2403A52L05—B50

E. Sai Nithin Ramanujan

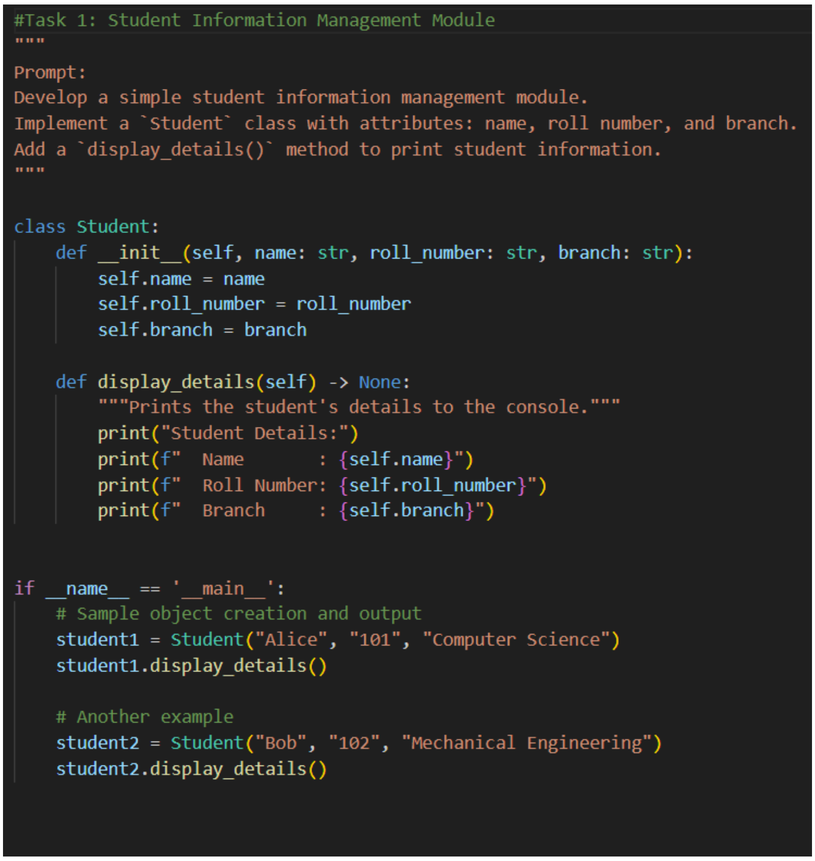
ASS:6.3

Task 1: Classes (Student Class)

Scenario:

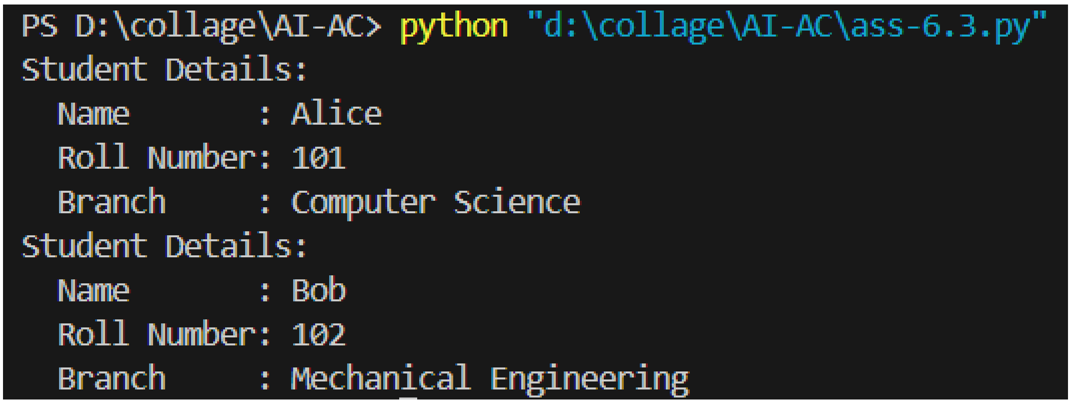
You are developing a simple student information management module.

