Task 1:

| #Write your code here  #Driver code  s1= Student("Yellow", "King")  s2= Student("Black")  s3= Student()  print(s1.closet)  print(s2.closet)  print(s3.closet)  print(s1.bed)  print(s2.bed)  print(s3.bed)  x= s1.cost\_calc()  print(x)  y= s2.cost\_calc()  print(y)  z= s3.cost\_calc()  print(z)  s4= Student("Blue", "King", "Queen","Child")  print(s4.bed)  print(s4.cost\_calc()) | Output:  Yellow  Black  White  ('King',)  ()  ()  11000  5000  1000  ('King', 'Queen', 'Child')  23000 |
| --- | --- |

Task 2:

| #Write your code here  #Driver Code  amira= Amira()  amira.tempChange(30)  amira.existingTemp=40  print("New Temperature:",amira.existingTemp)  jawad= Amira()  jawad.tempChange(21)  jawad.existingTemp=12  print("New Temperature:",jawad.existingTemp)  print(amira)  print(amira.existingTemp)  print(jawad) | Output:  New Temperature: 30  New Temperature: 40  New Temperature: 21  New Temperature: 12  <\_\_main\_\_.Amira object at 0x7f05c7da8310>  40  <\_\_main\_\_.Amira object at 0x7f05c7da83d0> |
| --- | --- |

Task 3:

| #Write your code here  #Driver Code  #Declaring Object  flat\_2A= Building()  #Assigning values to the class variables  flat\_2A.temp= 80  flat\_2A.humidity= 0.5  flat\_2A.wall\_paint= "Yellow"  #Calling Setter Methods  flat\_2A.temperature\_increase()  flat\_2A.temperature\_deccrease()  flat\_2A.humidity\_change(0.35)  #Calling Getter Methods  print("Color of my walls are:", flat\_2A.what\_color\_are\_my\_walls())  # Tuple Unpacking  temp\_flat\_2A, humidity\_flat\_2A= flat\_2A.temperature\_humidity()  print("The temperature of Flat 2A is:", temp\_flat\_2A, "and the humidity is:",humidity\_flat\_2A) | Output:  New Temperature: 81  New Temperature: 80  New Humidity: 0.35  Color of my walls are: Yellow  The temperature of Flat 2A is: 80 and the humidity is: 0.35 |
| --- | --- |

Task 4:

| #Write your code here  #Driver Code  #Declaring Object  #Assigning values to the class variables while declaring objects  flat\_2A= Building(80,0.5,"Yellow")  flat\_2A.doorknob= True  # flat\_2A.temp= 80  # flat\_2A.humidity= 0.5  # flat\_2A.wall\_paint= "Yellow"  #Calling Setter Methods  flat\_2A.temperature\_increase()  flat\_2A.temperature\_deccrease()  flat\_2A.humidity\_change(0.35)  #Calling Getter Methods  print("Color of my walls are:", flat\_2A.what\_color\_are\_my\_walls())  temp\_flat\_2A, humidity\_flat\_2A= flat\_2A.temperature\_humidity()  print("The temperature of Flat 2A is:", temp\_flat\_2A, "and the humidity is:",humidity\_flat\_2A) | Output:  New Temperature: 81  New Temperature: 80  New Humidity: 0.35  Color of my walls are: 0.5  The temperature of Flat 2A is: 80 and the humidity is: 0.35 |
| --- | --- |

Task 5:

| #Write your code here  #Driver Code  #Declaring Object  #Assigning values to the class variables while declaring objects  flat\_2A= Building(80,0.5,"Yellow")  flat\_2A.doorknob= True  # flat\_2A.temp= 80  # flat\_2A.humidity= 0.5  # flat\_2A.wall\_paint= "Yellow"  #Calling Setter Methods  flat\_2A.temperature\_increase()  flat\_2A.temperature\_deccrease()  flat\_2A.humidity\_change(0.35)  #Calling Getter Methods  print("Color of my walls are:", flat\_2A.what\_color\_are\_my\_walls())  temp\_flat\_2A, humidity\_flat\_2A= flat\_2A.temperature\_humidity()  print("The temperature of Flat 2A is:", temp\_flat\_2A, "and the humidity is:",humidity\_flat\_2A) | Output:  New Temperature: 81  New Temperature: 80  New Humidity: 0.35  Color of my walls are: 0.5  The temperature of Flat 2A is: 80 and the humidity is: 0.35 |
| --- | --- |

