|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Python Programming  Practical 7 | | | | |
|  |  | |  |  | |
| **Name** | Sahil Shah | | **Roll Number** | 21302C0022 | |
| **Subject/Course:** | Python Programming | **Class** | | | SY BSc. IT |
| **Topic** | Class, Objects, Inheritance & Constructor | **Division** | | | C |
|  |  | |  |  | |
| 1. **Design a class that store the information of three students and display the same.** | | | | | |
| Program:  class students:  count = 0  def \_\_init\_\_(self, name):  self.name = name  self.marks = []  students.count = students.count + 1    def enterMarks(self):  for i in range(3):  m = int(input("Enter the marks of %s in %d subject: "%(self.name, i+1)))  self.marks.append(m)    def display(self):  print (self.name, "got ", self.marks)    name = input("Enter the name of Student: ")  s1 = students(name)  s1.enterMarks()  s1.display()  print ("")  name = input("Enter the name of Student: ")  s2 = students(name)  s2.enterMarks()  s2.display()  print ("")  name = input("Enter the name of Student: ")  s3 = students(name)  s3.enterMarks()  s3.display()  Output Screenshot | | | | | |
|  | | | | | |
| 1. **Write a Python program to Implement the concept of inheritance using python.** | | | | | |
| Program:  class Person(object):  def \_\_init\_\_(self, name, id):  self.name = name  self.id = id    def Display(self):  print(self.name, self.id)    emp = Person("Sahil", 7)  emp.Display()    Output Screenshot | | | | | |
| **3. Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).**  **Write a method called add which returns the sum of the attributes x and y.**  **Write a class method called multiply, which takes a single number parameter an and returns the product of a and MULTIPLIER.**  **Write a static method called subtract, which takes two number parameters, b and c, and returns b - c.** | | | | | |
| Program:  class Numbers:  MULTIPLIER=3.5    def \_\_init\_\_(self,x,y):  self.x=x  self.y=y    def add(self):  return self.x+self.y    @classmethod  def multiply(cls,a):  return cls.MULTIPLIER\*a    @staticmethod  def subtract(b,c):  return b-c    @property  def value(self):  return (self.x,self.y)    #setter  def set\_value(self,x,y):  self.x=x  self.y=y    #deleter  def del\_value(self):  del self.x  del self.y    obj1=Numbers(10,20)    print("add",obj1.add())    # invoke class method  print("mulyiply",Numbers.multiply(10))    # invoke static method  print("subtract ",Numbers.subtract(10,5))    #invoked property  print("property ",obj1.value)    #invoked setter  print(obj1.set\_value(100,200))    #invoked property  print("property ",obj1.value)    #invoked deletor  print(obj1.del\_value())    Output Screenshot | | | | | |