Batch: A3 **Experiment Number:1**

Roll Number:16010423099 **Name: Suryanshu Banerjee**

Aim of the Experiment: Write a program to create StudentInfo class .Calculate the percentage scored by the student

Program/ Steps:

Sr.No	Program	Output
1	class MyClass: x = 5 p1 = MyClass() print(p1.x)	5

2	class Person:
	  def __init__(self, name, age):
	  self.name = name
	  self.age = age
	
	p1 = Person("John", 36)
	
	print(p1.name)
	print(p1.age)
		John
		36

3	class Student:
	  # Constructor - non parameterized
	  def __init__(self):
		This is non parametrized constructor
		Hello John

	print("This is non parametrized constructor") def show(self,name): print("Hello",name) student = Student() student.show("John")	

4	class Student:
	roll_num = 101
	name = "Joseph"
	
	def display(self):
	print(self.roll_num,self.name)
	
	st = Student()
	st.display()
		101 Joseph

5	class Student:
	# Constructor - parameterized
	def __init__(self, name):
	print("This is parametrized constructor")
	self.name = name
	def show(self):
	print("Hello",self.name)
	student = Student("John")
	student.show()
		This is parametrized constructor
		Hello John

2. Write a program to accept Roll Number, Marks Obtained in four subjects, calculate total Marks and percentage scored by the student. Display the roll number, marks obtained, total marks and the percentage scored by the student. Use getter-setter methods.

```
class Student:
    def __init__(self, roll_number):
        self.roll_number = roll_number
        self.marks = [0, 0, 0, 0]

    def set_marks(self, marks):
        self.marks = marks

    def get_total(self):
        return sum(self.marks)

    def get_percentage(self):
        return self.get_total() / 4

    def display(self):
        print(f"Roll Number: {self.roll_number}")
        print(f"Marks: {self.marks}")
        print(f"Total Marks: {self.get_total()}")
        print(f"Percentage: {self.get_percentage():.2f}%")

roll_number = input("Enter Roll Number: ")
marks = list(map(int, input("Enter marks for four subjects separated by space: ").split()))

student = Student(roll_number)
student.set_marks(marks)
student.display()
```

Output/Result:

```
Output                                                    Clear
Enter Roll Number: 16010423099
Enter marks for four subjects separated by space: 56 67 87 98
Roll Number: 16010423099
Marks: [56, 67, 87, 98]
Total Marks: 308
Percentage: 77.00%

=== Code Execution Successful ===
```

Post Lab Question-Answers:

None

Outcomes:

CO1: Describe the fundamental principles of object-oriented programming, including classes, objects, inheritance, encapsulation, data hiding and polymorphism

Conclusion (based on the Results and outcomes achieved):

Successfully applied object oriented programming and executed the program.

References:

Books/ Journals/ Websites referred:

1. Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, First Edition 2017, India
2. Sheetal Taneja and Naveen Kumar, *Python Programming: A modular Approach*, Pearson India, Second Edition 2018,India