Prompt and code:



```
#Task 1: Student Information Management Module
"""
Prompt:
Develop a simple student information management module.
Implement a `Student` class with attributes: name, roll number, and branch.
Add a `display_details()` method to print student information.
"""

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

    def display_details(self) -> None:
        """Prints the student's details to the console."""
        print("Student Details:")
        print(f"  Name       : {self.name}")
        print(f"  Roll Number: {self.roll_number}")
        print(f"  Branch     : {self.branch}")


if __name__ == '__main__':
    # Sample object creation and output
    student1 = Student("Alice", "101", "Computer Science")
    student1.display_details()

    # Another example
    student2 = Student("Bob", "102", "Mechanical Engineering")
    student2.display_details()
```

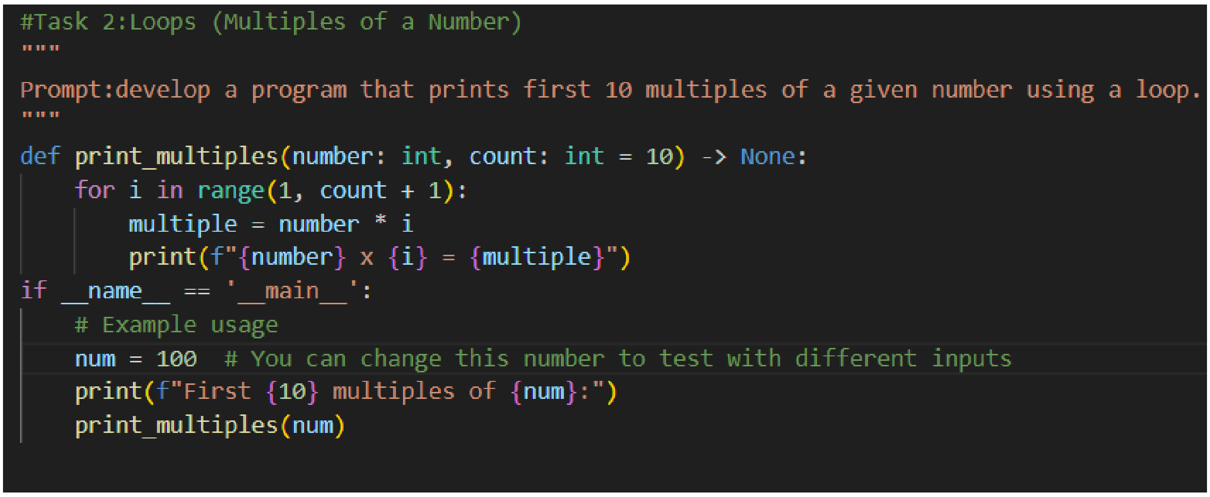
Output:



```
PS D:\collage\AI-AC> python "d:\collage\AI-AC\ass-6.3.py"
Student Details:
  Name       : Alice
  Roll Number: 101
  Branch     : Computer Science
Student Details:
  Name       : Bob
  Roll Number: 102
  Branch     : Mechanical Engineering
```

