall module):** | *20%* |
| **Coursework Title:** | *Homework 3* |
| **Average Study Time Required by Student:** | *8 hours* |

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

|  |
| --- |
| **Date of Handout to Students:**  Week 6 |
| **Mechanism for Handout to Students:**  *via elp* |
| **Date and Time of Submission by Student:**  During Week 8 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 8. |
| **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**

**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:

|  |  |  |
| --- | --- | --- |
| Category | Mark | Description |
| Coding style | 2 | use of header & method comments  visibility  layout and naming conventions |
| Program compiles | 1 | **AccountList** class compiles. |
| AccountList class  **(Must contain student’s name in @author tag or 0 marks for class!)** | 2 | Use of ArrayList |
| 7 | 1 mark each |

**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.

# Homework 3

This work is due at the start of your lab in week 8. It counts for 20% of the overall module mark.

This homework is due to be marked at the start of your lab class in week 8. 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. These will be collected by the tutor 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 (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.

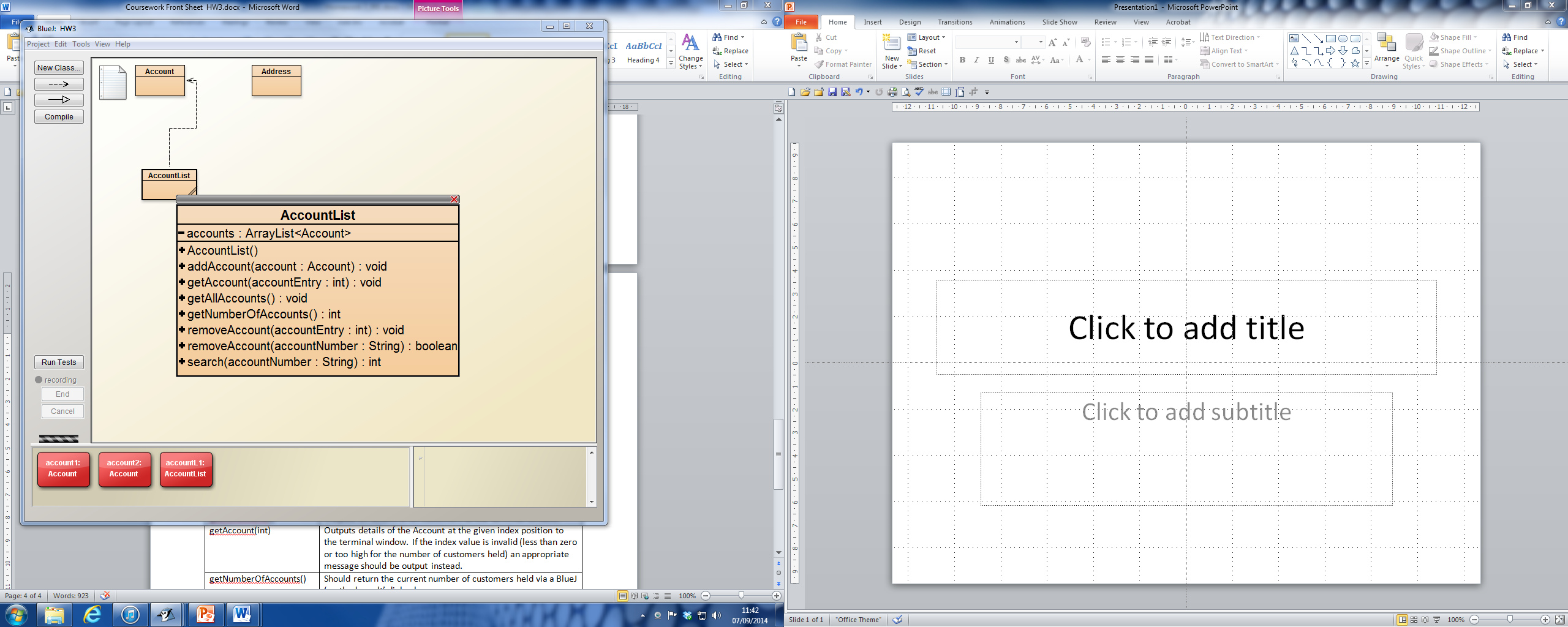
***The work must be wholly your own. Collusion counts as academic misconduct and will be punished according to the University’s regulations detailed in “Assessment Regulations for Northumbria Awards” (ARNA) a copy of which is available on the University website.***

## The Task

Create a project should be called “HW3”.

For this homework you are expected to write a program based upon the **Account** and **Address** classes.

You are expected to write a class **AccountList** which represents a collection of Accounts. (The work to do this will be based upon the material delivered in weeks 6 & 7). **Account and Address** classes are supplied on Blackboard. There is no copy of the **AccountList** class. You need to write this class. The structure of the class is:



The class must make use of an ArrayList. Methods are:

|  |  |
| --- | --- |
| Method | Comments |
| addAccount(Account) | Allows an Account to be added to the collection. |
| getAllAccounts() | Outputs all details of all customers to the terminal window. |
| getAccount(int) | Outputs details of the Account at the given index position to the terminal window. If the index value is invalid (less than zero or too high for the number of customers held) an appropriate message should be output instead. |
| getNumberOfAccounts() | Should return the current number of customers held via a BlueJ ‘method result’ dialog box. |
| removeAccount(int) | Should remove the Account at the index position specified as a parameter. |
| removeAccount(String) | Should remove the Account with the account number specified as a parameter if it the Account exists. The appropriate boolean value should then be returned via a BlueJ ‘method result’ dialog box. |
| search(String) | Should search the ArrayList for a Account with the account number specified as a parameter. The appropriate index value should then be returned via a BlueJ ‘method result’ dialog box. If not found ‘-1’ should be returned instead. |