**CS4346: Advanced Artificial Intelligence**

**Project #1: Spring 2020**

**Due Date**: Feb 19, 2020

Create an intelligent computer expert system for career advising to a freshman student who wishes to choose a major to pursue his/her studies. First use Backward Chaining methodology to determine the profession in which the student is interested. Use at least ten professions like Engineering, Science, Business, Medical, Applied Arts, Liberal Arts, Fine Arts, Communications, Health Care, and Education. Then use Forward Chaining methodology to recommend a specific area in which a student should pursue. For example, if the profession is engineering the area could be chosen from Electrical, Mechanical, Civil, petroleum, chemical etc.; if the profession is medical, the area could be chosen from mental health, oncologist, orthopedic, cardiologist, and General practitioner etc.; if the profession is health care, the area could be chosen from nursing, Health Services Management, Healthcare Administration, Physician Assistant, and Public Health, etc. Use a minimum of five areas for each profession. You can use Web or any other source to collect knowledge to develop knowledge bases.

A student who wishes to avail advising service will enter data in response to prompts from the Expert system. Your expert system will first advise the profession and then a specific area in that profession.

After collecting knowledge, develop two decision trees; one for Backward Chaining to advise the profession and second for Forward Chaining to advise the specific area in the chosen profession. Then transform the decision trees into rules. The rules should contain variables. At least 50 rules should be used in total for backward chaining and forward chaining. Also, develop a user-friendly interface in restricted English format, which receives input data from a student, uses keyword matching, and responds in a restricted English format.

Implement the expert system program, employing Backward Chaining and Forward Chaining methodologies. Programs based on these methodologies are provided on TRACS. To give you experience in rewriting a better source code, these programs, written in C, are intentionally written poorly and are inefficient and erroneous. Rewrite these programs in C++ by employing Software Engineering principles which prohibits ‘GO TO’ statements and discourage global variable. Separate Knowledge base and Inference Engine parts of each program and bring efficiency in functionality and output using your creativity. Efficiency methods include dynamic memory management, use of objects, and Hashing functions. Though you can totally rewrite the programs, they must be based on the methodologies used in these programs. Using any programs from any other source including web will be treated as plagiarism subject to severe punishment.

Develop a user-friendly interface, which receives input data from a user in restricted English format, uses keyword matching, and responds in a restricted English format. Use Eclipse C++ language. Your ‘main’ function will call the two other functions, i.e., ‘Profession\_BW’ and ‘Area\_FW’.

**Form a team of a maximum of three students. You are expected to develop the decision trees, rules, interface, and the inference engines as a team. After programs are developed, each member of the team will follow the following steps individually:**

1. Run the program with the rules for a minimum of five students.
2. Print intermediate results to trace your program.
3. Analyze the results, and the efficiency of the program. It should include how good are the results, how much memory is used, how fast was the program, and **how the changes you made could have affected the efficiency and processing of the program.**
4. Write his/her REPORT containing the explanation of the problem and the domain, decision tree, rules, methodologies used (Backward and Forward), the program, detailed explanations of the modifications to the program you implemented, and the analysis of the program and the analyses of results as defined in item #3 above. Also, clearly identify the contribution of each team member.
5. To run your programs, downloads your Files with full instructions to run your programs.

**Warning: EACH STUDENT MUST PERFORM STEPS 1 TO 5 INDIVIDUALLY. DO NOT COPY PROGRAMS FROM ANY SOURCE EXCEPT THE PROGRAMS I HAVE PROVIDED THROUGH TRACS. ALSO, DO NOT COPY REPORT FROM YOUR PARTNER OR ANY OTHER PERSON.**

**Note: Each team must demonstrate the execution of the programs in Dr. Ali’s**

**office.**

**Project Submission Instructions**

**HARD COPY SUBMISSION:**

Prepare and submit your well documented hard copy **[ABSOLUTELY NO EMAILS; Minus 50 points for email submissions]** of the project report (**spiral bound**) containing the explanation of the problem and the domain, decision tree, rules, methodologies used (Backward and Forward), the program, detailed explanations of the modifications to the program you implemented, and the analysis of the program and the analyses of results.

**ELECTRONIC SUBMISSION:**

Upload source code file (.cpp) as well as other files needed to run the program and verify your results. Also, include a file with complete detailed instruction for running your program. Source code file naming convention must be: Project1-student ID.cpp; Naming convention of other files will be similar except the file extensions. The file uploading instructions are given below:

Login to TRACS

Access CS 5346 sec. #1 Fall 2018, then follow the following steps:

* In the left Tool list of the site, click [Assignments].
* Click the assignment's title.
* Add an attachment(s) to the assignment.
* To add an attachment, click [Browse]. Locate your attachment and click [Open].  If you would like to add multiple attachments, click [Browse] and browse for your file.
* If you need to remove an attachment, click [Remove].
* The attachment(s) will be listed in the order they were entered.
* You may also click [Preview] to view your submission.
* Click [Submit] when you are done.