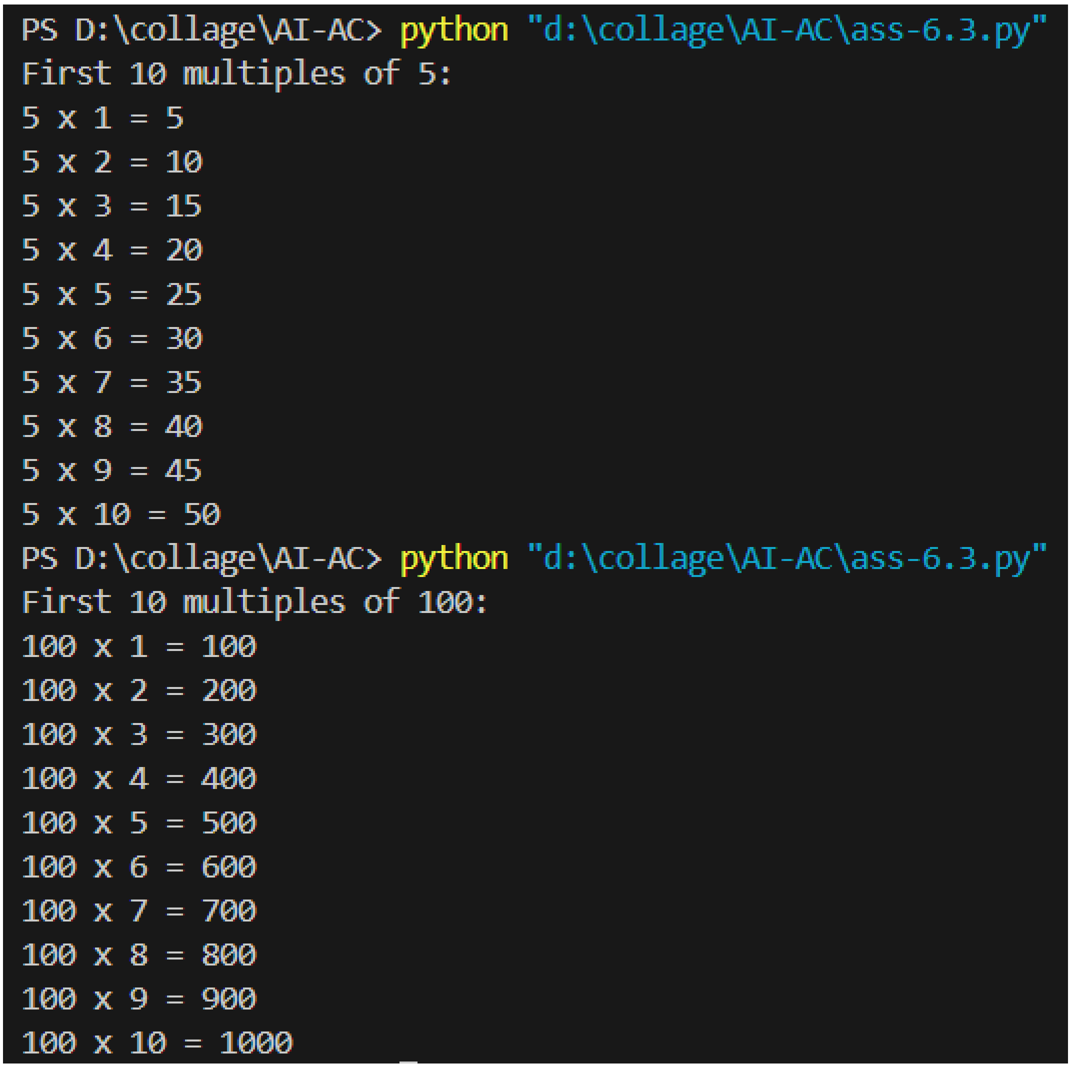
Task 2: Loops (Multiples of a Number)

Prompt and Code:



```
#Task 2:Loops (Multiples of a Number)
"""
Prompt:develop a program that prints first 10 multiples of a given number using a loop.
"""
def print_multiples(number: int, count: int = 10) -> None:
    for i in range(1, count + 1):
        multiple = number * i
        print(f"{number} x {i} = {multiple}")
if __name__ == '__main__':
    # Example usage
    num = 100  # You can change this number to test with different inputs
    print(f"First {10} multiples of {num}:")
    print_multiples(num)
```

