**Aim:** Write a python program using durable-rules module with forward-chaining rules for

course-and -extracurricular activities suggestion system for a non-graduating student based on grades and interests. Make your own rules and test it out with facts.

**Requirements:**

1. Installing durable-rules module in python.
2. Jupyter notebook.

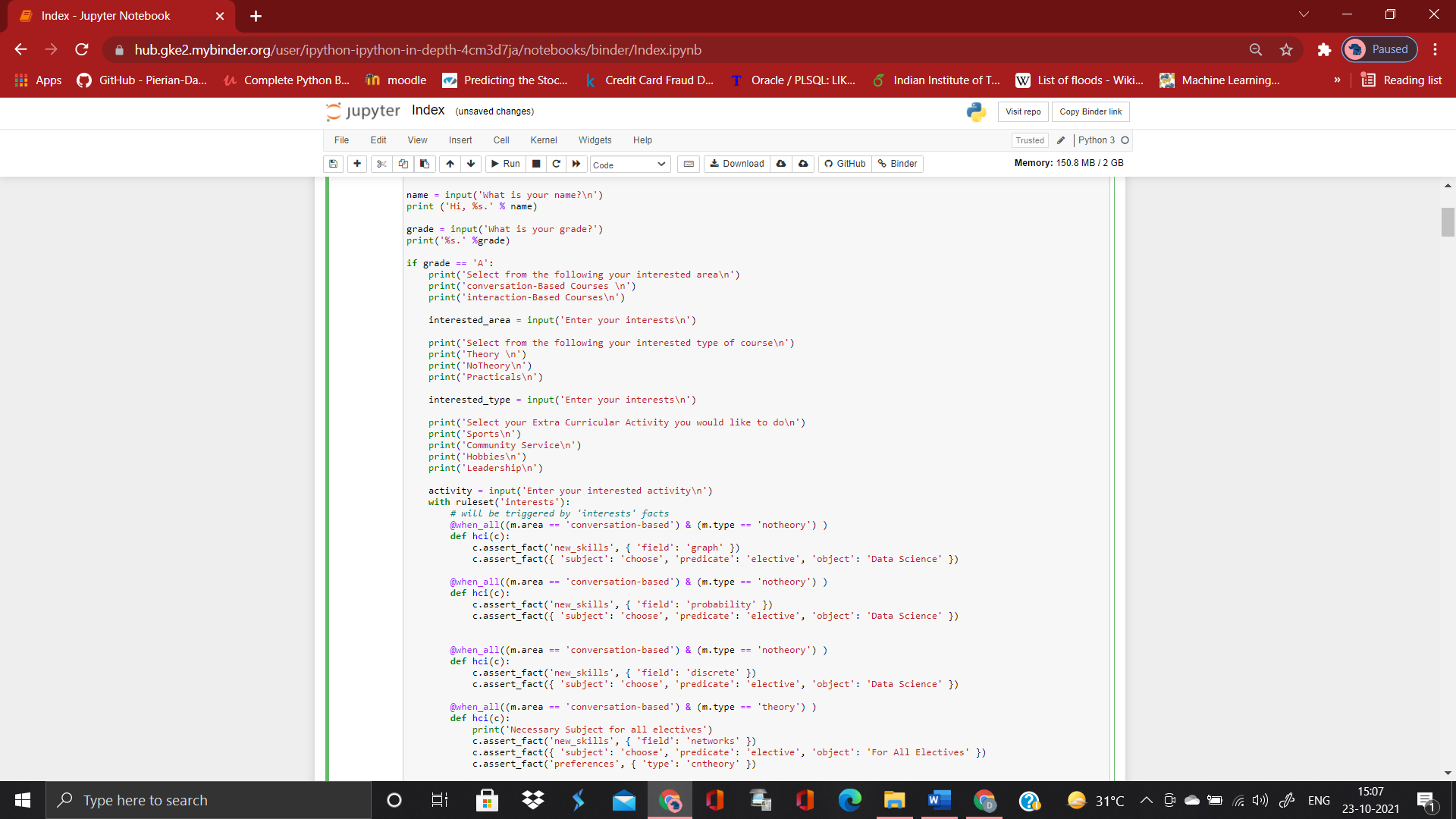
**Background:** Forward Chaining. (RETE Algorithm)

**About the code:**

(The code has been referenced from the given example by sir!)

(The definitions have been taken from internet).

1. Using the durable-rules module for applying forward chaining.
2. Using a full forward chaining implementation (A.K.A. Rete).
3. To define a rule, all you need to do is describe the event or fact pattern to match (antecedent) and the action to take (consequent).
4. Defining a ruleset for interests and activities of the student. The interests is divided into two area that can be conversation-based or interaction-based and type that are theory, notheory and practical. These are further divided into fields so that we can suggest a person what he/she wants to select. The activities is divided into several fields that the institute offers and a student is recommended according to his/her needs.
5. Inputting the following:
6. Name of the student.
7. Grade of the student.
8. Interested area of course.
9. Interested field of course.
10. Interested type of course (theory, notheory, practical based)
11. Interested activity.
12. According to the grades the questions will be put up against a candidate and he/she has to select the choice.



