Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Final Year B. Tech (EE)

Trimester: VII **Subject: AIML**

Name: Shreerang Mhatre

Class: Final Year B. Tech (EE)

Roll No: 52

Batch: A3

A3

Experiment No: 03

Name of the Experiment: Vacuum Cleaner World Agent using Python

Performed on: 11/09/2023

Submitted on: 11/09/2023

Marks	Teacher's Signature withdate

Aim: To develope a simple reflex agent program in Python for the vacuum-cleaner world problem.

Prerequisite: Knowledge of Agents.

Objective:

To create a Vacuum Cleaner World Agent using Python Programming.

Components and Equipment required:Python

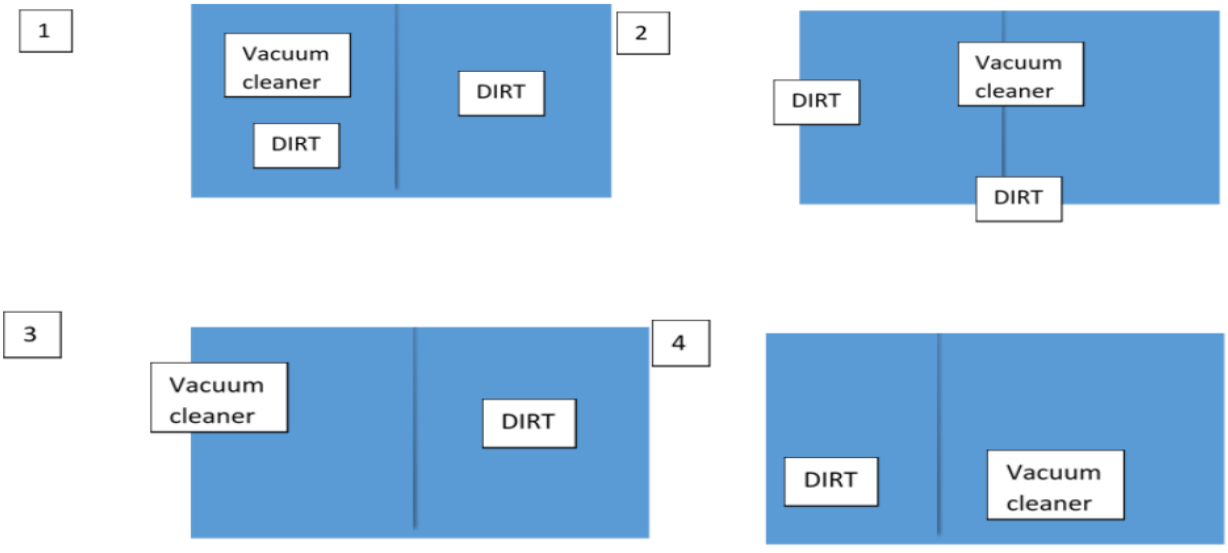
Theory:

Vacuum cleaner problem is a well-known search problem for an agent which works on ArtificialIntelligence. In this problem, our vacuum cleaner is our agent. It is a goal based agent, and the goal of this agent, which is the vacuum cleaner, is to clean up the whole area. So, in the classical vacuum cleaner problem, we have two rooms and one vacuum cleaner. There is dirt in both the roomsand it is to be cleaned. The vacuum cleaner is present in any one of these rooms. So, we have to reacha state in which both the rooms are clean and are dust free.

