**Write a python script to define a class student having members roll no, name, age, gender. Create a subclass called Test with member marks of 3 subjects. Create three objects of the Test class and display all the details of the student with total marks**

class Student:

def \_\_init\_\_(self, roll\_no, name, age, gender):

self.roll\_no = roll\_no

self.name = name

self.age = age

self.gender = gender

class Test(Student):

def \_\_init\_\_(self, roll\_no, name, age, gender, marks1, marks2, marks3):

super().\_\_init\_\_(roll\_no, name, age, gender)

self.marks1 = marks1

self.marks2 = marks2

self.marks3 = marks3

def total\_marks(self):

return self.marks1 + self.marks2 + self.marks3

def display(self):

print(f"Roll No: {self.roll\_no}")

print(f"Name: {self.name}")

print(f"Age: {self.age}")

print(f"Gender: {self.gender}")

print(f"Total Marks: {self.total\_marks()}")

# Creating three objects

student1 = Test(101, 'Alice', 20, 'Female', 85, 90, 92)

student2 = Test(102, 'Bob', 21, 'Male', 78, 82, 88)

student3 = Test(103, 'Charlie', 22, 'Male', 92, 89, 95)

student1.display()

student2.display()

student3.display()

**. Define a class Employee having members id, name, department, salary. Create a subclass called manager with member bonus. Define methods accept and display in both the classes. Create n objects of the manager class and display the details of the manager having the maximum total salary (salary + bonus).**

class Employee:

def \_\_init\_\_(self, emp\_id, name, department, salary):

self.emp\_id = emp\_id

self.name = name

self.department = department

self.salary = salary

def accept(self):

pass

def display(self):

print(f"ID: {self.emp\_id}, Name: {self.name}, Department: {self.department}, Salary: {self.salary}")

class Manager(Employee):

def \_\_init\_\_(self, emp\_id, name, department, salary, bonus):

super().\_\_init\_\_(emp\_id, name, department, salary)

self.bonus = bonus

def total\_salary(self):

return self.salary + self.bonus

def accept(self):

pass

def display(self):

super().display()

print(f"Bonus: {self.bonus}")

print(f"Total Salary: {self.total\_salary()}")

# Creating Manager objects

manager1 = Manager(101, 'John', 'Sales', 50000, 8000)

manager2 = Manager(102, 'Emma', 'HR', 60000, 10000)

manager3 = Manager(103, 'David', 'Finance', 55000, 12000)

# List of managers

managers = [manager1, manager2, manager3]

# Find the manager with the maximum total salary

max\_salary\_manager = max(managers, key=lambda m: m.total\_salary())

max\_salary\_manager.display()

**. Write Python class to perform addition of two complex numbers using binary + operator overloading.**

class Complex:

def \_\_init\_\_(self, real, imag):

self.real = real

self.imag = imag

def \_\_add\_\_(self, other):

return Complex(self.real + other.real, self.imag + other.imag)

def display(self):

print(f"{self.real} + {self.imag}i")

# Creating complex number objects

c1 = Complex(3, 4)

c2 = Complex(1, 2)

# Adding complex numbers

result = c1 + c2

result.display()

**Write a Python class which has two methods get\_String and print\_String. get\_String accept a string from the user and print\_String print the string in upper case. Further modify the program to reverse a string word by word and print it in lower case.**

class StringManipulator:

def \_\_init\_\_(self):

self.input\_string = ""

def get\_string(self):

self.input\_string = input("Enter a string: ")

def print\_string(self):

print(self.input\_string.upper())

def reverse\_word\_by\_word(self):

words = self.input\_string.split()

reversed\_words = ' '.join(reversed(words))

print(reversed\_words.lower())

# Using the class

str\_manipulator = StringManipulator()

str\_manipulator.get\_string()

str\_manipulator.print\_string()

str\_manipulator.reverse\_word\_by\_word()

**. Write a Python script using class to reverse a String (sentence) word by word.**

class SentenceReversal:

def \_\_init\_\_(self):

self.sentence = ""

def get\_sentence(self):

self.sentence = input("Enter a sentence: ")

def reverse\_sentence(self):

words = self.sentence.split()

reversed\_sentence = ' '.join(reversed(words))

print(reversed\_sentence)

# Using the class

sentence\_reversal = SentenceReversal()

sentence\_reversal.get\_sentence()

sentence\_reversal.reverse\_sentence()