For grade ‘A’ all the choices are open, for grade ‘B’ all choices are open just number of subjects are less, for grade ‘C’ a student can not take notheory course for any grade less than ‘C’ a student can only take theory course.

1. Recommending a student which subject he/she must choose and also what extracurricular activities can be taken up.

**Code Explanation:**

1. Defining the ruleset for interest first where we have two categories
2. area 🡪 conversation-based

* interaction-based

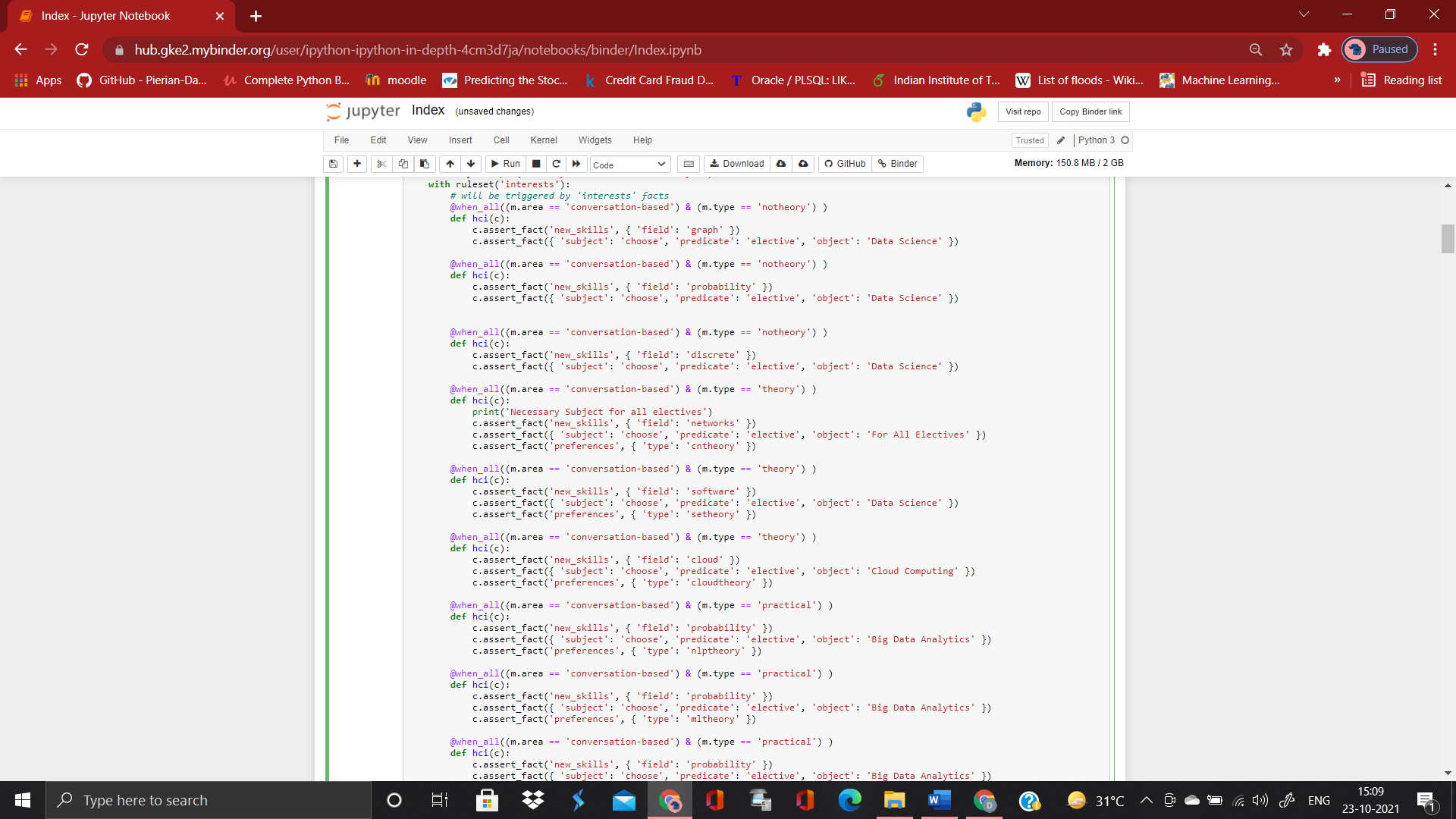
1. type -> notheory

* theory
* practical based subjects

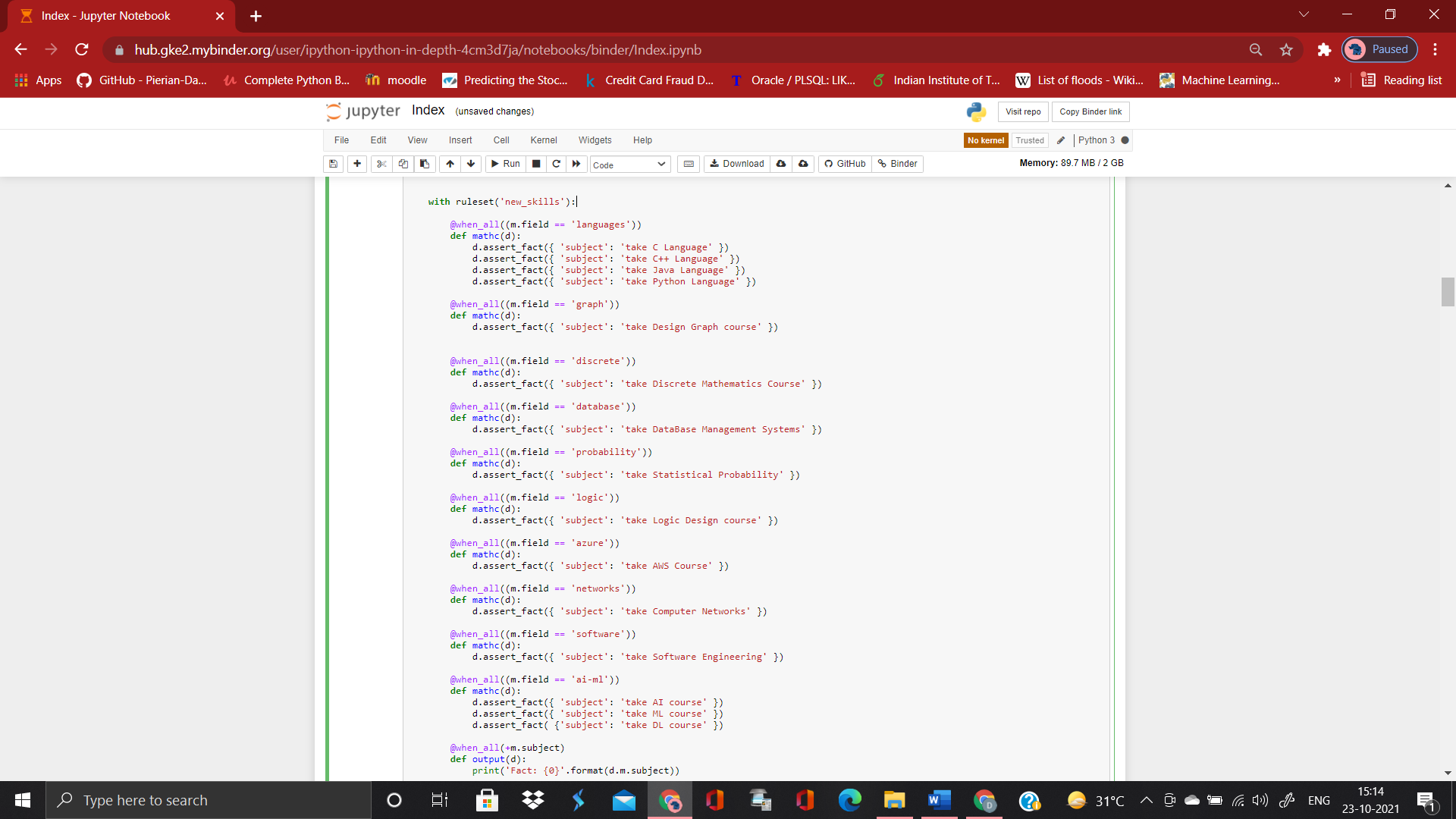
1. In assert\_facts we have new\_skills ruleset which is defined later and field is defined.

According to this the suggestion is given. It’s all chaining as they are connected to each other.

For theory type I have defined a ruleset preference which suggests the theory course.



