

DEPARTMENT OF COMPUTER SCIENCE ASSESSMENT DESCRIPTION 2014/15 (EXAM TESTS AND COURSEWORK)

MODULE DETAILS:

Module Number:	08226	Semester:	2
Module Title:	Artificial Intelligence		
Lecturer:	Mike Brayshaw		

COURSEWORK DETAILS:

Assessment Number:	1	of	1
Title of Assessment:	MicroWatson 2015		
Format:	Program	Demonstration	
Method of Working:	Individual		
Workload Guidance:	Typically, you should expect to spend between	35	and 40 hours on this assessment
Length of Submission:	This assessment should be no more than: (over length submissions will be penalised as per University policy)		N/A Coding Exercise (excluding diagrams, appendices, references, code)

PUBLICATION:

Date of issue:	21 st July 2015
----------------	----------------------------

SUBMISSION:

ONE copy of this assessment should be handed in via:	E-Bridge		If Other (state method)	
Time and date for submission:	Time	10:00	Date	25 th August 2015
If multiple hand-ins please provide details:				
Will submission be scanned via TurnitinUK?	No	If submission is via TurnitinUK within E-Bridge students MUST only submit Word, RTF or PDF files. Students MUST NOT submit ZIP or other archive formats.		

The assessment must be submitted **no later** than the time and date shown above, unless an extension has been authorised on a *Request for an Extension for an Assessment* form which is available from the Departmental Office (RB-308) or
<http://intra.net.dcs.hull.ac.uk/student/exam/Advice%20regarding%20resits%20in%20modules%20passed%20by%20compe/Forms/AllItems.aspx>.

A student who has submitted the wrong file to E-Bridge will still incur a late penalty if their resubmission is made after the coursework deadline.

MARKING:

Marking will be by:	Student Name
---------------------	--------------

COURSEWORK COVERSHEET:

BEFORE submission, you must ensure you complete the correct departmental ACW cover sheet (if required) and attach it to your work. The coversheets are available from: http://intra.net.dcs.hull.ac.uk/student/ACW%20Cover%20Sheets/Forms/AllItems.aspx	NO coversheet required as E-Bridge submission
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------

ASSESSMENT:

The assessment is marked out of:	100	and is worth	40	% of the module marks
N.B If multiple hand-ins please indicate the marks and % apportioned to each stage above (i.e. Stage 1 – 50, Stage 2 – 50). It is these marks that will be presented to the exam board.				

ASSESSMENT STRATEGY AND LEARNING OUTCOMES:

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

LO	Learning Outcome	Method of Assessment {e.g. report, demo}
LO 1	<i>Conceptualise simple cognitive tasks and processes in formal reasoning terms</i>	Demo
LO 2	<i>Devise an AI solution for a formalised task or process and to meet specified objectives.</i>	Demo
LO 3	<i>Practical Subject Skill Design and write programs in a symbolic programming language</i>	Demo

Assessment Criteria	Contributes to Learning Outcome	Mark
Prolog Language Program Design and Implementation	Demonstrate Practical Skills	100

FEEDBACK

Feedback will be given via:	Feedback Sheet	Feedback will be given via:	N/A
Exemption (staff to explain why)			
Feedback will be provided no later than 4 'teaching weeks' after the submission date.			

This assessment is set in the context of the learning outcomes for the module and does not by itself constitute a definitive specification of the assessment. If you are in any doubt as to the relationship between what you have been asked to do and the module content you should take this matter up with the member of staff who set the assessment as soon as possible.

You are advised to read the **NOTES** regarding late penalties, over-length assignments, unfair means and quality assurance in your student handbook, also available on the department's student intranet at:

- <http://intra.net.dcs.hull.ac.uk/student/ug/Handbooks/Forms/AllItems.aspx> (for undergraduate students)
- <http://intra.net.dcs.hull.ac.uk/student/pgt/Student%20Handbook/Forms/AllItems.aspx> (for postgraduate taught students).

In particular, please be aware that:

- Your work will be awarded zero if submitted more than 7 days after the published deadline.
- The overlength penalty applies to your written report (which includes bullet points, and lists of text you have disguised as a table. It does not include contents page, graphs, data tables and appendices). Your mark will be awarded zero if you exceed the word count by more than 10%.

Please be reminded that you are responsible for reading the University Code of Practice on the use of Unfair means (<http://student.hull.ac.uk/handbook/academic/unfair.html>) and must understand that unfair means is defined as any conduct by a candidate which may gain an illegitimate advantage or benefit for him/herself or another which may create a disadvantage or loss for another. You must therefore be certain that the work you are submitting contains no section copied in whole or in part from any other source unless where explicitly acknowledged by means of proper citation. In addition, **please note** that if one student gives their solution to another student who submits it as their own work, **BOTH** students are breaking the unfair means regulations, and will be investigated.

In case of any subsequent dispute, query, or appeal regarding your coursework, you are reminded that it is your responsibility, not the Department's, to produce the assignment in question.

