

| SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE                          |   | DEPARTMENT OF COMPUTER SCIENCE ENGINEERING |                           |
|---|---|--|---------------------------|
| ProgramName:B. Tech   | Assignment Type: Lab  |  | AcademicYear:2025-2026    |
| CourseCoordinatorName   | Venkataramana Veeramsetty   |  |                           |
| Instructor(s)Name   | Dr. V. Venkataramana (Co-ordinator)<br>Dr. T. Sampath Kumar<br>Dr. Pramoda Patro<br>Dr. Brij Kishor Tiwari<br>Dr.J.Ravichander<br>Dr. Mohammand Ali Shaik<br>Dr. Anirodh Kumar<br>Mr. S.Naresh Kumar<br>Dr. RAJESH VELPULA<br>Mr. Kundhan Kumar<br>Ms. Ch.Rajitha<br>Mr. M Prakash<br>Mr. B.Raju<br>Intern 1 (Dharma teja)<br>Intern 2 (Sai Prasad)<br>Intern 3 (Sowmya)<br>NS_2 ( Mounika) |  |                           |
| CourseCode  | 24CS002PC215  | CourseTitle                                | AI Assisted Coding        |
| Year/Sem  | II/I  | Regulation                                 | R24                       |
| Date and Day of Assignment  | Week3 - Wednesday   | Time(s)                                    |                           |
| Duration  | 2 Hours   | Applicableto Batches                       |                           |
| AssignmentNumber:6.3(Present assignment number)/24(Total number of assignments) |   |  |                           |
| Q.No.   | Question  |  | Expected Time to complete |
| 1   | <p>Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals</p> <p><b>Lab Objectives:</b></p> <ul style="list-style-type: none"> <li>• To explore AI-powered auto-completion features for core Python constructs.</li> <li>• To analyze how AI suggests logic for class definitions, loops, and</li> </ul>  |  | Week3 - Wednesday         |

|  |   |  |
|--|---|--|
|  | <p>conditionals.</p> <ul style="list-style-type: none"> <li>To evaluate the completeness and correctness of code generated by AI assistants.</li> </ul> <p><b>Lab Outcomes (LOs):</b></p> <p>After completing this lab, students will be able to:</p> <ul style="list-style-type: none"> <li>Use AI tools to generate and complete class definitions and methods.</li> <li>Understand and assess AI-suggested loops for iterative tasks.</li> <li>Generate conditional statements through prompt-driven suggestions.</li> <li>Critically evaluate AI-assisted code for correctness and clarity.</li> </ul> <p><b>Task Description#1 (Classes)</b></p> <ul style="list-style-type: none"> <li>Use AI to complete a Student class with attributes and a method.</li> <li>Check output</li> <li>Analyze the code generated by AI tool</li> </ul> <p><b>Expected Output#1</b></p> <ul style="list-style-type: none"> <li>Class with constructor and display_details() method</li> </ul> <p><b>Task Description#2 (Loops)</b></p> <ul style="list-style-type: none"> <li>Prompt AI to complete a function that prints the first 10 multiples of a number using a loop.</li> <li>Analyze the generated code</li> <li>Ask AI to generate code using other controlled looping</li> </ul> <p><b>Expected Output#2</b></p> <ul style="list-style-type: none"> <li>Correct loop-based implementation</li> </ul> <p><b>Task Description#3 (Conditional Statements)</b></p> <ul style="list-style-type: none"> <li>Ask AI to write nested if-elif-else conditionals to classify age groups.</li> <li>Analyze the generated code</li> <li>Ask AI to generate code using other conditional statements</li> </ul> <p><b>Expected Output#3</b></p> <ul style="list-style-type: none"> <li>Age classification function with appropriate conditions and with explanation</li> </ul> <p><b>Task Description#4 (For and While loops)</b></p> <ul style="list-style-type: none"> <li>Generate a sum_to_n() function to calculate sum of first n numbers</li> <li>Analyze the generated code</li> <li>Get suggestions from AI with other controlled looping</li> </ul> <p><b>Expected Output#4</b></p> <ul style="list-style-type: none"> <li>Python code with explanation</li> </ul> <p><b>Task Description#5 (Class)</b></p> <ul style="list-style-type: none"> <li>Use AI to build a BankAccount class with deposit, withdraw, and</li> </ul> |  |
|--|---|--|

- balance methods.
- Analyze the generated code
  - Add comments and explain code