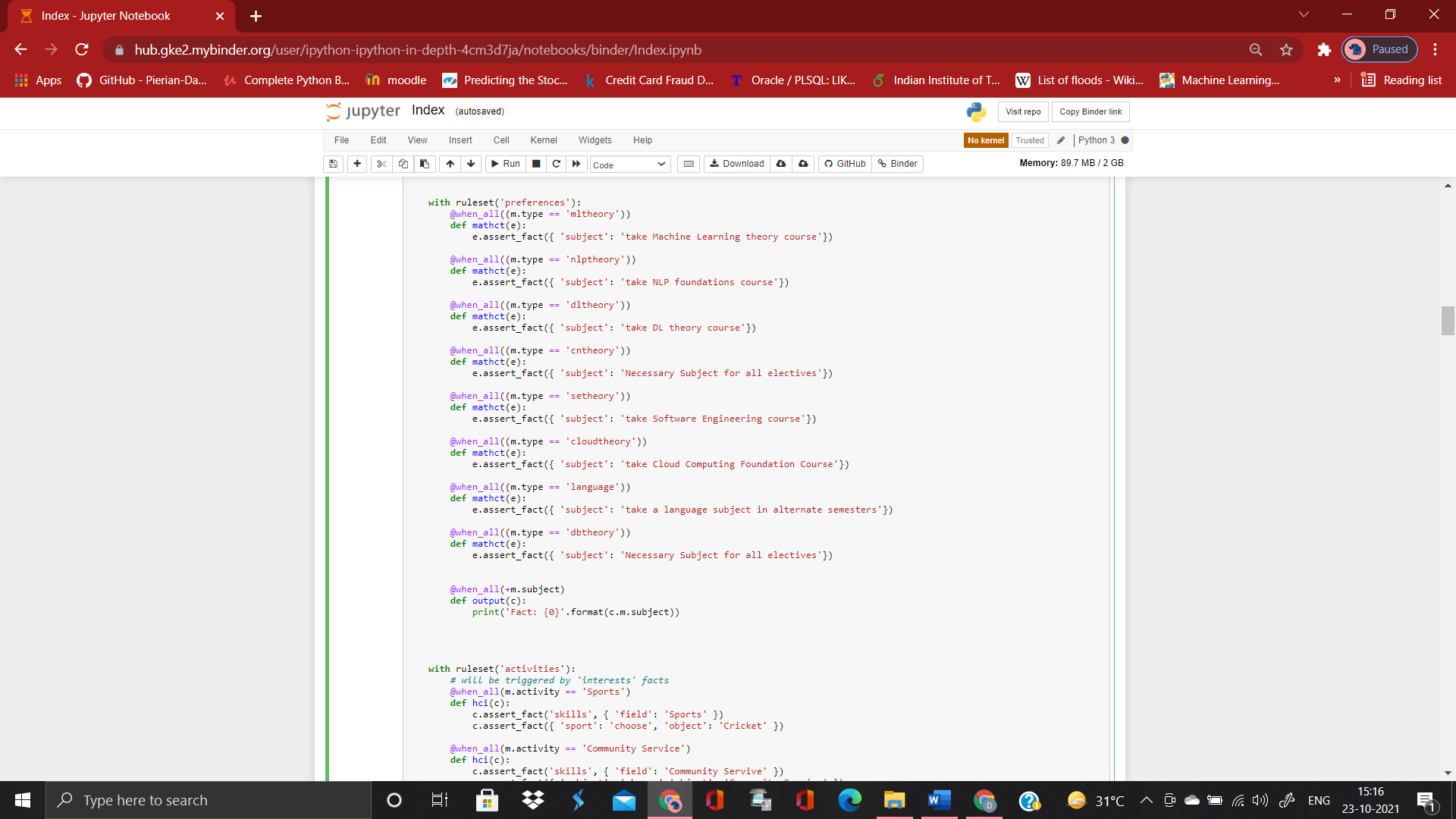
1. This shows ruleset new\_skills, where according to the filed given above I have specified the facts what course a student can select.