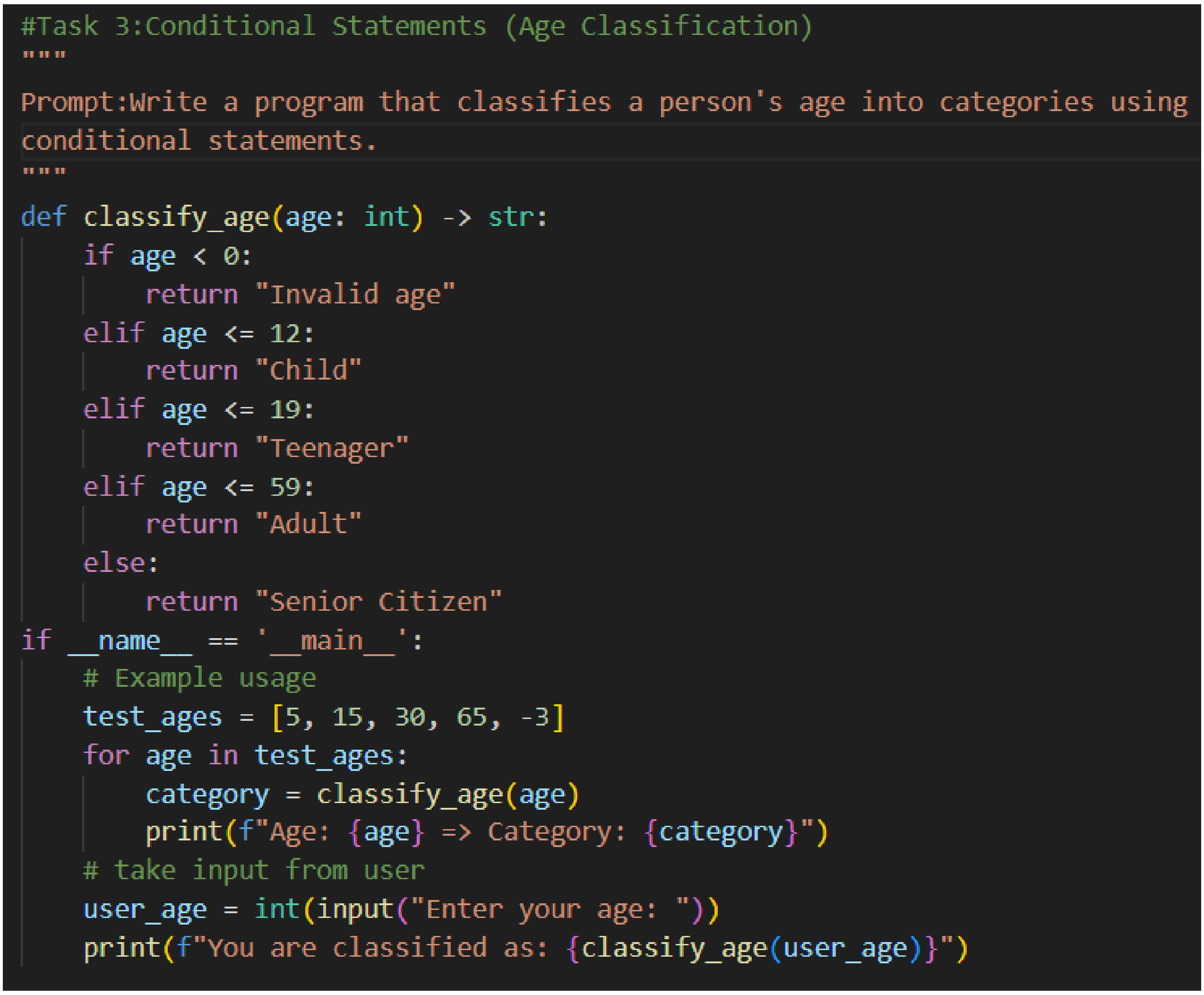
Output:



```
PS D:\collage\AI-AC> python "d:\collage\AI-AC\ass-6.3.py"
First 10 multiples of 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
PS D:\collage\AI-AC> python "d:\collage\AI-AC\ass-6.3.py"
First 10 multiples of 100:
100 x 1 = 100
100 x 2 = 200
100 x 3 = 300
100 x 4 = 400
100 x 5 = 500
100 x 6 = 600
100 x 7 = 700
100 x 8 = 800
100 x 9 = 900
100 x 10 = 1000
```

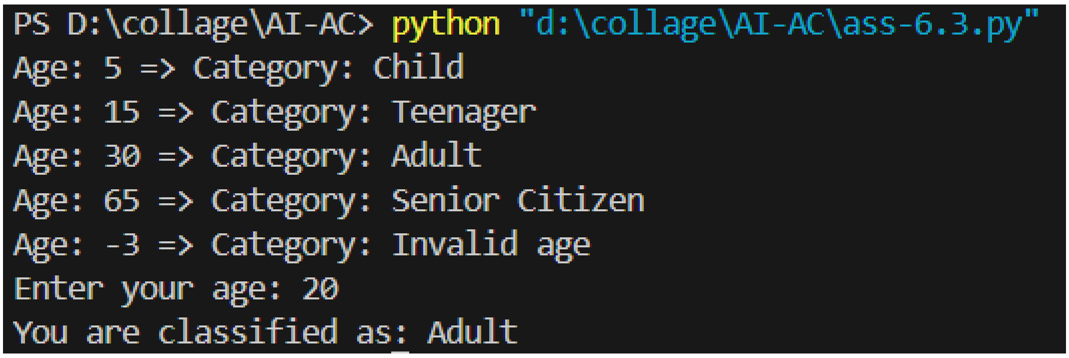
Task 3: Conditional Statements (Age Classification)