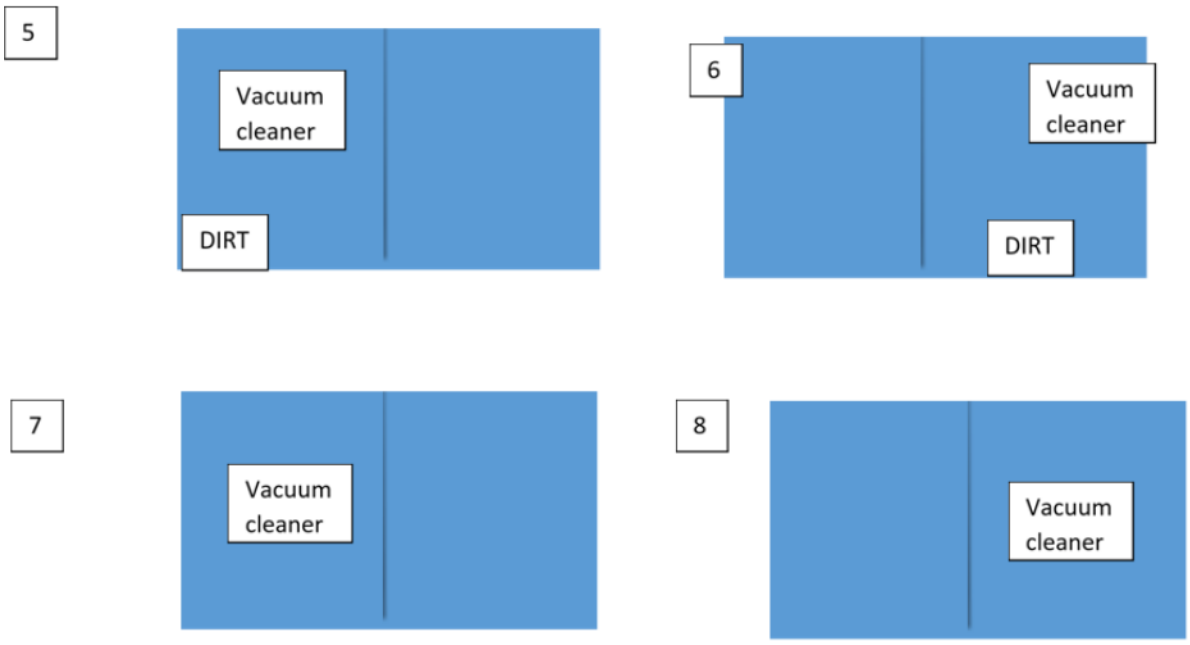
This program defines the States, Goal State, Goal Test, Actions, Transition Model, and Path Cost. Foreach possible initial state, the program returns a sequence of actions that leads to the goal state, along with the path cost. Generates two test cases.

1. **EnterLOCATIONA/BincaptialletterswhereAandBarethetwoadjacentrooms respectively.**
2. **EnterStatusO/1accordinglywhere0meansCLEANand1meansDIRTY.**
3. **VacuumCleanersensesthestatusoftheotherroombeforeperforminganyaction, also known as Environmentsensing.**

So,thereareeightpossiblestatespossibleinourvacuumcleanerproblem**.Thesecanbewellillustratedwiththehelpofthefollowingdiagrams:**



Exp. 3- 2



Here, states 1 and 2 are our initial states and state 7 and state 8 are our final states (goal states).This means that, initially, both the rooms are full of dirt and the **vacuum cleaner** can reside in any room. And to reach the final goal state, both the rooms should be clean and the **vacuum cleaner** again can reside in any of the two rooms.

The **vacuum cleaner** can perform the following functions: move left, move right, move forward,move backward and to suck dust. But as there are only two rooms in our problem, the vacuum cleaner performs only the following functions here: move left, move right and suck.

Here the performance of our agent (vacuum cleaner) depends upon many factors such as timetaken in cleaning, the path followed in cleaning, the number of moves the agent takes intotal,etc. But we consider two main factors for estimating the performance of the agent. Theyare:

1. **Search Cost:** How long the agent takes to come up with thesolution.
2. **Path cost:** How expensive each action in the solutionare.

By considering the above factors, the agent can also be classifies as a utility based agent.

Procedure:

- Create an Environment and Vacuum Cleaner Agent
- Select random location and random status for Vacuum Cleaner Agent
- Verify Initial Location and Environment Status.
- Achieve the goal state.

