**Code:**

**# Classes, Objects & Inheritance**

**# person is a base class**

**class Person:**

**def \_\_init\_\_(self, n, a):**

**self.name = n**

**self.age = a**

**# employee is the class derived from person using single inheritance**

**class Employee(Person):**

**def \_\_init\_\_(self, n, a, d, s):**

**Person.\_\_init\_\_(self, n, a)**

**self.designation = d**

**self.salary = s**

**def show(self):**

**print("Employee Details: ")**

**print(" Name: ", self.name, "\n Age:", self.age, "\n Designation:", self.designation, "\n Salary:", self.salary)**

**class Student:**

**def \_\_init\_\_(self, id\_, rno):**

**self.studentId = id\_**

**self.room\_no = rno**

**# resident is a class derived from person and student using multiple inheritance**

**class Resident(Person, Student):**

**def \_\_init\_\_(self, n, a, id\_, rno):**

**Person.\_\_init\_\_(self, n, a)**

**Student.\_\_init\_\_(self, id\_, rno)**

**def show(self):**

**print("Resident Details:")**

**print(" Name:", self.name, "\n Age: ", self.age, "\n Id:", self.studentId, "\n Room no.:", self.room\_no)**

**# Creating objects of employee and resident classes**

**e1 = Employee("Sufiyan", 21, "Data Scientist", 200000)**

**r1 = Resident("Vivek", 20, 201900025, 203)**

**e1.show()**

**r1.show()**

**# Exception Handling**

**try:**

**number1, number2 = eval(input("Enter two numbers separated by a comma:"))**

**result = number1 / number2**

**print("Result is",result)**

**except ZeroDivisionError:**

**print("Division by Zero")**

**except SyntaxError:**

**print("A comma may be Missing in the Input")**

**except RuntimeError:**

**print("May be Meaningless ")**

**except:**

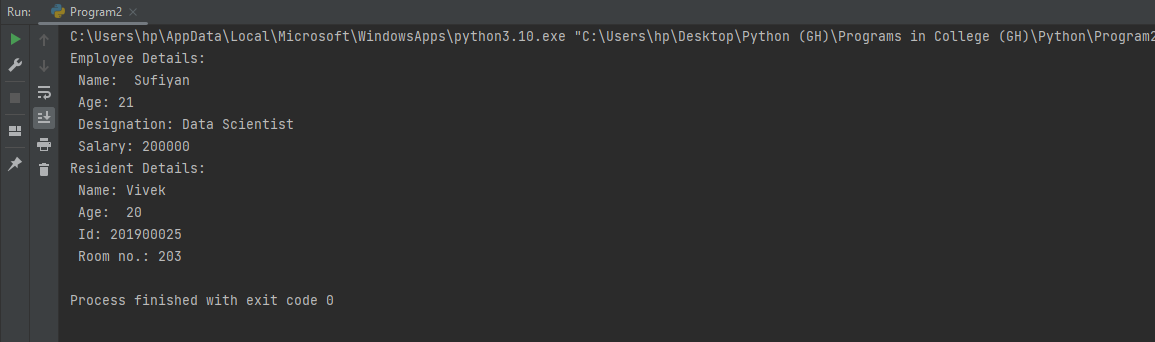
**print("Something Wrong in the Input")**

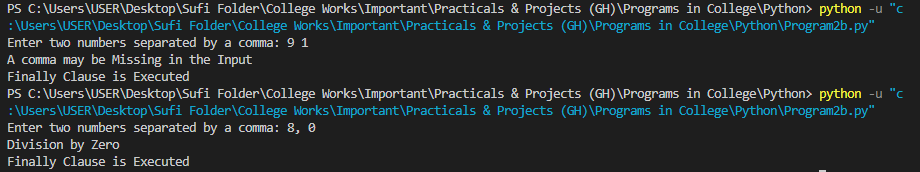
**else:**

**print("No Exceptions")**

**finally:**

**print("Finally Clause is Executed ")**

**Outputs:**

****