Prompt and Code:



```python
#Task 3:Conditional Statements (Age Classification)
"""
Prompt:Write a program that classifies a person's age into categories using
conditional statements.
"""
def classify_age(age: int) -> str:
    if age < 0:
        return "Invalid age"
    elif age <= 12:
        return "Child"
    elif age <= 19:
        return "Teenager"
    elif age <= 59:
        return "Adult"
    else:
        return "Senior Citizen"
if __name__ == '__main__':
    # Example usage
    test_ages = [5, 15, 30, 65, -3]
    for age in test_ages:
        category = classify_age(age)
        print(f"Age: {age} => Category: {category}")
    # take input from user
    user_age = int(input("Enter your age: "))
    print(f"You are classified as: {classify_age(user_age)}")
```

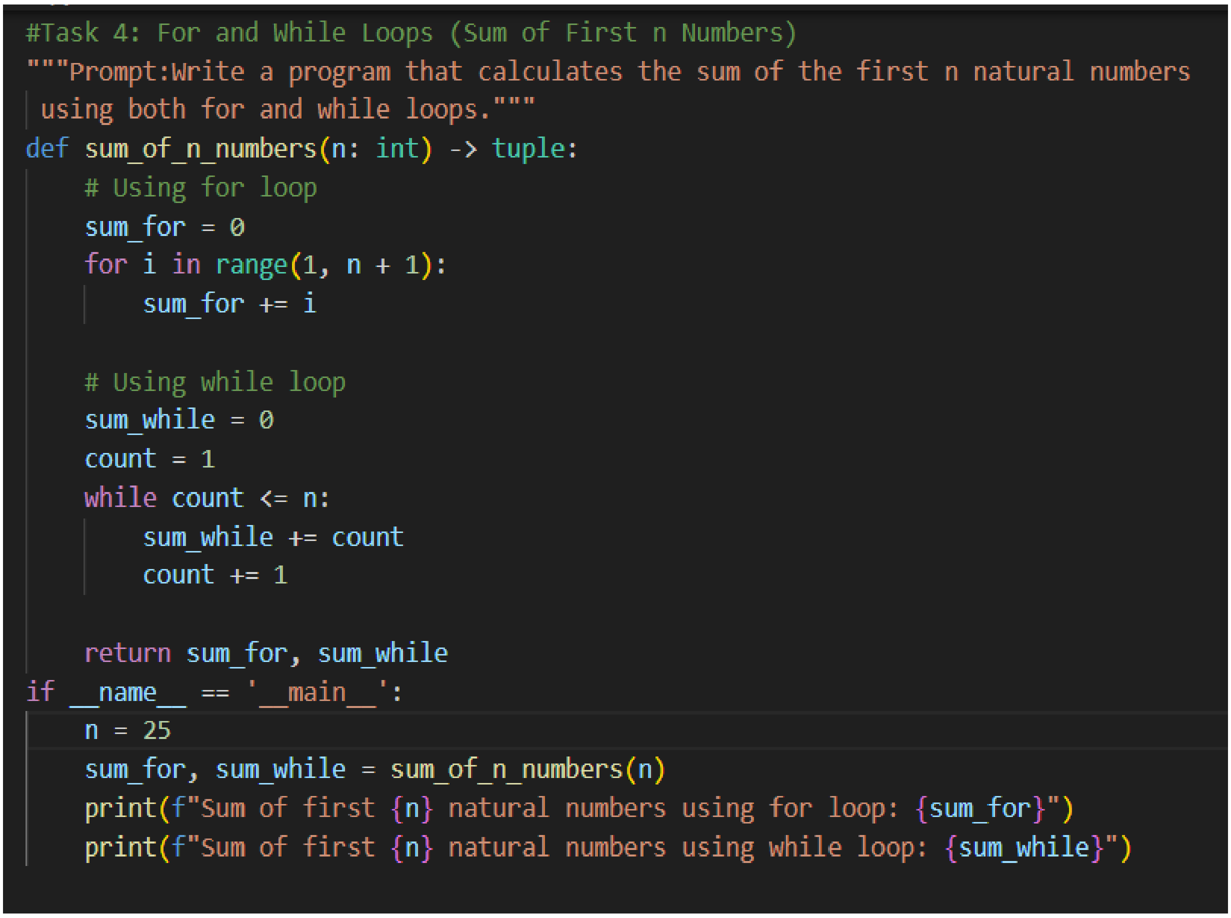
Output:



```
PS D:\collage\AI-AC> python "d:\collage\AI-AC\ass-6.3.py"
Age: 5 => Category: Child
Age: 15 => Category: Teenager
Age: 30 => Category: Adult
Age: 65 => Category: Senior Citizen
Age: -3 => Category: Invalid age
Enter your age: 20
You are classified as: Adult
```