Programme Code

```
import random
class Environment(object):
   definit(self): self.locationcondition={'A'
      : '1' , 'B' :'1'} #RANDOM
CONDITION
      self.locationcondition['A']=random.randint(0,1)
      self.locationcondition['B']=random.randint(0,1)


class Sreflexagent(Environment):
   definit(self,Environment):
      #print(Environment.locationcondition)#place
      vacum at random location
      vacuumlocation=random.randint(0, 1) #if
      vacuum at A
      if vacuumlocation==0:
         print("vacuum is randomly placed at locationn A")#and if
         location A is dirty
         if Environment.locationcondition['A']==1:
            print("Location A is dirty")
            #suck the dirt and mark it clean
            Environment.locationcondition['A']=0
            print("Location A has been cleaned")
            print("moving to location B")
            vacuumlocation=1
         else:
            print("Location A is clean")#move
         to B
            print("moving to location B")
            vacuumlocation=1
```

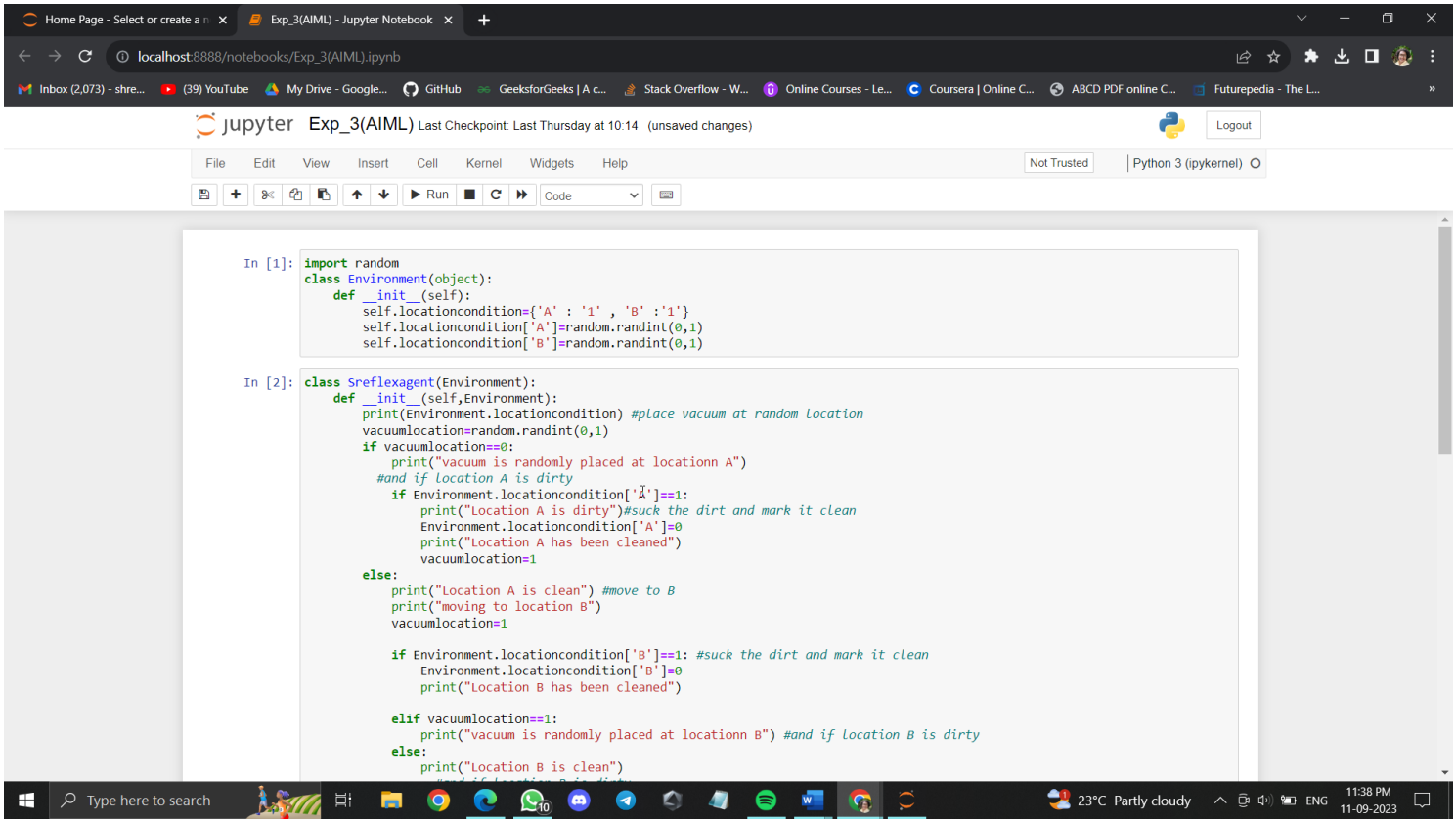
```
if
        Environment.locationcondition['B']==1
        : #suck the dirt and mark itclean
          Environment.locationcondition['B']=0
          print("Location B has been cleaned")
      else:
          print("Location B is clean")elif
    vacuumlocation==1:
      print("vacuum is randomly placed at locationn B")#and if
      location B is dirty
      if Environment.locationcondition['B']==1:
        print("Location B is dirty")
        #suck the dirt and mark it clean
        Environment.locationcondition['B']=0;
        print("Location B has been cleaned")
        print("moving to location A")
        vacuumlocation=0
      else:
        print("Location B is Clean")#Move
        to A
        print("moving to location A")
        vacuumlocation=0
      if vacuumlocation==0:
        if Environment.locationcondition['A']==1:
          print("Location A is dirty")
        #suck the dirt and mark it clean
          Environment.locationcondition['A']=0;
          print("Location A has been cleaned")
        else:
          print("Location A is Clean")



#DONE CLENING

#print(Environment.locationcondition)
theEnvironment=Environment()
thevacuum=Sreflexagent(theEnvironment)
```

