UNIVERSITY OF Hull

DEPARTMENT OF COMPUTER SCIENCE
ASSESSMENT DESCRIPTION 2018/19
(EXAM TESTS WORTH ≤15% AND COURSEWORK)

MODULE DETAILS:

Module Number:	500086	Trimester:	1
Module Title:	Artificial Intelligence		
Lecturer:	Dr Mike Brayshaw		

COURSEWORK DETAILS:

Assessment Number:	1	of	1		
Title of Assessment:	Implementing a Simple Agent Chatbot				
Format:	Program				
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)*	please select **words** *(excluding diagrams, appendices, references, code)*			

PUBLICATION:

Date of issue:	2nd January, 2019

SUBMISSION:

ONE copy of this assessment should be handed in via:	Canvas		If Other (state method)	
Time and date for submission:	**Time**	2pm	**Date**	14th January, 2019
If **multiple hand–ins** please provide details*:*				
Will submission be scanned via TurnitinUK?	No	If submission is via TurnitinUK students MUST only submit Word, RTF or PDF files. Students MUST NOT submit ZIP or other archive formats. Students are reminded they can **ONLY** submit **ONE** file and must ensure they upload the correct file.		

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: http://www2.hull.ac.uk/student/registryservices/currentstudents/usefulforms.aspx

If submission is via TurnitinUK within Canvas staff must set resubmission as standard, allowing students to resubmit their work, though only the last assessment submitted will be marked and if submitted after the coursework deadline late penalties will be applied.

MARKING:

Marking will be by:	Student Number

ASSESSMENT:

The assessment is marked out of:	100	and is worth	60	% 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	*Conceptualise simple cognitive tasks and processes in formal reasoning terms*	Program
LO 2	*Devise an AI solution for a formalised task or process and to meet specified objectives.*	Program
LO 3	*Design and write programs in a symbolic programming language*	Program

Assessment Criteria	Contributes to Learning Outcome	Mark
Prolog Language Program Design and Implementation	Demonstrate Practial Skills 1, 2, 3	100

FEEDBACK

Feedback will be given via:	Feedback Sheet	Feedback will be given via:	Select secondary method
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, which is available on Canvas - https://canvas.hull.ac.uk/courses/17835/files/folder/Student-Handbooks-and-Guides.

In particular, please be aware that:

- Your work has a 10% penalty applied if submitted up to 24 hours late

- Your work has a 10% penalty applied and is capped to 40 (50 for level 7 modules) if submitted more than 24 hours late and up to and including 7 days after the deadline
- 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%.

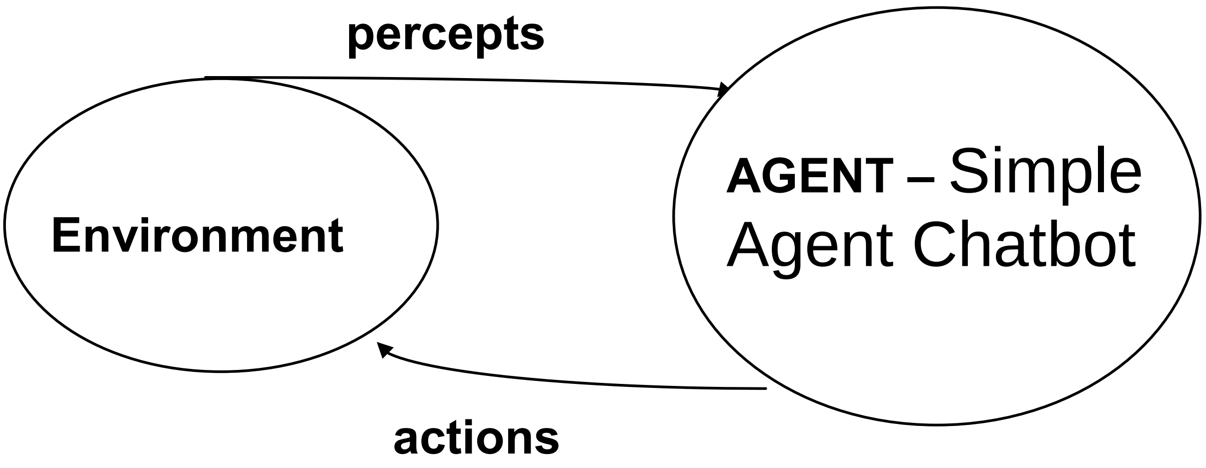
Please be reminded that you are responsible for reading the University Code of Practice on the use of Unfair means (http://www2.hull.ac.uk/student/studenthandbook/academic/unfairmeans.aspx) and must understand that unfair means is defined as any conduct by a candidate which may gain an illegitimate advantage 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.

500086 ACW: Artificial Intelligence: Implementing a Simple Agent Chatbot.

This assignment requires you to implement in Prolog an Agent that takes in Prolog list based natural language and chooses what action to take based first on best match using simple rules, Details can be found in the Lecture Materials, in particular lecture material in Week 8 and the Labs 10, 11, and 12. This assignment builds upon work you have already done in those Labs. This itself draws on material from Russell and Norvig http://aima.cs.berkeley.edu/ which should also be consulted for the necessary background to this ACW (in particular see Chapter 2).

