Faculty of Technology – Course work Specification

Module name:	**AI Programming**	
Module code:	**IMAT5118**	
Title of the Assignment:	**Implementation of NLP using Prolog**	
This coursework item is: (delete as appropriate)	Summative	
This summative coursework will be marked anonymously	Yes	
The learning outcomes that are assessed by this coursework are: a. Learn to do programming with Prolog b. Learn about NLP c. Learn the basic of OS shell commands		
This coursework is: (delete as appropriate)	Individual	
If other or a mixed ... explain here:		
This coursework constitutes 50 % **to the overall module mark.**		
Date Set:	**Pending**	
Date & Time Due:	**Pending**	
The 'normal' coursework return date for this work is:		
When completed you are required to submit your coursework to: 1. Blackboard shell via TurnitIn		
Late submission of coursework policy: Late submissions will be processed in accordance with current University regulations which state: *"the time period during which a student may submit a piece of work late without authorisation and have the work capped at 40% if passed is **14 calendar days**. Work submitted unauthorised more than 14 calendar days after the original submission date will receive a mark of 0%. These regulations apply to a student's first attempt at coursework. Work submitted late without authorisation which constitutes reassessment of a previously failed piece of coursework will always receive a mark of 0%."*		
Academic Offences and Bad Academic Practices: These include plagiarism, cheating, collusion, copying work and reuse of your own work, poor referencing or the passing off of somebody else's ideas as your own. If you are in any doubt about what constitutes an academic offence or bad academic practice you must check with your tutor. Further information is available at: http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/academic-offences.aspx and http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/bad-academic-practice.aspx		
Tasks to be undertaken: **The task you are set is to use Prolog to expand the template system studied in the lessons that lets the user type commands to MS-DOS into Natural English Language. The system will translate English into a formal language, in this case MS-DOS command language.** **You will expand this system so that it becomes a practical (if still modest) piece of software. Modify it in any ways you can think of that will make it more useful. This is an open-ended project.** You have a number of tasks: • Add 5 more basic MS-DOS commands to the system. • Make a good attempt at considering a large amount of Natural English Language		

sentences that the system understands.

We will be looking for:

- Sensible extensions to the system.
- Sensible choice of commands.
- Sensible choice of simplified and translation rules.
- Careful thought about the various design parameters for the system.

For instance: the Natural English Language phrase:

What are the files on my a: drive

should be translated into the MS-DOS command:

dir a:*.*

We are not looking for script English commands such as:

show files in a:

instead we are looking for proper English language sentences to be translated into useful MS-DOS commands

Task 2
Having developed your system with Prolog, the next task is to evaluate Prolog according to your work. To do this, you need to compare Prolog with a popular procedural language, such as C, C++, Java, VB, etc. Based on your evaluation, you should answer the question: which language would you use if a real world system like this was assigned to you.

You will write a report summarizing your approach and results for task 1 and 2. An indicative structure might be:

- Introduction
- The System
- Results

1. Simplified rules
2. Translation rules
3. Examples of the system working
4. Comparison of Prolog with a procedural language
4. Appendices

Deliverables to be submitted for assessment:
The main body of the report should be a maximum of eight pages. A separate ASCII (No WORD document) file containing the Prolog source code should be submitted to the Blackboard separately.

A viva will be required after your submit your work: 5 minutes presentation, 5 minutes demonstration of your programme and 5 minutes questions. Distance learners need to be prepared to do your demo with systems like Skype. This will be specified near the time.

We will be looking for the following:

Some thought has gone into how the basic template system can be improved. You should show that you have thought about the simplification and translation rules.

You should have thought about what MS-DOS commands and English templates to use to enhance the system.
You should demonstrate good knowledge on the advantages and limits of Prolog language in comparison with other procedural languages.
The report should be well constructed and well written.
We are looking for initiative in tackling the problem and presenting the results.

How the work will be marked:

A - An excellent, well-written report that is well structured and makes an interesting read. You will have explored the templates in a number of ways and have a program that performs well. You have analysed the performance of the system and presented the results in an interesting and sound way. As well as dealing with points B,C and D, you should provide a program that can be used by any user who is not familiar with MS-DOS. Extra credit will be given for attempts to adapt this program to work under Linux command shells. Your comparison of Prolog against a procedural language gives a convincible argument for your point, and you demonstrate excellent knowledge of both languages.

B - A well-written report that is well structured and an interesting read. You have a program that performs well on the data and you have analysed their performance. You have exhibited some initiative in the approach taken and the results are presented clearly. As well as doing requirements C and D, your program should be able to filter out the relevant information from a sentence. Your comparison of Prolog with a procedural language is interesting but not fully convincible enough.

C- A reasonable report that presents an account of the approach taken and the templates used. The program performs well and the results are presented reasonably clearly. In addition to the requirements in section D, you have changed the logic of the program so that it tells the user when it does not understand new previously unseen commands. The comparison is reasonable but not enough details and no clear evidence.

D **- A report that presents some results of enhanced system. You have added a reasonable amount of templates to the existing Prolog program and the program runs properly without halting with errors. A basic comparison with no details and evidence.**

F **- Either no report submitted or a report that shows little or no understanding of how the template matching technique works. No understanding of either languages in the comparison.**

For more details on the grading scheme please see attached appendix.

Module leader/tutor name:	**David Elizondo, Muhanad Younis**
Contact details:	**Elizondo@dmu.ac.uk, muhanad.younis@email.dmu.ac.uk**

Extensions: Please read the regulations for extensions. ONLY IF you believe that you fall into any of the criteria, please send me an email clearly highlighting the criteria that justifies an extension.

What constitutes acceptable grounds for extension, and, more importantly, what does not.

The student regulations give a range of **acceptable** grounds, and these can be found at this link

http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/deferral-of-assessments.aspx

The link also indicates that third-party evidence will normally be required.

In particular, these are examples of grounds that are considered **unacceptable** grounds for granting an extension:

- Failure of technical media resulting in lost files, this includes hard disk crashes, memory stick failure, memory stick loss, virus attack on a computer. As users of technology, students should make regular backups of all their work; their account on the DMU network is a good place to store files as they are backed up by the server network automatically
- Several assignments all being due in at the same time; students should plan their time around all their assessment deadlines

There are two other points to bear in mind:

- If there is some technical incident that applies to the whole module or class, then the module leader could and should take some appropriate action, for example extend the deadline for all the students
- If there are relevant local rules in programme handbooks, then that should also be taken into account