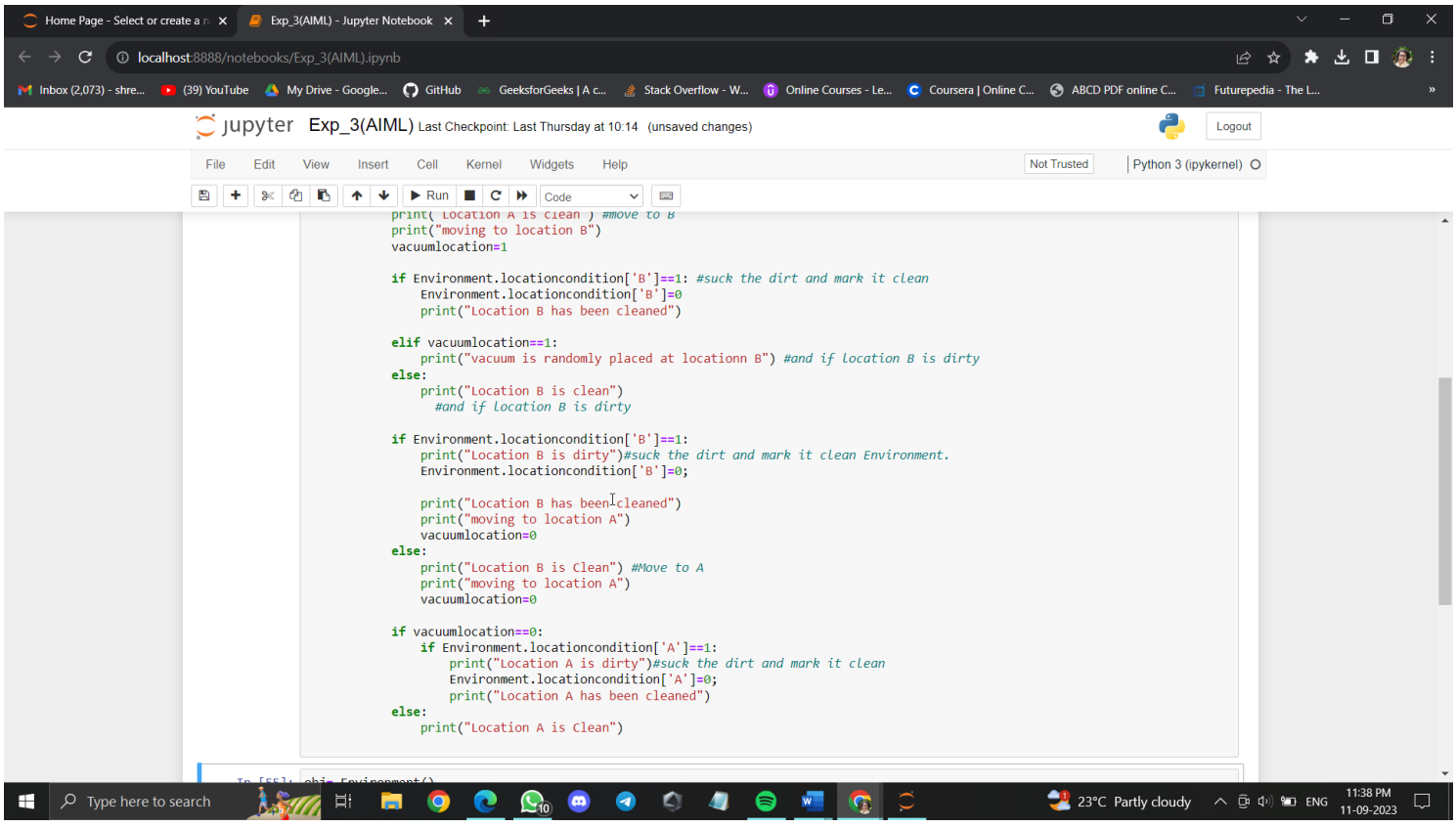
Conclusion:

