6. Object Oriented Programming(OOP):

Create a simple class with attributes and methods.

Instantiate objects from the class and call their methods.

Python code:

class student:

     # Constructor to initialize attributes

    def \_\_init\_\_(self, name, course, year):

        self.name = name

        self.course = course

        self.year = year

     # Method to display information

    def display\_info(self):

        print(f"Name: {self.name}, Course: {self.course}, Year: {self.year}")

students = []

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

for i in range(num):

    print(f"\nEnter details for Student {i + 1}:")

    name = input("Enter name: ")

    course = input("Enter course: ")

    year = int(input("Enter year: "))

 # Create a Student object with the provided information and add it to the list

    pupil = student(name, course, year)

    students.append(pupil)

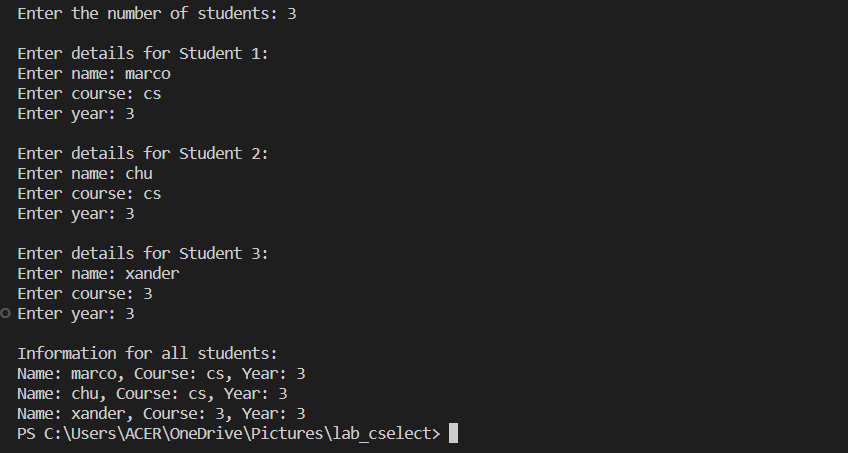
# Display information for all students

print("\nInformation for all students:")

for student in students:

    student.display\_info()

output:



7. Exception Handling:

Write a program that include try-except blocks to handle exceptions gracefully

while True:

    try:

        dividend = float(input("Enter the dividend: "))

        divisor = float(input("Enter the divisor: "))

        result = dividend / divisor

        print(f"The result of {dividend} divided by {divisor} is {result}")

    except ValueError:

        print("Error: Please enter valid numbers for the dividend and divisor.")

        continue

    except ZeroDivisionError:

        print("Error: Cannot divide by zero.")

        continue

    except Exception as e:

        print(f"An error occurred: {e}")

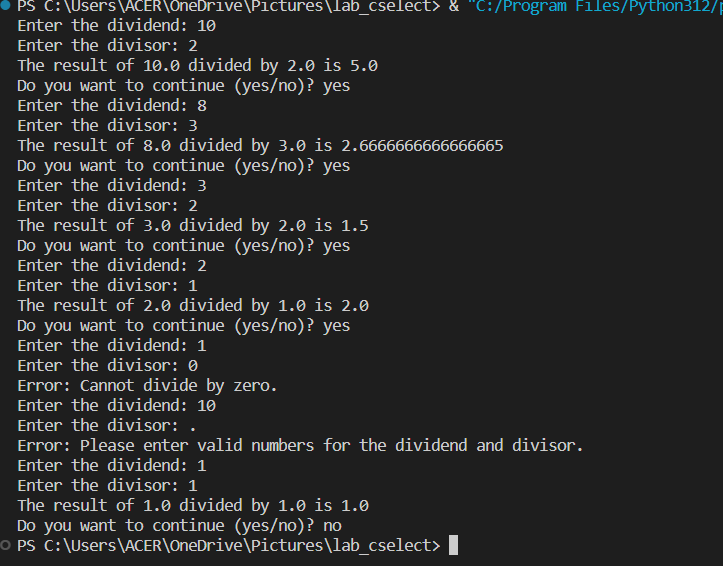
        continue

    response = input("Do you want to continue (yes/no)? ").lower()

    if response != 'yes':

        break

output:



8. File Handling:

Read from and write to a file using file handling operations.

try:

    # Read from input.txt

    with open('input.txt', 'r') as file:

        content = file.read()  # Corrected: Added () to invoke the read() method

    # Write the content read from input.txt into output.txt

    with open('output.txt', 'w') as file:

        file.write(content)

    print("Content copied successfully from input.txt to output.txt.")

except FileNotFoundError:

    print("Error: File not found.")

except Exception as e:

    print(f"An error occurred: {e}")

input.txt



Output.txt





9. Version Control:

Set up a version control system (ie.g., git)

Create a repository, add files, commit changes, and push to a remote repository.

10. Debugging:

Introduced intentional bugs in your code and use debugging tools to identify and fix them.