Lab Assignment 6.3 – AI Assisted Coding

Question 1: Student Class

Algorithm:

```python
# Student Class Example

class Student:
    def __init__(self, name, roll_number, branch):
        self.name = name
        self.roll_number = roll_number
        self.branch = branch

    def display_details(self):
        print("Student Name:", self.name)
        print("Roll Number:", self.roll_number)
        print("Branch:", self.branch)


# Creating an object
student1 = Student("Mani", "23CS001", "CSE")
student1.display_details()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    Python

PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.exe" "
 manikanta/OneDrive/Desktop/ai assistant/lab6.3.py"
Student Name: Mani
Roll Number: 23CS001
Branch: CSE
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>
```

1. Define a Student class
2. Initialize attributes using constructor
3. Create display method4. Create object and display details

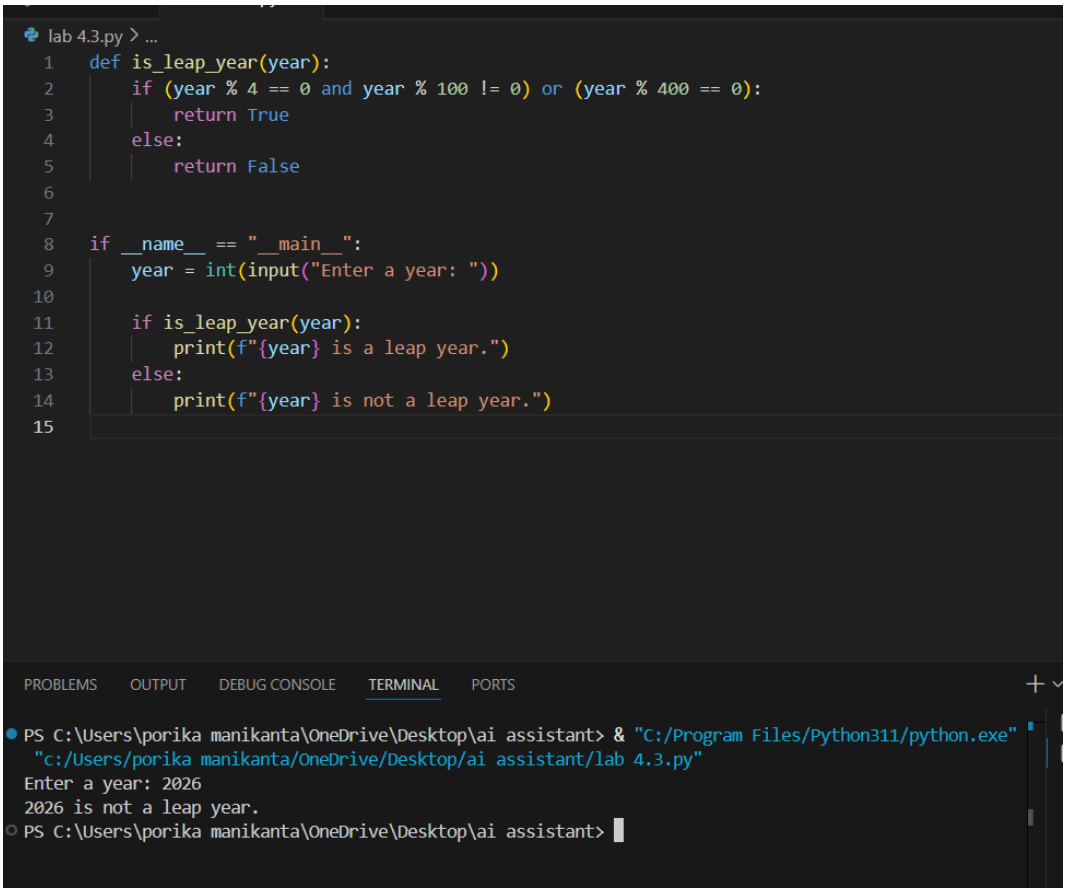
Pseudo Code:

START
Define class Student
Initialize name, roll, branch
Display details
END

Question 2: Multiples Using Loops

Algorithm:
1. Take a number
2. Use loop to print first 10 multiples

Pseudo Code:
START
FOR i from 1 to 10
Print num * i
END

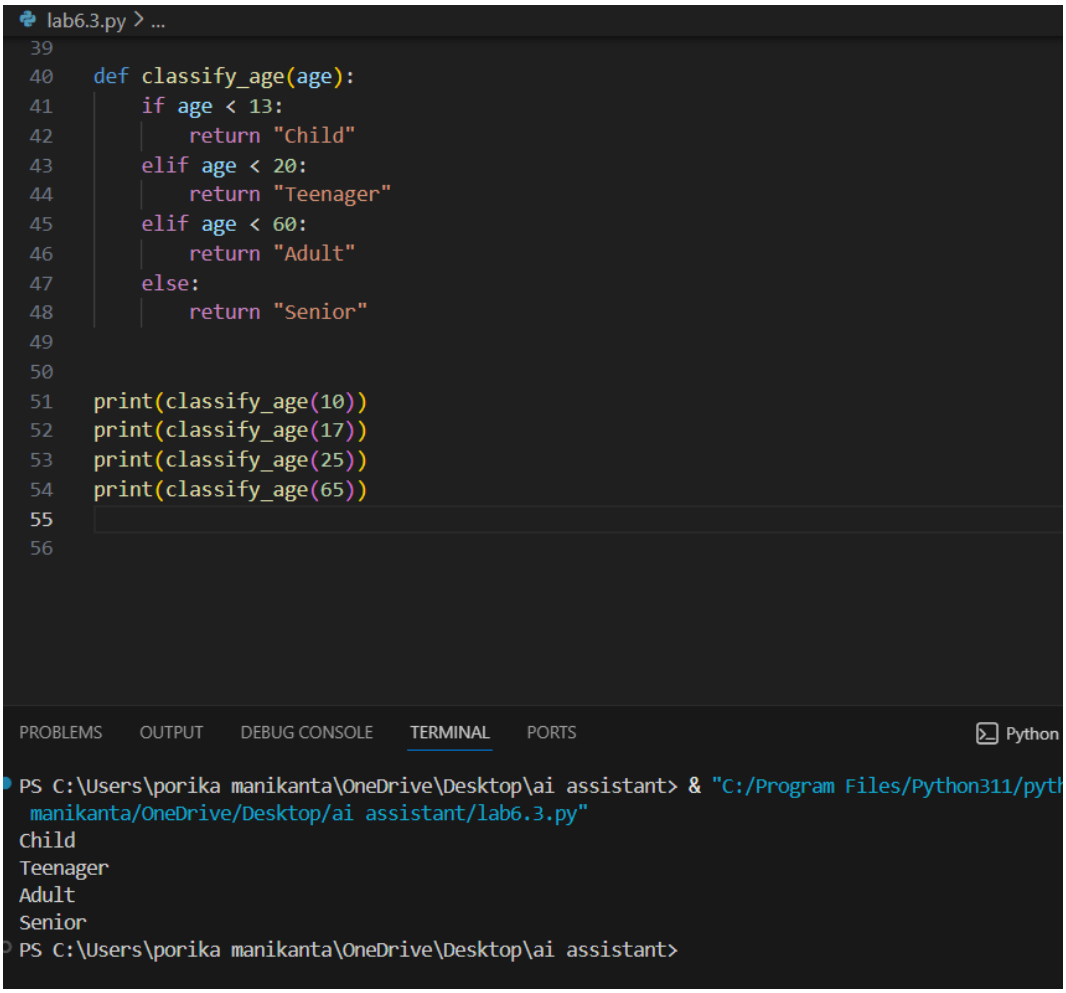


Question 3: Age Classification

Algorithm:
1. Take age as input 2. Use
if-elif-else to classify

Pseudo Code:

START
IF age < 13 print Child
ELSE IF age < 20 print Teenager
ELSE IF age < 60 print Adult
ELSE print Senior
END



Question 4: Sum of First n Numbers

Algorithm:

1. Initialize sum
2. Use loop/formula to compute sum

Pseudo Code:
START
sum = n*(n+1)/2
Print sum
END

Question 5: Bank Account Class

Algorithm:

1. Define BankAccount class 2. Implement deposit, withdraw, balance methods

Pseudo Code:
START
Create class BankAccount
Deposit, Withdraw, Check balance
END

lab6.3.py > ...

```python

# Bank Account Class
class BankAccount:
    def __init__(self, account_holder, balance=0):
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited ₹{amount}")
        else:
            print("Invalid deposit amount")

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            print(f"Withdrawn ₹{amount}")
        else:
            print("Insufficient balance")

    def check_balance(self):
        print(f"Current Balance: ₹{self.balance}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Pyth

```
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/py
 manikanta/OneDrive/Desktop/ai assistant/lab6.3.py"
Current Balance: ₹1000
Deposited ₹500
Withdrawn ₹300
Current Balance: ₹1200
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>
```