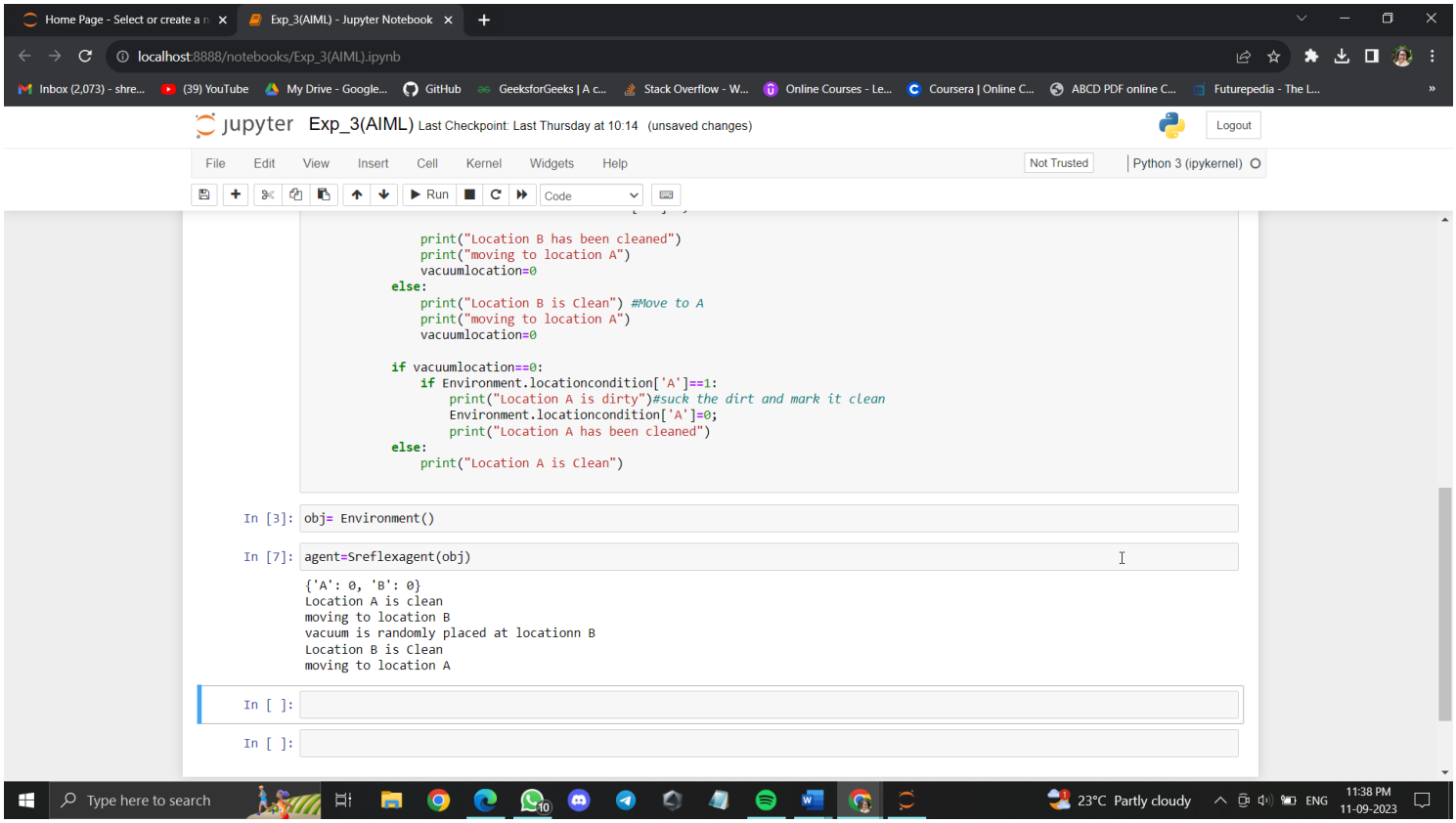
A vacuum cleaner world agent is developed for two locations considering randomstatus using Python Programming.