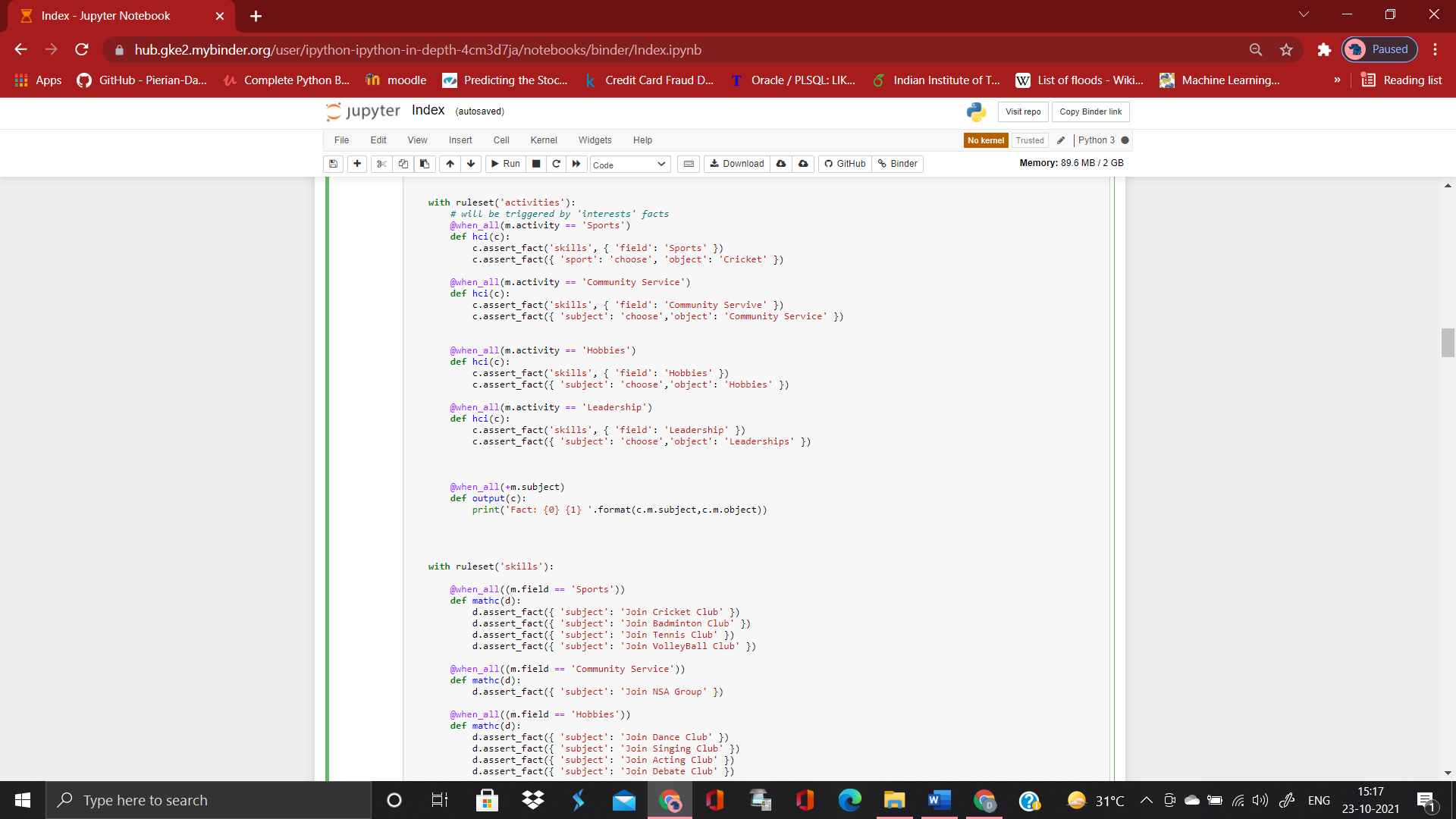
Outputting the facts.

1. This is ruleset preferences.

Suggesting the type of theory course for a student.



