





| Experiment No.4 |
| --- |
| Creating functions, classes and objects using Python |
| Date of Performance: 14/02/2024 |
| Date of Submission: 21/02/2024 |

















**Code:**

class Student:

def \_\_init\_\_(self, name="", marks=[0]):

self.name = name

self.marks = marks

def gradeCalculater(self):

res = sum(self.marks) / len(self.marks)

if 90 <= res: return "Grade A"

elif 60 <= res: return "Grade B"

elif 40 <= res: return "Grade C"

else: return "Grade D"

students = []

l = int(input("Enter the number of students: "))

no = int(input("Enter the number of subjects: "))

for i in range(l):

name = input("Enter name: ")

marks = [int(input("Enter a number: ")) for j in range(no)]

obj = Student(name, marks)

students.append(obj)

for i in students:

print(f"Name: {i.name}\nGrade: {i.gradeCalculater()}")

**Output:**

Enter the number of students: 5

Enter the number of subjects: 3

Enter name: adi

Enter a number: 90

Enter a number: 90

Enter a number: 90

Enter name: advait

Enter a number: 56

Enter a number: 56

Enter a number: 56

Enter name: hasi

Enter a number: 23

Enter a number: 23

Enter a number: 23

Enter name: nash

Enter a number: 43

Enter a number: 43

Enter a number: 43

Enter name: ignc

Enter a number: 61

Enter a number: 61

Enter a number: 61

Name: adi

Grade: Grade A

Name: advait

Grade: Grade C

Name: hasi

Grade: Grade D

Name: nash

Grade: Grade C

Name: ignc

Grade: Grade B

**Conclusion:**

In Python, classes are blueprints for creating objects, which are instances of those classes. Classes define the behavior and properties of objects through attributes (variables) and methods (functions). Objects encapsulate data and behavior, allowing for a more organized and modular code structure. Functions are blocks of reusable code that perform specific tasks, and they can be defined both within classes as methods or independently. Classes enable the implementation of object-oriented programming (OOP) principles such as encapsulation, inheritance, and polymorphism, facilitating the creation of complex systems with modular and maintainable code. Together, classes, objects, and functions form the backbone of Python programming, providing a powerful framework for building scalable and flexible applications.