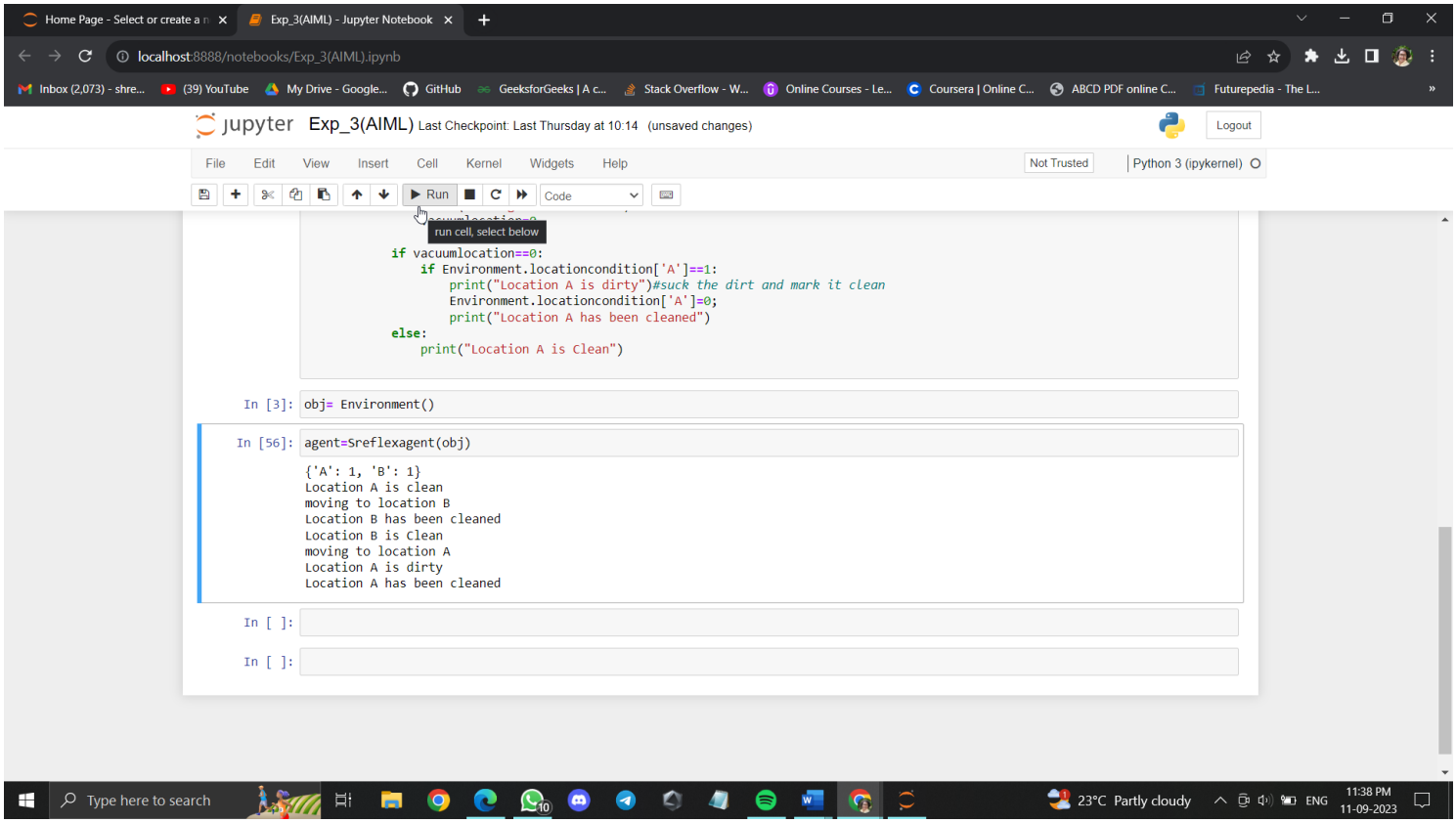
Post Lab Questions:

1. What are the different types of agents? Explain the pseudocode forutility based agent.
2. Explain different terms related to agent.
3. what are the different types of environment?


Home Page - Select or create a n
Exp_3(AIML) - Jupyter Notebook
localhost:8888/notebooks/Exp_3(AIML).ipynb
jupyter Exp_3(AIML) Last Checkpoint: Last Thursday at 10:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [1]:
import random
class Environment(object):
def __init__(self):
self.locationcondition={'A' : '1' , 'B' :'1'}
self.locationcondition['A']=random.randint(0,1)
self.locationcondition['B']=random.randint(0,1)
In [2]:
class Sreflexagent(Environment):
def __init__(self,Environment):
print(Environment.locationcondition) #place vacuum at random location
vacuumlocation=random.randint(0,1)
if vacuumlocation==0:
print("vacuum is randomly placed at locationn A")
#and if location A is dirty
if Environment.locationcondition['A']==1:
print("Location A is dirty")#suck the dirt and mark it clean
Environment.locationcondition['A']=0
print("Location A has been cleaned")
vacuumlocation=1
else:
print("Location A is clean") #move to B
print("moving to location B")
vacuumlocation=1
if Environment.locationcondition['B']==1: #suck the dirt and mark it clean
Environment.locationcondition['B']=0
print("Location B has been cleaned")
elif vacuumlocation==1:
print("vacuum is randomly placed at locationn B") #and if location B is dirty
else:
print("Location B is clean")
Type here to search
23°C Partly cloudy
ENG
11:38 PM
11-09-2023


Home Page - Select or create a n
Exp_3(AIML) - Jupyter Notebook
localhost:8888/notebooks/Exp_3(AIML).ipynb
jupyter Exp_3(AIML) Last Checkpoint: Last Thursday at 10:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
print("moving to location B")
vacuumlocation=1
if Environment.locationcondition['B']==1: #suck the dirt and mark it clean
Environment.locationcondition['B']=0
print("Location B has been cleaned")
elif vacuumlocation==1:
print("vacuum is randomly placed at locationn B") #and if location B is dirty
else:
print("Location B is clean")
#and if location B is dirty
if Environment.locationcondition['B']==1:
print("Location B is dirty")#suck the dirt and mark it clean Environment.
Environment.locationcondition['B']=0;
print("Location B has been cleaned")
print("moving to location A")
vacuumlocation=0
else:
print("Location B is Clean") #Move to A
print("moving to location A")
vacuumlocation=0
if vacuumlocation==0:
if Environment.locationcondition['A']==1:
print("Location A is dirty")#suck the dirt and mark it clean
Environment.locationcondition['A']=0;
print("Location A has been cleaned")
else:
print("Location A is Clean")
Type here to search
23°C Partly cloudy
ENG
11:38 PM
11-09-2023


