AI Assignment-3

Abhinav Saurabh (MT20127)

Screenshots:

Interest is data-scientist and Grade is 7+

```
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 % python soln3.py
Fact: Take DL course
Fact: Take ML course
Fact: Take DMG course
Fact: Probability and Statistics course
Fact: Participate in Academic Induction Activities
Fact: Participate in Student Council
Fact: Participate in Student senate
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 %
```

Interest is software engineer and Grade is 7+

```
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 % python soln3.py
Fact: Python programming
Fact: Web Development
Fact: Data Structure and Algorithms
Fact: Participate in Academic Induction Activities
Fact: Participate in Student Council
Fact: Participate in Student senate
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 %
```

Interest is ML engineer and Grade is 7+

```
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 % python soln3.py
Fact: take DL course
Fact: take ML course
Fact: take AI course
Fact: Participate in Academic Induction Activities
Fact: Participate in Student Council
Fact: Participate in Student senate
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 %
```

Interest is data-scientist and Grade is 7-

```
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 % python soln3.py
Fact: Take DL course
Fact: Take ML course
Fact: Take DMG course
Fact: Probability and Statistics course
Fact: Participate in few Club events
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 %
```

Interest is software engineer and Grade is 7-

```
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 % python soln3.py
Fact: Python programming
Fact: Web Development
Fact: Data Structure and Algorithms
Fact: Participate in few Club events
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 %
```

Interest is ML engineer and Grade is 7-

```
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 % python soln3.py
Fact: take DL course
Fact: take ML course
Fact: take AI course
Fact: Participate in few Club events
(base) abhinavsaurabh@Abhinavs-MacBook-Pro Assignment 3 %
```

I have implemented forwarding chaining in taking interest and their and accordingly given then the electives and extra-curricular activities. If they have less than 7 cgpa then fewer extra-curricular activities should be taken into consideration else they can do a lot of activities.

Codes

```
#Here we are importing the durable lang

#here imported the durang library
from durable.lang import *
#Further you can do the program with interests ,electives and extracurricular activities


#Here rules are made according to the interests and grades
with ruleset('interests'):
    # will be triggered by 'interests' facts
#This is for the data-scientist with 7+ cgpa
    @when_all((m.area == 'data-scientist') & (m.grades == '7+') )
    def hci(c):
        c.assert_fact('electives', { 'field': 'data' })
        c.assert_fact('extra-curricular', { 'type': 'heavy' })

#This is for the software engineer with 7+ cgpa
    @when_all((m.area == 'software-engineer') & (m.grades == '7+') )
    def rob(c):
        c.assert_fact('electives', { 'field': 'algorithms' })
        c.assert_fact('extra-curricular', { 'type': 'heavy' })

#This is for the ML-engineer with 7+ cgpa
    @when_all((m.area == 'ML-engineer') & (m.grades == '7+') )
    def rob(c):
        c.assert_fact('electives', { 'field': 'ai-ml' })
        c.assert_fact('extra-curricular', { 'type': 'heavy' })

#This is for the data-scientist with 7- cgpa
    @when_all((m.area == 'data-scientist') & (m.grades == '7-') )
    def hci(c):
        c.assert_fact('electives', { 'field': 'data' })
        c.assert_fact('extra-curricular', { 'type': 'little' })

#This is for the software-engineer with 7- cgpa
    @when_all((m.area == 'software-engineer') & (m.grades == '7-') )
    def rob(c):
        c.assert_fact('electives', { 'field': 'algorithms' })
        c.assert_fact('extra-curricular', { 'type': 'little' })

#This is for the ML-engineer with 7- cgpa
    @when_all((m.area == 'ML-engineer') & (m.grades == '7-') )
    def rob(c):
        c.assert_fact('electives', { 'field': 'ai-ml' })
        c.assert_fact('extra-curricular', { 'type': 'little' })


    @when_all(+m.subject)
    def output(c):
        print('Fact: {0} {1} {2}'.format(c.m.subject, c.m.predicate, c.m.object))
```

```
#Here are rules for the elective selection.
with ruleset('electives'):
#these electives  are for the data scientist consideration.
    @when_all((m.field == 'data'))
    def data(d):
        d.assert_fact({ 'subject': 'Probability and Statistics course' })
        d.assert_fact({ 'subject': 'Take DMG course' })
        d.assert_fact({ 'subject': 'Take ML course' })
        d.assert_fact({ 'subject': 'Take DL course' })

#these electives  are for the ML Engineer consideration.
    @when_all((m.field == 'ai-ml'))
    def ml(d):
        d.assert_fact({ 'subject': 'take AI course' })
        d.assert_fact({ 'subject': 'take ML course' })
        d.assert_fact({'subject': 'take DL course' })

#these electives  are for the ML Engineer consideration.
    @when_all((m.field == 'algorithms'))
    def algorithm(d):
        d.assert_fact({ 'subject': 'Data Structure and Algorithms' })
        d.assert_fact({ 'subject': 'Web Development' })
        d.assert_fact({ 'subject': 'Python programming' })



    @when_all(+m.subject)
    def output(d):
        print('Fact: {0}'.format(d.m.subject))

#Here are rules for the Extra-Curricular Activities
with ruleset('extra-curricular'):
#These are extra curricular activities for students with greater than 7 cgpa
    @when_all((m.type == 'heavy'))
    def ec(e):
        e.assert_fact({ 'subject': 'Participate in Student senate'})
        e.assert_fact({ 'subject': 'Participate in Student Council'})
        e.assert_fact({ 'subject': 'Participate in Academic Induction Activities'})

#These are extra curricular activities for students with less than 7 cgpa
    @when_all((m.type == 'little'))
    def ec(e):
        e.assert_fact({ 'subject': 'Participate in few Club events'})



    @when_all(+m.subject)
    def output(c):
        print('Fact: {0}'.format(c.m.subject))

```