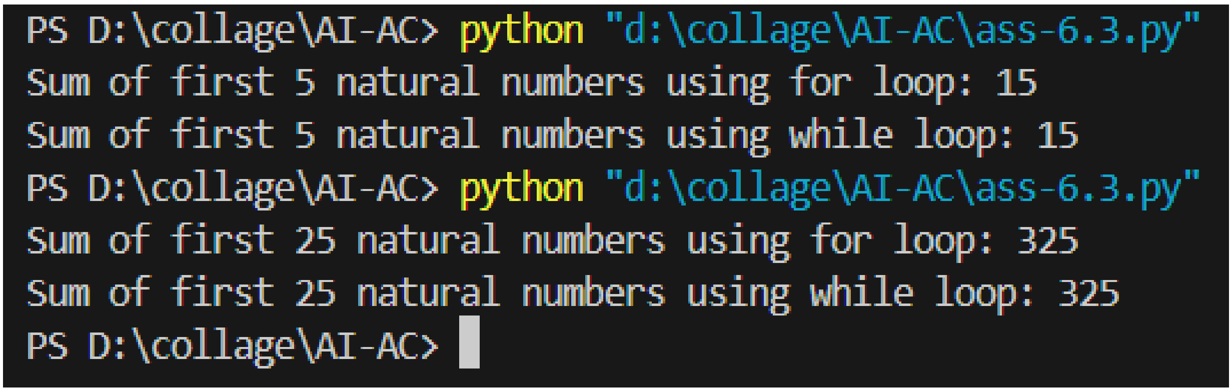
Task 4: For and While Loops (Sum of First n Numbers)

Prompt and code:



```python
#Task 4: For and While Loops (Sum of First n Numbers)
"""Prompt:Write a program that calculates the sum of the first n natural numbers
 using both for and while loops."""
def sum_of_n_numbers(n: int) -> tuple:
    # Using for loop
    sum_for = 0
    for i in range(1, n + 1):
        sum_for += i

    # Using while loop
    sum_while = 0
    count = 1
    while count <= n:
        sum_while += count
        count += 1

    return sum_for, sum_while
if __name__ == '__main__':
    n = 25
    sum_for, sum_while = sum_of_n_numbers(n)
    print(f"Sum of first {n} natural numbers using for loop: {sum_for}")
    print(f"Sum of first {n} natural numbers using while loop: {sum_while}")
```

