

# CS156 Module 3 Week 6

## Programming Assignment #1

### SUMMARY:

This programming exercise involves completing the code of one function named `pl_true` that is part of Python module that contains functions to manipulate and interface with logic-based KBs. It is important to first complete the reading assignment for chapter 7 in the textbook, and review the lecture slides on Module 3 logic-based representation topic.

Is a function that takes as input a logical expression and a model. It returns True if the logical expression is true in the model and False/None otherwise. You will first need to review several functions in the `logic.py`, `logic_driver.py`, and possibly other modules to understand how propositional statements are inserted, removed, and evaluated, and so forth in a KB and models.

The function in the `logic.py` module is partially completed. Within that function are several lines with the following text.

```
<put the correct programming statement here.>
```

The inserted text indicates places where you need to insert Python statements/code to complete the implementation of the `pl_true` function. The file named “`correct_pl_true_output.txt`” contains the output one should get if the code for `pl_true` is completed correctly. There are comments in the `pl_true` function that are intended to provide hints and guidance about what your inserted code should accomplish. In each place where the

```
<put the correct programming statement here.>
```

text appears, only one or two lines of code is needed ... assuming you understand what needs to be done at each location in the code.

## INSTRUCTIONS:

Consider the problem of deciding whether a propositional logic sentence is true in a given model. Write a recursive algorithm `pl_true(s, m)` that returns *True* if and only if the sentence *s* is true in the model *m* (where *m* assigns a truth value for every symbol in *s*).

Completing this assignment will require you to read and understand sections 7.1 to and including 7.4 in the required textbook. After this programming assignment is completed, you will have good foundational knowledge and experience with propositional logic representation of knowledge. You will also begin to obtain a good sense of how the surrounding code works, how inferencing can be carried out, and so forth.

You will complete the assignment by modifying an extending the function named `pl_true` that is located in the Python file named `logic.py`. The `pl_true` function has a statement at the beginning that describes the overall purpose of the function. There are comments liberally placed in the function that described what the inserted code should accomplish. This is indicated by the text

```
"<put the correct programming statement here.>"
```

Each place where the above text appears, you are expected to remove that comment and insert your code that accomplishes what the preceding comments indicate what needs to be accomplished.

Between reading the textbook, instruction provided in lecture, and reviewing the code, and your understanding of logic you should be able to complete this assignment.

If your code for `pl_true` is correct, running the driver for the `logic.py` code should produce exactly the same output as contained in a file named `correct_pl_true_output.txt`.

To begin working on this assignment, follow the steps below.

1. Download the “CS156\_Programming\_Assignment\_1.zip” file that contains all of the Python 3.9 compatible code and accompanying .txt files from the Canvas -> Module 3 -> Week 6 section on Canvas. The main driver module is named `logic_driver.py`. The `pl_true` function is located in the module named `logic.py`
2. Before making any modifications to any code, run the `logic_driver.py` module and compare the output with that displayed in the `correct_pl_true_output.txt` file. Note the differences. It is unlikely you will understand the difference unless you have read sections 7.1 to 7.4 in the textbook and attended the lectures on this topic.
3. To begin understanding how the code works, trace the execution of the code from the calls to various functions starting with the `logic_driver.py` module. Some will call `pl_true` while others will not. Match the execution thread you discover by tracing the execution with how proposition logic works, conversion of logical statements such as  $A \Rightarrow B$  to conjunctive normal form such as  $\sim A$  or  $B$ , and so forth. Match what you see in the code with the discussion and your understanding from reading sections 7.1 to 7.4.
4. Then write down, in pseudocode, what you need to do to modify `pl_true`. Then incrementally make modifications to the function and run it to check its output with that displayed in `correct_pl_true_output.txt`. Modify, correct, update, and extend as needed.

Do not spin your wheels too long if and when you are uncertain. The instructor can provide hints or suggestions as appropriate to help you figure out the answer on your own.

Best of luck.