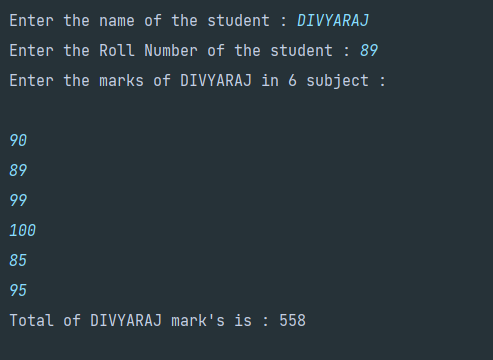
AIM : Consider an example of declaring the examination result. Design three classes: Student, Exam, and Result. The Student class has data members such as those representing rollNumber, Name, etc. Create the class Exam by inheriting Student class. The Exam class adds fields representing the marks scored in six subjects. Derive Result from the Exam class, and it has its own fields such as total\_marks. Write an interactive program to model this relationship.

CODE :

Language : Python.

*class* student:  
 Name = 'DIVYARAJ'  
 rollNumber = 0  
  
 *# function to set the id & name  
 def* details(*self*, rollNumber, Name):  
 *self*.Name = Name  
 *self*.rollNumber = rollNumber  
  
  
*# Creating a class exam from class student.  
class* exam(student):  
 marks\_list = []  
  
 *# function marks to set the marks of that student.  
 def* marks(*self*, marks\_list):  
 *self*.marks\_list = marks\_list  
 *return* marks\_list  
  
  
*# Creating a class result from class exam.  
class* result(exam):  
 marks\_gain = 0  
  
 *# Function to obtain the total of the marks of a student.  
 def* result\_of\_student(*self*, marks\_gain):  
 total\_marks = 0  
 *for* item *in* marks\_gain:  
 total\_marks += item  
 *return* total\_marks  
  
  
*# Creating an object of result class.*sobj = result()  
student\_name = *input*("Enter the name of the student : ")  
student\_id = *input*("Enter the Roll Number of the student : ")  
  
*# Setting the details.*sobj.details(student\_id, student\_name)  
*print*(f"Enter the marks of {student\_name} in 6 subject : \n")  
marks = []  
*for* i *in range*(0, 6):  
 marks.append(*int*(*input*()))  
  
*# Setting the marks.*marks\_obtain = sobj.marks(marks)  
total = sobj.result\_of\_student(marks\_obtain)  
*print*(f"Total of {student\_name} mark's is : {total}")

OUTPUT :



**GITHUB:** [**https://github.com/Dev5577/Programming\_In\_Python.git**](https://github.com/Dev5577/Programming_In_Python.git)