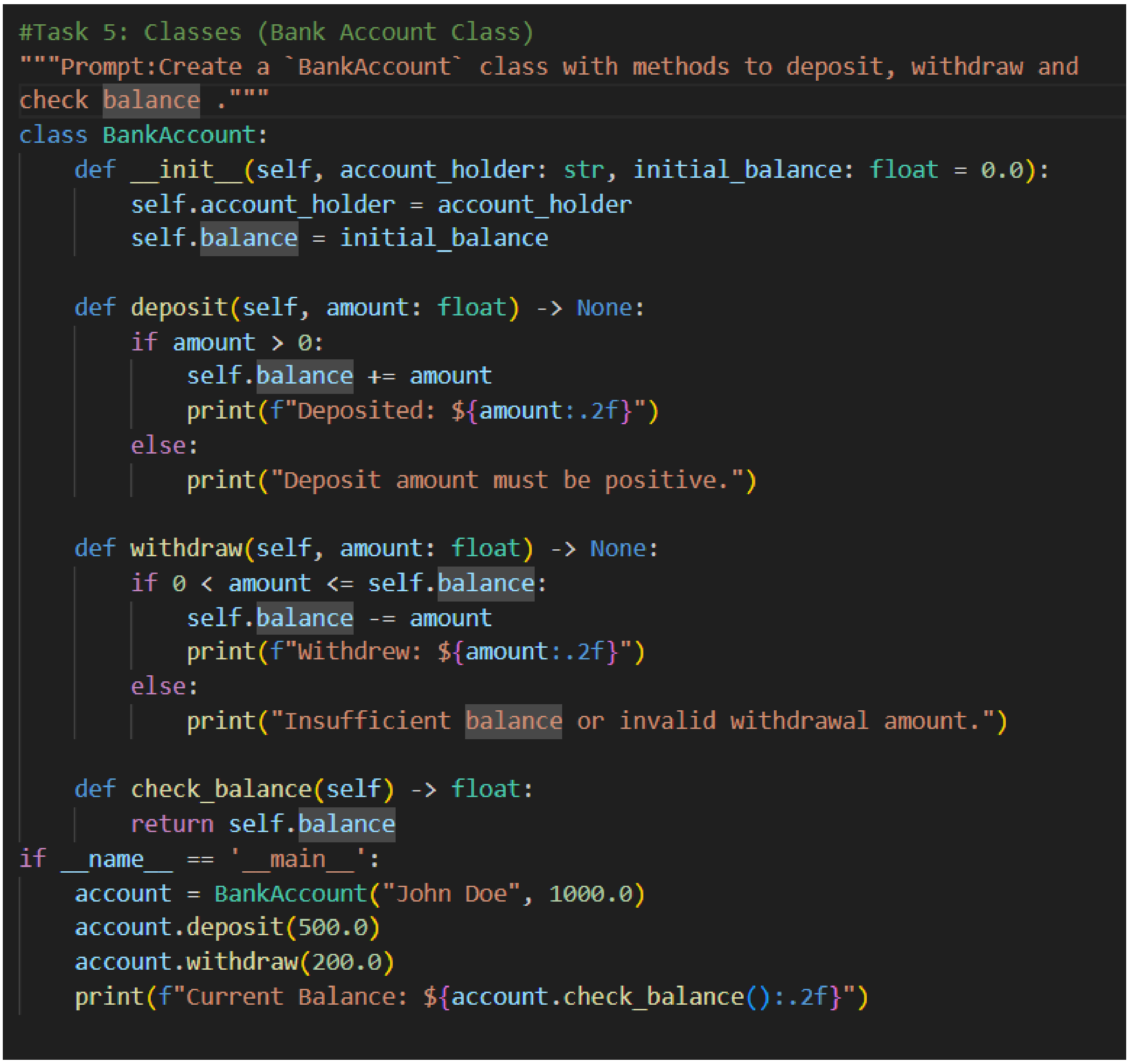
Output:



```
PS D:\collage\AI-AC> python "d:\collage\AI-AC\ass-6.3.py"
Sum of first 5 natural numbers using for loop: 15
Sum of first 5 natural numbers using while loop: 15
PS D:\collage\AI-AC> python "d:\collage\AI-AC\ass-6.3.py"
Sum of first 25 natural numbers using for loop: 325
Sum of first 25 natural numbers using while loop: 325
PS D:\collage\AI-AC> 
```

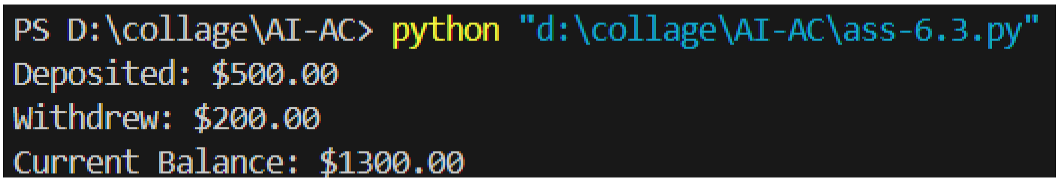
Task 5: Classes (Bank Account Class)

Prompt and Code:



```
#Task 5: Classes (Bank Account Class)
"""Prompt:Create a `BankAccount` class with methods to deposit, withdraw and
check balance ."""
class BankAccount:
    def __init__(self, account_holder: str, initial_balance: float = 0.0):
        self.account_holder = account_holder
        self.balance = initial_balance

    def deposit(self, amount: float) -> None:
        if amount > 0:
            self.balance += amount
            print(f"Deposited: ${amount:.2f}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount: float) -> None:
        if 0 < amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew: ${amount:.2f}")
        else:
            print("Insufficient balance or invalid withdrawal amount.")

    def check_balance(self) -> float:
        return self.balance
if __name__ == '__main__':
    account = BankAccount("John Doe", 1000.0)
    account.deposit(500.0)
    account.withdraw(200.0)
    print(f"Current Balance: ${account.check_balance():.2f}")
```

Output:



```
PS D:\collage\AI-AC> python "d:\collage\AI-AC\ass-6.3.py"
Deposited: $500.00
Withdrew: $200.00
Current Balance: $1300.00
```