The overall scheme of things can be summed up as follows



You should build upon the basic framework that we introduced in the Labs

(Lab 11 and 12) and the Prolog natural language parser we developed there. All answers *must* be extensions of this code.

So the top level will look like

```
agent:- perceive(Percepts),

action(Percepts).
```

Using **read(X)** the agent should perceive the world. **perceive/**1 should only contain actions associated with perception and not action. A clear distinction between agent functionality of perception and action will be required. The inputs will be in the form of the following sentences (cut and paste these into your editor as this will save retyping).

The program you have to write is an Chatbot the given a statement can suggest a suitable hobby based on what someone like/loves based on the input it is given. Below are given the English inputs that you are required to use and the corresponding outputs that are required

Inputs to be parsed	*Final Program Ouput*
[an, old, father, likes, a, good, book]	'He joins the book club.'
[a, teenage, boy, loves, horses]	'They join a riding club'
[a, sprightly, grandfather, loves, a, long, walk]	'He joins a rambling club.'
[a, social, person, likes, a, long, chat]	'They join a social club'
[a, young, student, likes, a, guitar]	'They should join a band'
[an, avid, petrolhead, loves, a, racing, car]	'They should go to the races'

Examples of how these sentences parse are given below (using **sentence/2**):

```
        sentence(np(det(an),np2(adj(old),np2(noun(father)))),vp(verb(likes),np(det(a),np2(adj
        (good),np2(noun(book))) ))

        sentence(np (det(a),np2(adj(teenage), np2(noun(boy)))),
vp(verb(loves),np(np2(noun(horses))))
```

The state agent will have some simple rules to do the translation. Rules are of the form **recommend/3**

```
recommend (likes,books, 'He joins the book club.').

recommend (loves,horses, 'They join a riding club').
```

You should use the following Grammar to parse **sentence/2** above
You need to parse this sentence using the following Context Free Grammer (this is as discussed in the lectures (Winograd, pp267) – it is the same grammar we use to build sentence in the early labs – do not implement adverbs)

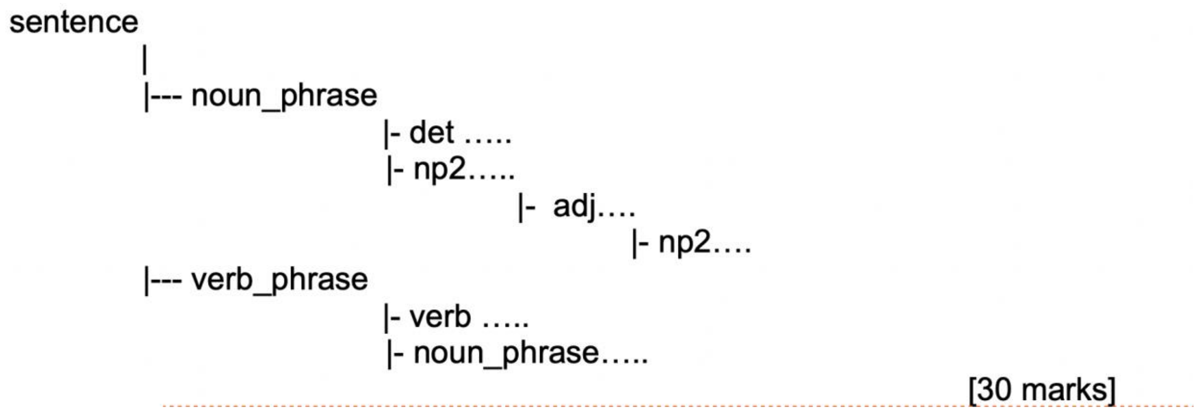
- S->NPVP /*John loves Mary*/
- S->VP /*Go!*/
- NP -> Det NP2

- NP -> NP2
- NP->NP PP
- NP2 -> Noun
- NP2 -> Adj NP2
- PP -> Prep NP

- VP -> Verb
- VP -> Verb Adverb NP
- VP -> Verb Adverb
- VP -> Verb PP
- VP -> Verb NP

Specifically the ACW requires you to do the following:

1. Develop a vocabulary e.g. **det(a**), **adj(short**), **noun(command)** that would allow Prolog to understand the words used in the sentence/2 examples above. This is similar to the vocabularies investigated earlier in the module. [10 marks]
2. Extend the definition of **recommend/3** so that they cater for the four other input sentences given. [10 marks]
3. Using your sentence parser and the rules that you have just developed write a parser that given the input given above produces a correct parse of all the sentences. Turn the sentence into a pretty printed parse tree linking structure of the parse with appropriate ASCII graphics to indicate the tree structure. The overall picture should look as follows



```
sentence
          |
          |--- noun_phrase
                          |- det …..
                          |- np2…..
                                    |-  adj….
                                             |- np2….
          |--- verb_phrase
                          |- verb …..
                          |- noun_phrase…..
                                                                  [30 marks]
```

4. Using only the structure produced in 3 above use the match **recommend/3** to find out and output the translation of the English sentence into. This is the Final Program Output in the table above. [25 marks]
5. One you have outputted your answer to Question 4 in the manner indicated your program should present the answer again in a graphical form in the following format.
   [25 marks]

```
sentence

        |         |- [an, old, father, likes, a, good, book]

        |         |- produces the recommendation that

                  |         |- ‘He joins the book club.’
```