Task 6:

| #Driver Code  std\_1= Student("Fuad", 2112, "CSE")  print("Student 1's ID:", std\_1.get\_id())  print("Student 1's Department:", std\_1.get\_dept())  print("Student 1's Name:", std\_1.get\_name())  std\_1.set\_id(2121)  std\_1.set\_name("Miftahul")  std\_1.set\_dept("BBA")  std\_2= Student("Rabbi", 2169, "CS")  print("Student 2's ID:", std\_2.get\_id())  print("Student 2's Department:", std\_2.get\_dept())  print("Student 2's Name:", std\_2.get\_name())  std\_2.set\_id(2420)  std\_2.set\_name("Tasnim")  std\_2.set\_dept("CSE") | #Output  ----------New Student Enrolled----------  Name: Fuad ID: 2112 Department: CSE  Student 1's ID: 2112  Student 1's Department: CSE  Student 1's Name: Fuad  Fuad's id changed to 2121  Fuad's name changed to Miftahul  Fuad's department changed to BBA  ----------New Student Enrolled----------  Name: Rabbi ID: 2169 Department: CS  Student 2's ID: 2169  Student 2's Department: CS  Student 2's Name: Rabbi  Rabbi's id changed to 2420  Rabbi's name changed to Tasnim  Rabbi's department changed to CSE |
| --- | --- |

Task 7:

Implement the design of the Student class so that the following output is produced:

Assume the credit for each course to be 3. For example: [3.3,4] can be calculated as:

CGPA = ((3.3 \* 3) + (4 \* 3)) / 6

[Here, for each course, the grade point is multiplied by 3. Total credit is the number of courses multiplied by 3. Since the example has 2 courses, therefore a total of 6 credits]

CGPA = sum of individual (grade point \* credit) / total credit

Academic Standing Rule: [CGPA>3.80 Highest Distinction, CGPA>3.65 High Distinction, CGPA>3.50 Distinction, CGPA>2.00 Satisfactory, CGPA<2.00 Can’t Graduate]

| # Write your code here  #Driver code  s1 = Student('Dora', '15995599','CSE', [4,3.7,3.7,4])  s1.calculate\_CGPA()  print("==========================")  s1.print\_details()  print("==========================")  s2 = Student('Pingu', '12312322', 'EEE', [1.7,1.3,1.3,1.3,1])  s2.calculate\_CGPA()  print("==========================")  s2.print\_details()  print("==========================")  s3 = Student('Bob', '13311331', 'CSE', [2,3,3,3.7,2.7,2.7])  s3.calculate\_CGPA()  print("==========================")  s3.print\_details() | Output:  ==========================  Name: Dora ID: 15995599 Department: CSE  CGPA: 3.85  Your academic standing is 'Highest Distinction'  ==========================  ==========================  Name: Pingu ID: 12312322 Department: EEE  CGPA: 1.3199999999999998  Sorry, you cannot graduate  ==========================  ==========================  Name: Bob ID: 13311331 Department: CSE  CGPA: 2.8499999999999996  Your academic standing is 'Satisfactory' |
| --- | --- |

Task 8:

| #Write your code here  #Driver code  d1= Dog("Bulldog", 1, "Black") #Breed,age,color  d1.age= 1.5  d1.change\_info("German Shepherd", 0.5, "White")  d1.print\_info()  d2= Dog("Golden Retriever", 2, "Yellow") #Breed,age,color  d2.print\_info()  d2.change\_info(1, "White") #Age, color  d2.print\_info()  d2.age=d1.age  print(d2.breed, "is", d2.age, "year(s) old.") | Output:  =======New Dog Brought!=======  Breed: German Shepherd , Age: 0.5 , Color: White  =======New Dog Brought!=======  Breed: Golden Retriever , Age: 2 , Color: Yellow  Breed: Golden Retriever , Age: 1 , Color: White  Golden Retriever is 0.5 year(s) old. |
| --- | --- |

Task 9:

| #Write your code here  #Driver code  n1= Number([5,7,1,-4,0,-6])  even, odd= n1.findEvenOdd()  print("Even Numbers:", even, "and Odd Numbers:", odd)  pos, neg= n1.findPosNeg()  print("Positive Numbers:", pos, "and Negative Numbers:", neg)  n2= Number()  even, odd= n2.findEvenOdd()  print("Even Numbers:", even, "and Odd Numbers:", odd)  pos, neg= n2.findPosNeg()  print("Positive Numbers:", pos, "and Negative Numbers:", neg) | Output:  Even Numbers: 3 and Odd Numbers: 3  Positive Numbers: 4 and Negative Numbers: 2  Even Numbers: 2 and Odd Numbers: 2  Positive Numbers: 4 and Negative Numbers: 0 |
| --- | --- |