1. A student also inputs his/her interest in activities for which a new ruleset activities has been made.

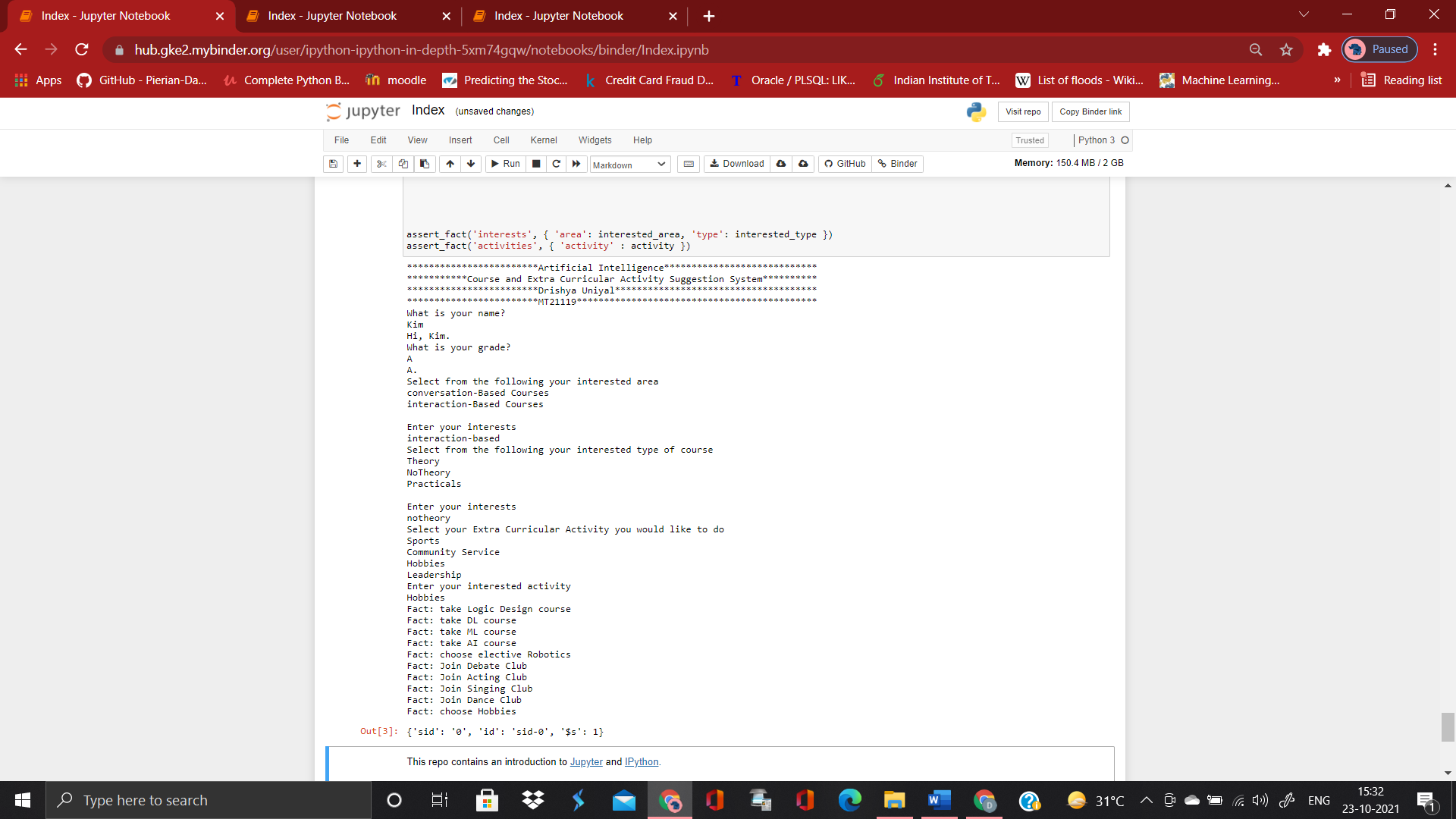


In this the activity can be classified as Sports, community Service, Leadership and Hobbies and accordingly a student can select the interest.

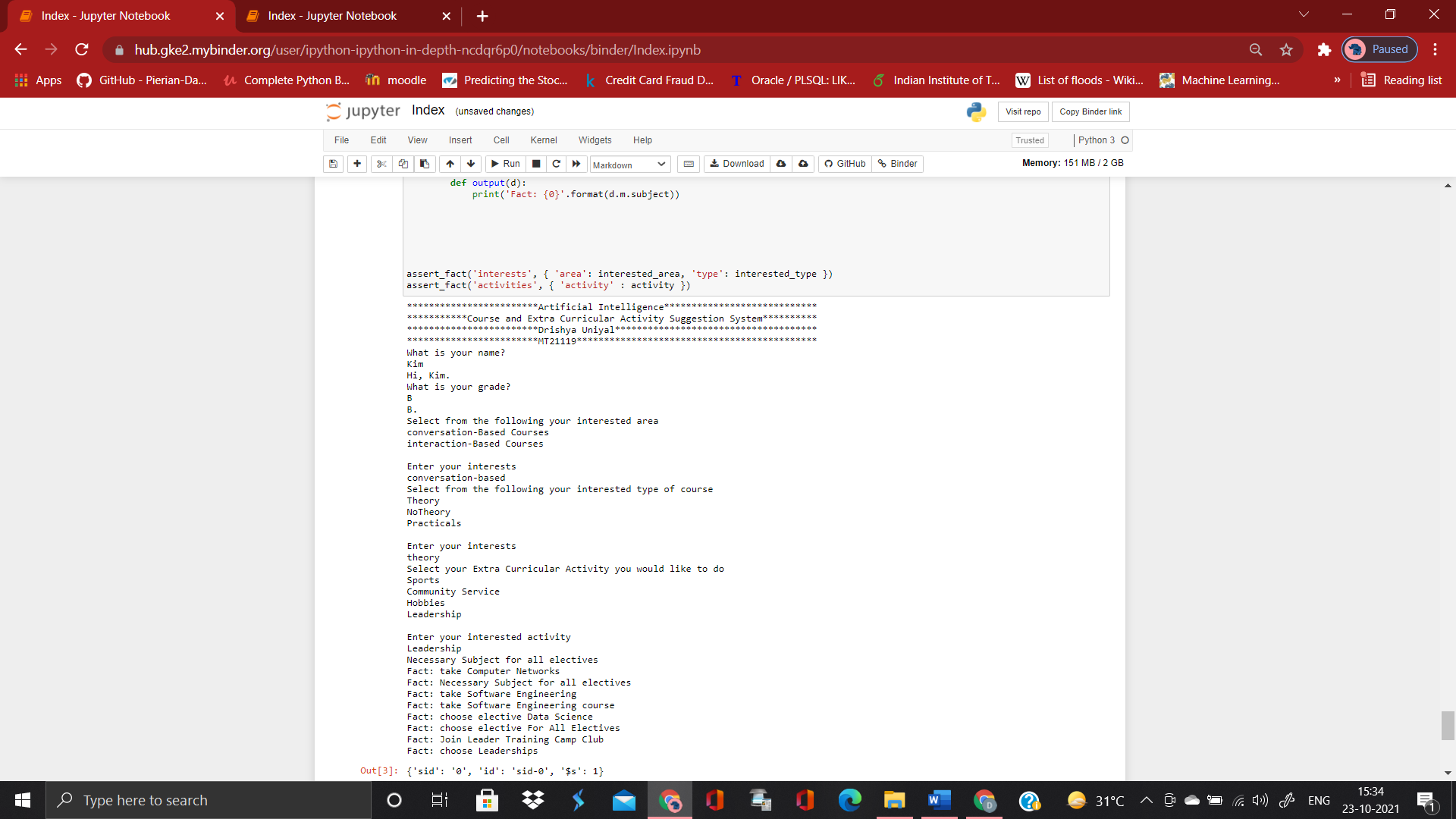
1. This was the basic code for a grade ‘A’ student. The code remains same for other grades just the categories change as explained earlier. The grades have been altered using the if-else conditions in python.