MicroWatson

You are required to write a cut down micro program in the spirit of IBM's Watson . For background see

http://www.nytimes.com/2011/02/17/science/17jeopardy-watson.html?_r=1

<http://www.research.ibm.com/deepqa/deepqa.shtml>

or the following videos

<http://www.youtube.com/watch?v=SyXHF3PyyWo>

http://www.youtube.com/watch?v=WFR3IOm_xhE

<http://www.bbc.co.uk/news/entertainment-arts-12464447>

You are required to provide questions to answers that are typed in to the computer in the style of the TV game Jeopardy (<http://www.jeopardy.com/>). You should parse the answers that the users' provide and then attempt to match them against a knowledge base provided to reproduce the sentence that would be the question.

Specifically you need to provide the following in your ACW.

(a) You need to provide a MicroWatson Top Level that is a menu based system. There are three types of questions which you need to prompt for in a menu style. The categories are

1. Novels/Films of the 20th Century
2. Novels of the 19th Century
3. Novels of the 18th Century

Choice 2 and 3 can just be stubs and lead to an under construction sign. Choosing 1 leads to the micro_watson listener described below.

The menu should be initiated by the following goal.

?- micro_watson.

Upon choosing 1 micro_watson should respond with a question and then give the following prompt

Please give me an answer.

micro_watson : <ready for the user to type in their response>

the input should be in the form of a list (see end of the ACW for required inputs).

[15 marks]

(b) Now represent as Prolog facts the vocabulary of nouns, verbs, determiners, adverbs, and adjectives used in the `plot_story` knowledge based described at the end of the assignment as a hint

determiners include a, the

adjectives include young,middle_aged,magic, faithful, paranoid

prepositions include on,by

verbs include finds, saves, is

nouns include hobbit,ring,butler,day

adverb includes quietly

this should be in the form

det(the).

adj(paranoid).

prep(on).

verb(is).

adverb(quietly).

noun(butler).

[5 marks]

2. The user then enters their answer as a list for example

micro_watson: [the,paranoid,robot,was,marvin]..

You need to parse this sentence using the following Context Free Grammer (this is the one discussed in the lectures (Winograd, pp267))

- S -> NP VP /* John loves Mary */
- S -> VP /* Go! */
- NP -> Det NP2
- NP -> NP2
- NP -> NP PP
- NP2 -> Noun
- NP2 -> Adj NP2
- PP -> Prep NP
- VP -> Verb
- VP -> Verb Adverb NP
- VP -> Verb Adverb
- VP -> Verb NP
- VP -> VP PP

The program should output the following Prolog predicate.

sentence(noun_phrase(det(the),np2(adj(paranoid),np2(noun(robot)))),verb_phrase(verb(was),np(noun(marvin)))).

In addition you are required to output a formatted print of this parse of the form

sentence(
 noun_phrase(det(the),np2(adj(paranoid),np2(noun(robot)))),

verb_phrase(verb(was),np(noun(marvin)))).

[20 marks]

3. From the Prolog structure you have produce in 2. extract the subject (the noun in the principle noun phrase), the verb (the main verb in the verb phrase), and the object (the noun in the noun phrase within the verb phrase) and construct them into the following fact *parse_sov/3*. You should then have a record of who is doing what to whom and how. Your result should be the *fact parse_sov(subject(robot),object(marvin),verb(was))*.

using the current example.

[30 marks]

4. You now need to compute the match between the sentence features you have extracted in 3 with the *plot_story/2* predicate given at the end of this assignment. To do this you have to define an algorithm built in the spirit of Watson showing how good the match is with each *plot_story/2* and then choose the closest match as your question from *micro_watson* to the answer it was given as input..

Implement a predicate that compares the matches with the length of the *target list* (the first argument of *plot_story/2*) and chooses the best option. If the intersection is low (less than a majority) the microwatson should offer no answer. If it is greater than the majority or 100% as here the best answer should be given. In this case we have an exact match. The final question is going to be of the form 'Give me a plot element from the following book', then the name of the novel. An example question would thus look like:

Give me a plot element from the following book The Hitch Hikers Guide to the Galaxy.

[30 marks]

The Knowledge Base of Facts to use in this assignment.

A knowledge base made up of Prolog facts featuring some knowledge about the 20th Century literature.

plot_story([subject(hobbit),object(ring),verb(finds)],'The Hobbit').

plot_story([subject),object(ring),verb(destroys)],'The Lord of the Rings').

plot_story([subject(valet),object(day),verb(saves)],'Thankyou Jeeves').

plot_story([subject(holden),object(expelled),verb(is)] , 'The Catcher in the Rye').

plot_story([subject(robot),object(marvin),verb(was)],'The Hitchhikers Guide to the Galaxy').

.

Sample Questions that you need to be able to answer are

[a,young,hobbit,finds,a,ring].

[a,middle_aged,hobbit,destroys,a,ring].

[a,faithful,valet,saves, quietly,the,day].

[holden,is,expelled].

[the,paranoid,robot,was,marvin]..