Name: ID: Section:

1. Implement the “HumanoidRobot” class that is derived from the “Robot” class so that the following output is generated. [Hint: For each working hour 10% of battery level is decreased] **[Marks 10] [CO5]**

| class Robot:  robot\_count = 0  def \_\_init\_\_(self, id, name = 'No name'):  Robot.robot\_count+=1  self.name = name  self.id = id  self.battery = 100  print(f'------(Robot {Robot.robot\_count})------')  if self.name == 'No name':  print(f'Name: {self.name}\nID: {self.id}')  def battery\_level(self):  return f'Battery is {self.battery}% charged'  def activities(self):  print('No activities are mentioned')  r1 = Robot('RT101')  r1.activities()  hR1 = HumanoidRobot('RT101A', 'Sophia', '240VAC')  print('---------------------------')  print(hR1.battery\_level())  hR1.workingHours(4)  print('---------------------------')  print(hR1.activities('Care Giving', 'Personal Assistant'))  hR2 = HumanoidRobot('RT101B', 'Desdemona', '210VAC')  print('---------------------------')  print(hR2.battery\_level())  hR2.workingHours(7)  print('---------------------------')  print(hR2.activities('Space Exploration) | **Output**  ------(Robot 1)------  Name: No name  ID: RT101  No activities are mentioned  ------(Robot 2)------  \*\*\*Robot Sophia is initialized\*\*\*  Name: Sophia  ID: RT101A  Power: 240VAC  ---------------------------  Battery is 100% charged  Sophia has worked for 4 hours!!  Battery is 60% charged  ---------------------------  Activities of Robot 2:  {'Job 1': 'Care Giving', 'Job 2': 'Personal Assistant'}  ------(Robot 3)------  \*\*\*Robot Desdemona is initialized\*\*\*  Name: Desdemona  ID: RT101B  Power: 210VAC  ---------------------------  Battery is 100% charged  Desdemona has worked for 7 hours!!  Battery is 30% charged  ---------------------------  Activities of Robot 3:  {'Job 1': 'Space Exploration'} |
| --- | --- |

1. Trace the below table and write the outputs in the question paper.  **[Marks 10][CO4]**

| **1** | **class A:** |
| --- | --- |
| **2** | **temp = 3** |
| **3** | **def \_\_init\_\_(self):** |
| **4** | **self.y = A.temp - 6** |
| **5** | **self.sum = self.temp + 2** |
| **6** | **def methodA(self, m, n):** |
| **7** | **self.y = self.y + m + (A.temp)** |
| **8** | **self.sum = self.sum + self.temp** |
| **9** | **self.temp+=1** |
| **10** | **print(self.temp, self.y, self.sum)** |
| **11** | **class B(A):** |
| **12** | **temp = 1** |
| **13** | **def \_\_init\_\_(self, v1=None):** |
| **14** | **super().\_\_init\_\_()** |
| **15** | **self.temp = self.temp + B.temp** |
| **16** | **self.sum = 11 + B.temp + A.temp** |
| **17** | **if v1 != None:** |
| **18** | **v1.methodB(3, 9)** |
| **19** | **else:** |
| **20** | **self.methodB(1, -2)** |
| **21** | **def methodB(self, m, n):** |
| **22** | **y = self.temp + self.y + n** |
| **23** | **B.temp = m + self.y + n** |
| **24** | **self.methodA(n, m)** |
| **25** | **self.sum = self.y + A.temp** |
| **26** | **print(self.temp , y, self.sum)** |

| **Write the output of the following code:**  **b1 = B()**  **b2 = B(b1)** | Outputs | | |
| --- | --- | --- | --- |
|  |  |  |
|  |  | 1 |
|  |  |  |
| 4 |  | 13 |