**Output:**

1. For a student having grade ‘A’

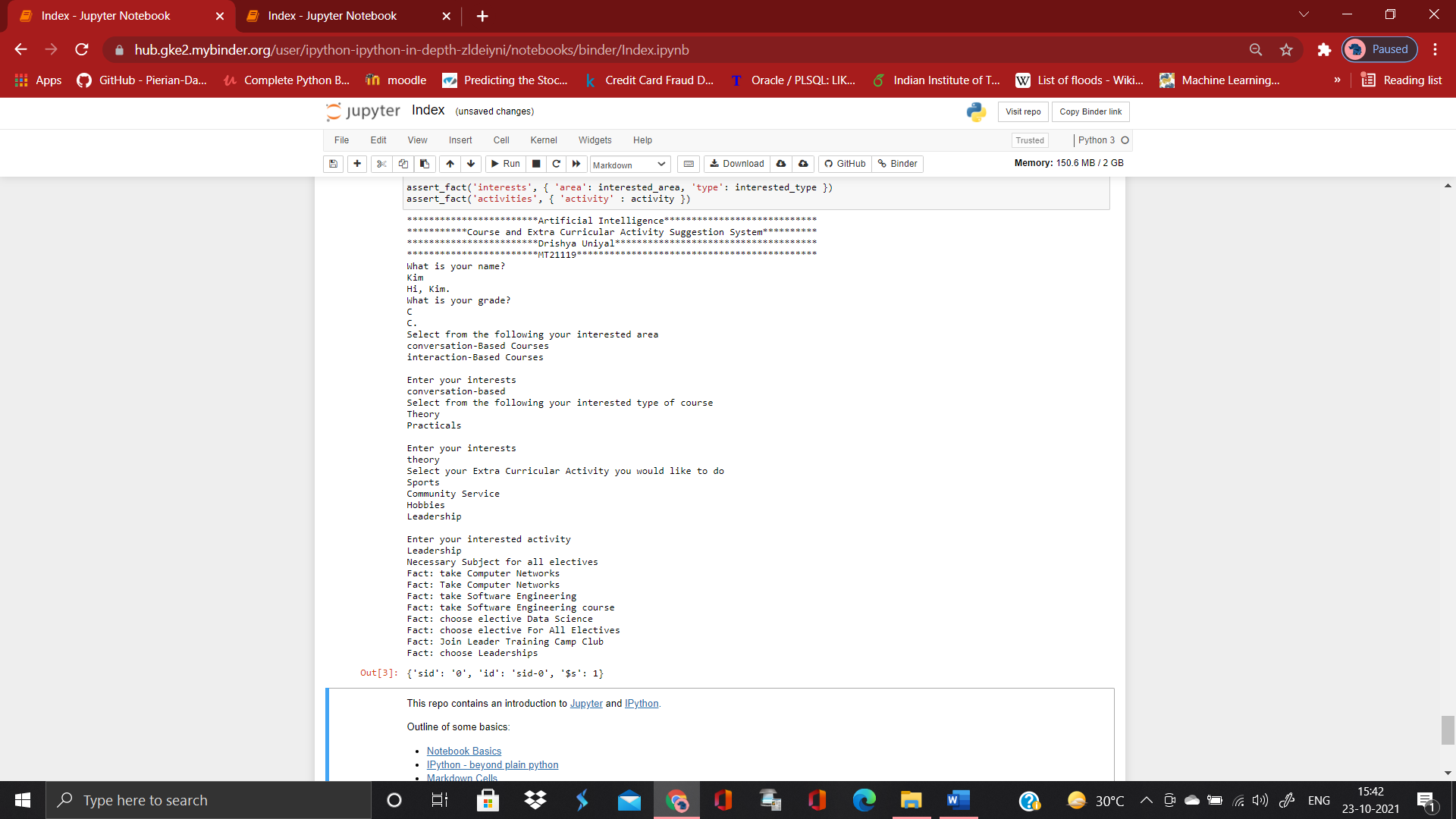


1. For a student having grade ‘B’



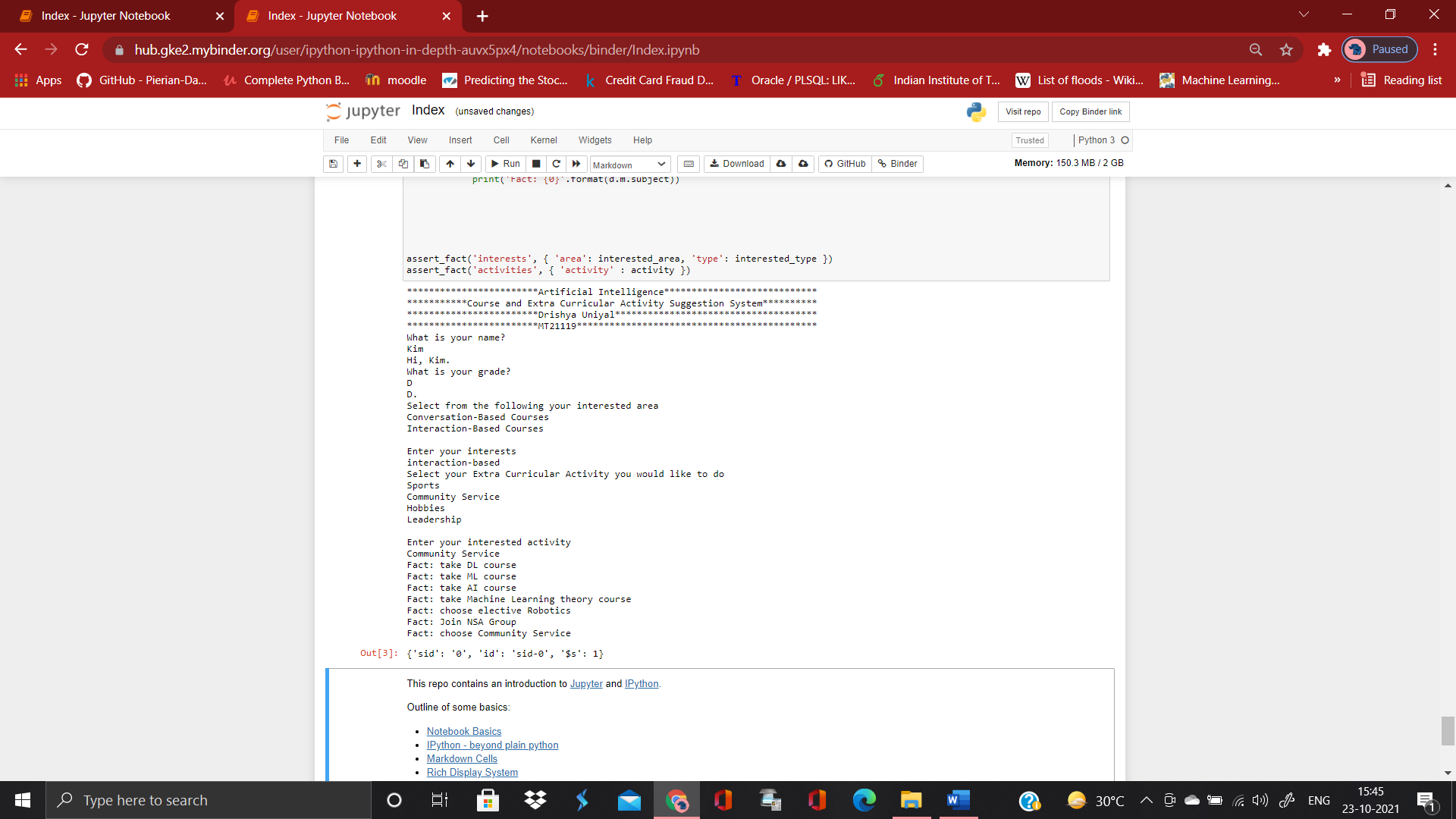
1. For a student having grade ‘C’

For a student having ‘C’ grade it can be seen that only theory and practical option was given.



1. For a student having grade less than ‘C’

For a student having any grade less than ‘C’ the choices are restricted and he/ she is not asked to take theory, practical, notheory and by default only theory is there.



**Results:**

According to grades and interests the courses vary.

The extra-curricular activities remain same for everyone.

**Running the code:**

The code is run on online Jupyter notebook where we have to first do !pip install durable-rules and then run the code. This has to be done each time we run the code otherwise Jupyter notebook throws an error that the ruleset has already been defined.