18DCS007

RUDRA BARAD

PP - EXTERNAL PRACTICAL

12/11/2021

Q1. Consider an example of declaring the examination result. Design three classes: Student, Exam, and Result. The Student class has data members/methods such as roll number, name etc. Create the class Exam by inheriting Student class. The Exam class adds fields/methods representing the marks scored in six subjects. Derive the Result from the Exam class, and it has its own fields/methods such as total_marks, SGP (), Show (). Write a python program to model this relationship. (use super (), method overriding)

```python
class Student:
  def __init__(self, name, number):
    self.name = name
    self.number = number

class Exam(Student):
  def __init__(self, name, number, marks_list):
    super().__init__(name, number)
    self.marks_list = marks_list
class Result(Exam):
  def __init__(self, name, number, marks_list):
    super().__init__(name, number, marks_list)
    self.total_marks = 0
    self.sgp = 0

  def calculate_total(self):
    self.total_marks = sum(self.marks_list)

  def calculate_sgp(self):
    self.sgp = self.total_marks / 60

  def show(self):
    print("Name of the student : ",self.name)
    print("Number of the student : ",self.number)
    print("Marks of all subjects : ",self.marks_list)
    print("Total marks : ",self.total_marks)
    print("SGP : ",self.sgp)

roll = input("Please enter your roll number: ")
name = input("Please enter your name: ")
marks = input("Please enter marks in 6 subjects (separated by space): ")
marks = marks.split(" ")
marks = [int(m) for m in marks]

if __name__ == "__main__":
  r = Result(roll, name, marks)
  r.calculate_total()
  r.calculate_sgp()
  r.show()
```

```
    Please enter your roll number: 18DCS007
    Please enter your name: RUDRA BARAD
    Please enter marks in 6 subjects (separated by space): 7 3 5 4 6 2
    Name of the student :  18DCS007
    Number of the student :  RUDRA BARAD
    Marks of all subjects :  [7, 3, 5, 4, 6, 2]
    Total marks :  27
    SGP :  0.45
```

Q2. In a Canadian postal code, the first, third and fifth characters are letters while the second, fourth and sixth characters are numbers. The province can be determined from the first character of a postal code, as shown in the following table. No valid postal codes currently begin with D, F, I, O, Q, U, W, or Z.

The second character in a postal code identifies whether the address is rural or urban. If that character is a 0 then the address is rural. Otherwise it is urban. Create a program that reads a postal code from the user and displays the province associated with it, along with whether the address is urban or rural. For example, if the user enters T2N 1N4 then your program should indicate that the postal code is for an urban address in Alberta. If the user enters X0A 1B2 then your program should indicate that the postal code is for a rural address in Nunavut or Northwest Territories. Use a dictionary to map from the first character of the postal code to the province name. Display a meaningful error message if the postal code begins with an invalid character.

```python
s = input()
c = s[0]
l = s[1]
```

```python
city = "Invalid postal code."
address = "urban"
code_map={'A':"Newfoundland","B":"Nova Scotia","C":"Prince Edward Island",
"E":"New Brunswick","G":"Quebec","H":"Quebec","J":"Quebec",
"K":"Ontario","L":"Ontario","M":"Ontario","N":"Ontario","P":"Ontario",
"R":"Manitoba","S":"Saskatchewan","T":"Alberta","V":"British Columbia",
"X":"Nunavut,Northwest Territories","Y":"Yukon"}

if c in code_map:
  city = code_map[c]
if l=='0':
  address = "rural"
print(city)
print(l)
print(address)
```

```
X0A 1B2
Nunavut,Northwest Territories
0
rural
```

✓ 15s   completed at 9:46 AM   ● ✕