Department of Computer Science

Coursework Assessment Specification

Module Number: 08120 Title: Programming 2

Lecturer: Rob Miles

Coursework Assessment Number: 2 of **2**

Title: Mastermind

Method of Working: INDIVIDUAL

Workload Guidance: Typically, you should expect to spend between **3** and **10** hours on this assessment in addition to time spent in scheduled laboratories.

Date of publication: Monday 23rd of April 2007

Date and time for submission: 9.30 am on **Thursday 10th May 2007**

This assignment should be submitted electronically via the Class Server program on the due date unless an extension has been authorised on a *Request for an Extension for an Assessment* form which is available from the Office or http://www.student-admin.hull.ac.uk/Mitcircs.doc. The extension form, once authorised by the lecturer concerned, should be attached to the assignment on submission (or given to the lecturer in the case of electronic submission). The assignment will be demonstrated by the student**.**

Notes:

Late penalties
For work submitted late the penalty is loss of **20% marks per day**. Work that is **5 or more days late** will automatically be graded as **FAIL**, and no re-submission will be allowed. For full details, see the Department's student handbook.

Use of Unfair Means
You are reminded of the University's Code of Practice on the Use of Unfair Means (http://www.student-admin.hull.ac.uk/unfair.html) and that the work you submit for assessment should contain no section copied in whole or in part from any other source unless where explicitly acknowledged by means of proper citation.

ASSESSMENT STRATEGY AND ASSESSED COURSEWORK

This assignment is worth **30** % of the module marks.

The overall assessment strategy is designed to evaluate the student’s achievement of the module learning outcomes, and is subdivided as follows:

ASSESSMENT STRATEGY AND LEARNING OUTCOMES

This assignment contributes towards the following module learning outcomes.

LO	Learning Outcome	Method of Assessment
1	*Analyse Problems*	*Software Presentation*
2	*Analyse Algorithms*	*Software Presentation*
3	*Object based C# Programming*	*Software Presentation*
4	*Algorithmic Deployment*	*Software Presentation*
5	*Use of Abstraction*	*Software Presentation*
6	*Evaluate Existing Code*	*Software Presentation*

ASSESSMENT CRITERIA

Assessment Criteria	Contributes to Learning Outcome	Mark
Application functionality	1,2,3,4,5,6	**40%**
Test Harness	3,4	**10%**
Additional Features	3,4	**20%**
Appropriate Programmer Documentation	3,4	**10%**
Appropriate User Documentation	3,4	**10%**
Evidence of Good Design	2,3,4,5,6	**10%**

Account System

For this practical assignment you are required to create a Bank Account management system. The information that must be held in each account is:

- The name of the account holder
- The address of the account holder
- The balance of the account
- The overdraft limit on the account

This information is to be held in an Account class, and you have been employed to create this class and a user interface which will interact with it.

The set of behaviours that the customer has asked for is as follows:

- pay money into the account
- draw money out of the account
- view the balance
- change the name and address of the account holder
- change the overdraft limit
- view the overdraft limit

All the methods in the Account class should perform proper validation. In particular:

- It should not be possible to pay in a negative amount of funds.
- It should not be possible to withdraw a negative amount of funds.
- It should not be possible to withdraw more money that is in an account.
- It should not be possible to create an account with a name which contains numeric digits.

The system should include a set of tests which can be run as evidence that the Account class performs the above behaviours correctly.

The bank itself will be represented by a Bank class which will provide facilities for storing and retrieving accounts, based on the account name property. The Bank class should refuse to store an account which has the same name as an existing one.

User Interface

The system should provide a Windows Forms based user interface which will allow the above account behaviours to be accessed.

This should include a means by which a given account can be located by account name. In addition, when displaying the balance of an account this value should be displayed in red if the account is overdrawn.

Bank Account Application Extensions

If you want to add extensions to the bank system you can also think about adding the following:

- Bank accounts should have an account number as well as an account holder name. This number should be unique for every account. It should be possible to find an account in the bank based on the account number instead of the name.
- The bank account information should be stored in a file when the program exits and loaded again when the program starts.

You can add further extensions if you wish, once the core features have been implemented and tested, but you should remember that beyond a certain point it will not be possible for them to attract further marks for this particular assignment.

Rob Miles