Task 10:

| #Write your code here  #Driver Code  lc1= LetterChanger("Hello World")  new\_str= lc1.upperToLowerConverter()  print(new\_str)  new\_str= lc1.letterReplacer("h","m")  print(new\_str)  lc1.addStrings(" Kiddo! ", "You're gonna hate it.")  lc2= LetterChanger("Whasssup!!")  new\_str= lc2.upperToLowerConverter()  print(new\_str)  new\_str= lc2.letterReplacer("h")  print(new\_str)  new\_str= lc2.letterReplacer("s","k")  print(new\_str)  new\_str= lc2.letterReplacer()  lc2.addStrings(" Mid is coming! ") | Output:  ===================  New String Received  Upper to Lower Conversion Complete  hello world  Letter h replaced by m  mello world  Updated String after addition: mello world Kiddo! You're gonna hate it.  ===================  New String Received  Upper to Lower Conversion Complete  whasssup!!  Letter h replaced by #  w#asssup!!  Letter s replaced by k  w#akkkup!!  Nothing to replace  Updated String after addition: w#akkkup!! Mid is coming! |
| --- | --- |

Task 11:

Write the tracing output in a table

| class FinalT6A:  def \_\_init\_\_(self, x, p):  self.temp, self.sum, self.y = 4, 0, 1  self.temp += 1  self.y = self.temp - p  self.sum = self.temp + x  print(x, self.y, self.sum)  def methodA(self):  x = 0  y = 0  y = y + self.y  x = self.y + 2 + self.temp  self.sum = x + y + self.methodB(self.temp, y)  print(x, y, self.sum)  def methodB(self, temp, n):  x = 0  temp += 1  self.y = self.y + temp  x = x + 3 + n  self.sum = self.sum + x + self.y  print(x, self.y, self.sum)  return self.sum  q1 = FinalT6A(2,1)  q1.methodA()  q1.methodA()  q2 = FinalT6A(3,2)  q2.methodA() |
| --- |

Task 12:

Write the tracing output in a table

| class Test3:  def \_\_init\_\_(self):  self.sum, self.y = 0, 0  def methodA(self):  x, y = 2, 3  msg = [0]  msg[0] = 3  y = self.y + msg[0]  self.methodB(msg, msg[0])  x = self.y + msg[0]  self.sum = x + y + msg[0]  print(x, y, self.sum)  def methodB(self, mg2, mg1):  x = 0  self.y = self.y + mg2[0]  x = x + 33 + mg1  self.sum = self.sum + x + self.y  mg2[0] = self.y + mg1  mg1 = mg1 + x + 2  print(x, self.y, self.sum)  t3 = Test3()  t3.methodA()  t3.methodA()  t3.methodA()  t3.methodA() |
| --- |

Task 13:

Write the tracing output in a table

| class msgClass:  def \_\_init\_\_(self):  self.content = 0    class Q5:  def \_\_init\_\_(self):  self.sum = 1  self.x = 2  self.y = 3  def methodA(self):  x, y = 1, 1  msg = []  myMsg = msgClass()  myMsg.content = self.x  msg.append(myMsg)  msg[0].content = self.y + myMsg.content  self.y = self.y + self.methodB(msg[0])  y = self.methodB(msg[0]) + self.y  x = y + self.methodB(msg[0], msg)  self.sum = x + y + msg[0].content  print(x," ", y," ", self.sum)  def methodB(self, mg1, mg2 = None):  if mg2 == None:  x, y = 5, 6  y = self.sum + mg1.content  self.y = y + mg1.content  x = self.x + 7 +mg1.content  self.sum = self.sum + x + y  self.x = mg1.content + x +8  print(x, " ", y," ", self.sum)  return y  else:  x = 1  self.y += mg2[0].content  mg2[0].content = self.y + mg1.content  x = x + 4 + mg1.content  self.sum += x + self.y  mg1.content = self.sum - mg2[0].content  print(self.x, " ",self.y," ", self.sum)  return self.sum  q = Q5()  q.methodA() |
| --- |

Task 14:

Write the tracing output in a table

| class msgClass:  def \_\_init\_\_(self):  self.content = 0  class Q5:  def \_\_init\_\_(self,z):  self.sum = 1  self.x = 2  self.y = 3  self.z = z  def methodA(self):  x, y = 1, 1  msg = []  myMsg = msgClass()  myMsg.content = self.x  msg.append(myMsg)  msg[0].content = self.y + myMsg.content  self.y = self.y + self.methodB(self.z, msg[0])##  y = self.methodB(self.z, msg[0]) + self.y  x = y + self.methodB(self.z, msg[0], msg)  self.sum = x + y + msg[0].content  print(x," ", y," ", self.sum," ",self.z[0])    def methodB(self, mg0, mg1, mg2 = None):  if mg2 == None:  x, y = 5, 6  y = self.sum + mg1.content  self.y = y + mg1.content  x = self.x + 7 +mg1.content  self.sum = self.sum + x + y  self.x = mg1.content + x +8  mg0[0]+=2  print(x, " ", y," ", self.sum)  return y  else:  x = 1  self.y += mg2[0].content  mg2[0].content = self.y + mg1.content  x = x + 4 + mg1.content  self.sum += x + self.y  mg1.content = self.sum - mg2[0].content  print(self.x, " ",self.y," ", self.sum)  return self.sum  q = Q5([5])  q.methodA() |
| --- |