MATH6005 Referral Assignment 2017-18

# Instructions

Your final assignment submission should consist of an Excel file and accompanying final report submitted electronically via Blackboard. This assignment counts for **100% of the referral assessment** for MATH6005. The deadline for this assignment is **15:00pm on Friday 24th August 2018**. This applies to both the Excel file and final report. The deadline is strict. Normal penalties for late assignments apply: 10% of your marks are deducted per working day late, with no submission permitted after five days.

Ensure that you take frequent and multiple backups of your work, since excuses concerning lost or corrupted files will not be treated sympathetically. This holds also for faulty submissions via email. Please, verify that you follow all instructions carefully and that your work has been uploaded successfully.

## Electronic Submission

* The electronic submission should be:
  + Your VBA project in a single Excel workbook; and
  + Accompanying logbook in a single final report.
* Please name both your Excel and Word files using BS\_ (for Business School students) or OR\_ (for OR [and Finance] students) VBA\_ followed by your student ID, e.g. BS\_VBA\_12345678 or OR\_VBA\_12345678.
* Both your workbook and the accompanying logbook should be submitted via Blackboard and via email to [P.Hiscock@soton.ac.uk](mailto:P.Hiscock@soton.ac.uk) with “VBA Referral Assignment submission” in the title, by the deadline above.
* You are reminded that if the statement **Option Explicit** is not included at the top of the source code in every non-empty module, class and UserForm in your project, you will be awarded **0%** for the VBA project, including its functionality.
* Please also keep a copy of your VBA project and your logbook in case there is a problem with the files you submit.

## Collaboration, plagiarism and cheating

* Unlike the exercise sheets for the course, you should work on your own when carrying out the assignment.
* Your report should acknowledge clearly all the people with whom you have discussed any part of the assignment, as well as any references you may have used.
* Please refresh your memory of the University’s code on academic integrity, see http://www.calendar.soton.ac.uk/sectionIV/academic-integrity-regs.html.
* Please note that allowing somebody else to copy your work is counted as plagiarism: it carries the same penalty as copying work.

## Purpose of assessment

The purpose of this assignment is to assess your ability to:

* Demonstrate the ability to write a structured computer program.
* Demonstrate good programming practice, as discussed in the course notes, lectures and computer workshops. This includes the usage of object-oriented programming.
* Demonstrate good and correct use of VBA.
* Provide suitable documentation for the code so that it can be understood by a programming-literate reader.

Although the focus of this assessment is on programming skills and not on report writing, your printed document should be sensibly formatted (including page numbers and section headings) and well presented.

The Problem

# Banking Management System

## Introduction

A bank has hired you to develop a VBA application to emulate their customer management system. The program will have features for a manager to:

1. View the list of existing customers (their ID, balance and transaction history);
2. Add new customers; and
3. Save the customers.

The program should also allow customers to log-in, view their balance, withdraw and deposit money as outlined in section 2.2. Best practice requires you to provide the company with an accompanying logbook for the tool which meets the criteria laid out in section 2.4.

## The Banking Management System

The program has a user interface for the manager to see the current list of customers. A sample user interface is given in Figure 1.



Figure 1: Sample GUI showing customer list.

On clicking on a customer, the manager should be able to see further details on the transaction history, as shown in Figure 2.

****

Figure 2: Sample GUI showing transaction history of a customer.

The manager is able to add a new customer, as shown in Figure 3.

****

Figure 3: Sample GUI for the manager to create a new customer’s account.

The customers can log-in to the system to view their balance, to withdraw and deposit money, as shown in Figure 4.

****

Figure 4: Sample GUI for an ATM machine.

***These are only sample user-interfaces and you are encouraged to develop your application according to your imagination. For example, the transaction history GUI could have a bit more information about the customer. The ATM user interface could only enable the Deposit and Withdraw buttons after the user has successfully logged in. You could also move the ATM Interface to a separate form (which could make more sense than the sample GUI shown).***

## Your VBA Tasks

Your tasks include designing the user interfaces and implementing in a VBA project the system that provides features as described above. Your program should be able to:

* Read data from a comma-separated (CSV) file that contains the customer information;
* Recognise if there are any problems with the data (i.e. perform validation);
* Allow the manager to view the list of customers, view their transaction history and add new customers;
* Allow customers to log-in to the system to view their balance, to withdraw and deposit money;
* Update the customer information whenever changes are made;
* Save the updates; and
* Output the results.

In addition, a sample data file is included on Blackboard. You might find this useful for initial testing of your code, but you are also **strongly** encouraged to create your own to test the data validation in your program and to **test the results of your VBA code, ensuring they match solutions derived independently (e.g. by hand)**.

## Your written task

To convince users that the tool you have developed functions as necessary you are required to produce a logbook which contains the following.

* User guide section (worth 10%). The user guide should explain to an Excel-literate (but not VBA-literate) user how to use the software from the point of opening the workbook. This document should be a concise guide, bearing in mind that a well-designed user interface which follows a clear and natural workflow needs little explanation.
  + Screenshots are not essential but may be useful.
  + Assumptions and limitations must be specified.
  + Warnings and/or messages given by the program must be explained. It is not essential, but you might want to extend this to be a section on how to troubleshoot the software when things go wrong.
* Technical section (worth 10%). The technical documentation should describe the structure and components of the VBA project to a VBA-literate reader so that it can be maintained and enhanced in the future.
  + Concisely describe the purpose of each file (module, class, UserForm) in the VBA project.
  + Provide a **brief** description of the purpose of each routine, method, property and event. (This should be no more than a few sentences for each; often one will suffice.)
  + An explanation of how the files and the main elements of the program are linked together – what is the structure of the code, when is a routine called and from where?

# Marking Scheme

80% of the marks available for this assignment are for the VBA project; the remaining 20% of the marks are available for the accompanying document.

* Marks for the VBA project (80%) will be awarded on the basis of:
  + Functionality when it runs (correct results and output);
  + User-friendliness when running (robustness, data validation, error-handling);
  + Design and maintainability (including usage of object-oriented programming); and
  + Originality (in terms of design and/or additional functionality).
* Marks for the documentation (20%) will be awarded on the basis of:
  + Accuracy;
  + Completeness; and
  + Presentation.

***Remember: Option Explicit must be included in all the files used***. ***Otherwise, a zero mark will be given for the electronic submission!***

*Since Object-Oriented Programming is one of the most important topics in MATH6005, you are asked to make use of classes. A submission can only obtain a* ***Distinction*** *if your program contains at least one proper class with appropriate data and methods. The program should make significant use of this class (instead of just design it and leave it there). However, if you aim for a* ***Pass/Merit****, it is possible to avoid use of classes (but instead use UDT or a number of arrays).*