Home Page - Select or create a n
Exp_3(AIML) - Jupyter Notebook
localhost:8888/notebooks/Exp_3(AIML).ipynb
jupyter Exp_3(AIML) Last Checkpoint: Last Thursday at 10:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
print("Location B has been cleaned")
print("moving to location A")
vacuumlocation=0
else:
print("Location B is Clean") #Move to A
print("moving to location A")
vacuumlocation=0
if vacuumlocation==0:
if Environment.locationcondition['A']==1:
print("Location A is dirty")#suck the dirt and mark it clean
Environment.locationcondition['A']=0;
print("Location A has been cleaned")
else:
print("Location A is Clean")
In [3]: obj= Environment()
In [7]: agent=Sreflexagent(obj)
{'A': 0, 'B': 0}
Location A is clean
moving to location B
vacuum is randomly placed at locationn B
Location B is Clean
moving to location A
In [ ]:
In [ ]:
Type here to search
23°C Partly cloudy
ENG
11:38 PM
11-09-2023


Home Page - Select or create a n
Exp_3(AIML) - Jupyter Notebook
localhost:8888/notebooks/Exp_3(AIML).ipynb
jupyter Exp_3(AIML) Last Checkpoint: Last Thursday at 10:14 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
run cell, select below
if vacuumlocation==0:
if Environment.locationcondition['A']==1:
print("Location A is dirty")#suck the dirt and mark it clean
Environment.locationcondition['A']=0;
print("Location A has been cleaned")
else:
print("Location A is Clean")
In [3]: obj= Environment()
In [56]: agent=Sreflexagent(obj)
{'A': 1, 'B': 1}
Location A is clean
moving to location B
Location B has been cleaned
Location B is Clean
moving to location A
Location A is dirty
Location A has been cleaned
In [ ]:
In [ ]:
Type here to search
23°C Partly cloudy
ENG
11:38 PM
11-09-2023

PAGE No.
DATE 11/9/23

Exp 3 Vaccum Cleaner Agent using Python

* Past lab questions.

Q1) what are the different types of agents? Explain the pseudocode for utility based agent.

→ Types of agents are:
1. Simple Reflex agent
2. Model-based reflex agent
3. Goal based agents
4. Utility based agents

*Utility based agent

Define a set of possible actions (A)
Define a set of possible states or world states (s)
Define a utility function U(a,s) for each action a & state s
Repeat until a terminal state is reached.

1. Observe the current state s-current.
2. For each action a in A:
   a) Calculate the expected utility EU(a) for taking action a in the current state:
   EU(a) = Σ (U(a, s-next) * p(s-next/a, s-current)), for all possible s-next in S.

PAGE No.
DATE / /

③ choose the action a-max with the highest expected utility.
a-max argman (EU(a) forall a in A)
④ Execute the chosen action a-max.
⑤ observe the resulting state s-next
⑥ Repeat the process with the new state s-next.
End loop when a terminal state is reached.

(Q2) Explain different terms related to agents

→① Agent -
An agent is a software or hardware entity that perceives its environment, processes information & takes actions to achieve specific goals or objectives

② Environment -
The environment is the external context or surroundings in which an agent operates & interacts. It encompasses everything outside the agent itself.

③ Perception -
Perception is the process through which an agent gathers information about its environment using sensors or other means. Basically it is the input to the agent.

PAGE No.
DATE / /

④ State -
The state represents the current condition or configuration of the environment at a given point in time

⑤ Agent function.
An agent function maps the agent's perception of the current state to an action or a sequence of actions.
- It defines how an agent behaves in response to different states of the environment.

⑥ Agent Program -
An agent program is the specific implementation or code that embodies the agent function. It determines how the agent operates & makes decisions.

⑦ Rational Agent -
A rational agent is an agent that consistently selects actions that maximize its expected utility, given its knowledge & beliefs about the environment.

Q3) what are the different types of environment?

→ ① Physical environment
② Biological environment
③ Climatic environment
④ Marine environment
⑤ Man-made environment
⑥ Social Environment
⑦ virtual environment
⑧ Extreme environment
⑨ Aquatic environment
⑩ Artificial environment.