**MATH6005 VBA Coursework 2014/15**

**Instructions**

Your VBA coursework should consist of an electronic submission and printed copies of written documentation. The deadline for the coursework is at **3:00pm on Thursday 23rd April 2015**. This applies to both the electronic submission and the written documentation. 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 your work has been uploaded successfully.

**Electronic Submission**

• The electronic submission should be your VBA project in a single Excel workbook.

• Please name your Excel workbook using VBA\_ followed by student ID, e.g. VBA\_12345678.xlsm

• Your workbook should be submitted via blackboard and via email to m.l.penn@soton.ac.uk

• **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 electronic submission, including its functionality.**

• Please also keep copies of the file that is submitted electronically and your written documentation in case there is a problem with the files you submit.

**MSc OR [and Finance] students**

This coursework will count for 80% of the assessment for MATH6005.

• Your electronic submission must be submitted via blackboard and via email (see above).

• Printed copies of your written documentation must be handed in to the **Student Office Building 58 (Room 2127)**. An Assignment Hand In Cover must be completed and stapled to the front of your work prior to submission. Make sure that you are given a receipt when you submit your coursework. The office is open between 10:00 and 16:00, and you may only submit coursework during these hours. The deadline is strict. Please refer to the OR [and Finance] Handbooks for the penalties applied to late assignments.

**MSc Management Science students**

This coursework will count for 80% of the assessment of the Visual Basic for Applications component of MANG6005.

• Your electronic submission must be submitted via blackboard and via email (see above).

• Printed copies of your written documentation must be handed in to the **Student Office Building 58 (Room 2127)** using the procedure described above for MSc OR [and Finance] students. Normal penalties for late assignments apply.

**Collaboration, plagiarism and cheating**

• Unlike the exercise sheets for the course, you should work **on your own** when carrying out the coursework.

• Your report should acknowledge clearly all the people with whom you have discussed any part of the coursework, as well as any references you may have used.

• For cases of plagiarism and cheating the mark for this work will be reduced by 50% or a mark of zero will be given for this work.

• Please refresh your memory of your School’s stance on collaboration, plagiarism and cheating. These is listed in the OR [and Finance] handbooks.

• Please note that allowing somebody else to copy your work is counted as cheating; it carries the same penalty as copying work.

**Purpose of assessment**

The purpose of this assessment 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 your written documentation should be sensibly formatted (including page numbers and section headings) and well presented, the focus of this assessment is primarily on programming skills and not on report writing.

**Ship Rendezvous Problem**

**1 Introduction**

In the Ship Rendezvous Problem (SRP), a support ship must visit *n* other ships (the task force), visiting each ship exactly once. The support ship travels at a constant speed and changing direction takes negligible time. Each ship in the task force moves at a constant velocity (i.e. speed and direction). The objective is to minimise the total time taken to visit all the ships in the task force (i.e. the time taken to reach the final ship).

This coursework deals with the 2-dimensional (2D), Euclidean version of the SRP. A problem instance is defined by specifying the starting (x, y) coordinates for all ships (including the support ship), the speed of the support ship, and the velocity of the ships in the task force.

Note that if the support ship is faster than all other ships in the task force, then the SRP certainly has a finite solution. If, on the other hand, one or more of the ships in the task force is faster than the support ship then it is very likely (but not certain) that the SRP has no solution.

**2 Greedy heuristic for the SRP**

If the support ship is faster than all other ships in the task force then a simple way of finding a solution to the SRP is to use a greedy heuristic. This guarantees a solution, but it is very unlikely to be optimal.

The greedy heuristic works as follows:

Step 1. For each unvisited ship in the task force, calculate how long it would take the support ship to intercept it from the support ship's current position.

Step 2. Choose the ship, *i* say, which can be reached in the shortest amount of time.

Step 3. Visit ship *i* and update the positions of the ships in the task force.

Step 4. If unvisited ships remain, return to Step 1.

Note that in order to make the heuristic deterministic (i.e. to guarantee the same result each time it is run on the same problem instance), we must specify how ties are broken in Step 2. The most commonly used tie-breaker is to choose the ship with the smallest index (for example, if ships 5 and 7 can be reached in the same amount of time, ship 5 should be chosen in preference to ship 7). See the technical document (available via Blackboard) for details on how to calculate intercept times.

**3 Your Task**

You must implement the greedy heuristic for the 2D, Euclidean SRP in a VBA project in Excel. Your program must have a single user interface so that the user can:

* Read data from a comma-separated (CSV) file.
* Recognise any problems with the data.
* Run the greedy heuristic.
* Output the results.

Detailed instructions are given in the technical document.

**3.1 Getting you started**

You may find the Excel workbook ShipRendezvous\_Starter.xls a useful starting point. Note that it needs extending - search for the phrase “\TO DO" in the VBA project for starting points for further work. You are welcome to use as much or as little of ShipRendezvous\_Starter.xls as you wish in your final project.

In addition, two data files are attached. You might find these 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 algorithm, ensuring they match solutions derived independently (e.g. by hand).

**3.2 Written documentation**

You should provide documentation covering two areas:

1. User guide (worth 10%). The user guide should explain to an Excel-literate (but not VBA-literate) user how to use the software. It should be brief; bear in mind that well written software needs less explanation.

* Screenshots are not essential, but are useful.
* Any assumptions and limitations should be specified here.
* An explanation of any warnings and/or messages given by the program should be given. It is not essential, but you might want to combine this with a troubleshooting section.

2. Technical Documentation (worth 10%).

The technical documentation should describe the components of the program to a VBA-literate reader so that it can be maintained and enhanced in the future.

* The purpose of each file (module, class, UserForm).
* For each file, a brief description of the purpose of each routine, method, properties 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.

**3.3 Marking scheme**

20% of the marks will be available for the written documentation and the remaining 80% for the VBA project submitted electronically.

Marks for the documentation (20%) will be awarded on the basis of:

* its accuracy;
* its completeness;
* its presentation.

Marks for the electronic submission (80%) will be awarded on the basis of:

* its functionality when it runs (correct results and output);
* its user-friendliness when running (robustness, data validation, explanation of problems);
* its design and maintainability (readability);
* originality (for example additional features, or adding the option to run a better heuristic – after that specified above has been used).

**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 course work 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 designing it and leaving it there). However, if you aim for a* ***Pass*** *only, it is not essential that you use classes (but instead use UDTs and/or a number of arrays). Notice, however, design, maintainability and clarity are important marking criteria (see Marking Scheme). To help you with the coursework, the basic components of some classes have been included in the starter file (see section 3.1). You are welcome to use these or to design your own classes.*