**Expected Output#5**

- Python code with explanation

**Note:** Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

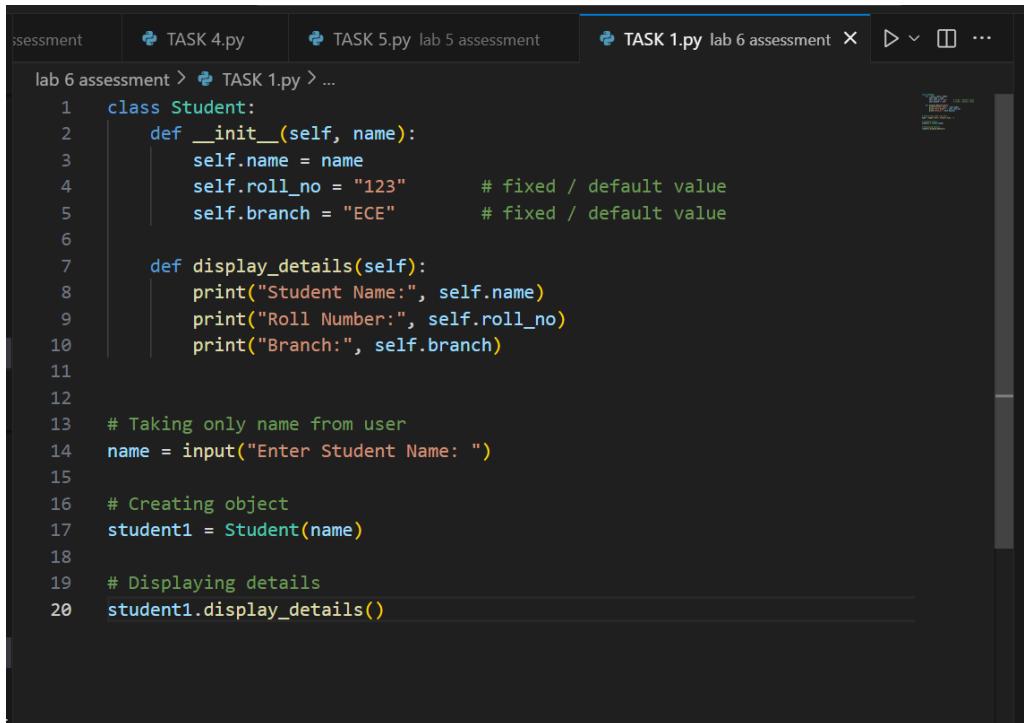
**Evaluation Criteria:**

| Criteria               | Max Marks        |
|------------------------|------------------|
| Class                  | 1.0              |
| Loops                  | 1.0              |
| Conditional Statements | 0.5              |
| <b>Total</b>           | <b>2.5 Marks</b> |

## Task Description#1 (Classes)

- Use AI to complete a Student class with attributes and a method.
- Check output
- Analyze the code generated by AI tool

**PROMPT:** Create a Python Student class with attributes and a method to display the student details.

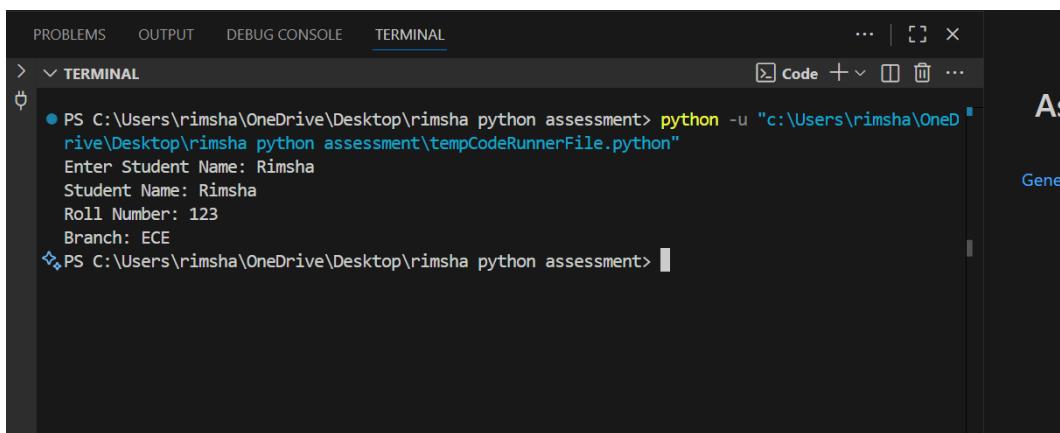


```
lab 6 assessment > TASK 1.py > ...
1  class Student:
2      def __init__(self, name):
3          self.name = name
4          self.roll_no = "123"      # fixed / default value
5          self.branch = "ECE"       # fixed / default value
6
7      def display_details(self):
8          print("Student Name:", self.name)
9          print("Roll Number:", self.roll_no)
10         print("Branch:", self.branch)
11
12
13 # Taking only name from user
14 name = input("Enter Student Name: ")
15
16 # Creating object
17 student1 = Student(name)
18
19 # Displaying details
20 student1.display_details()
```

### Expected Output#1

- Class with constructor and display\_details() method

**Practical Output:**

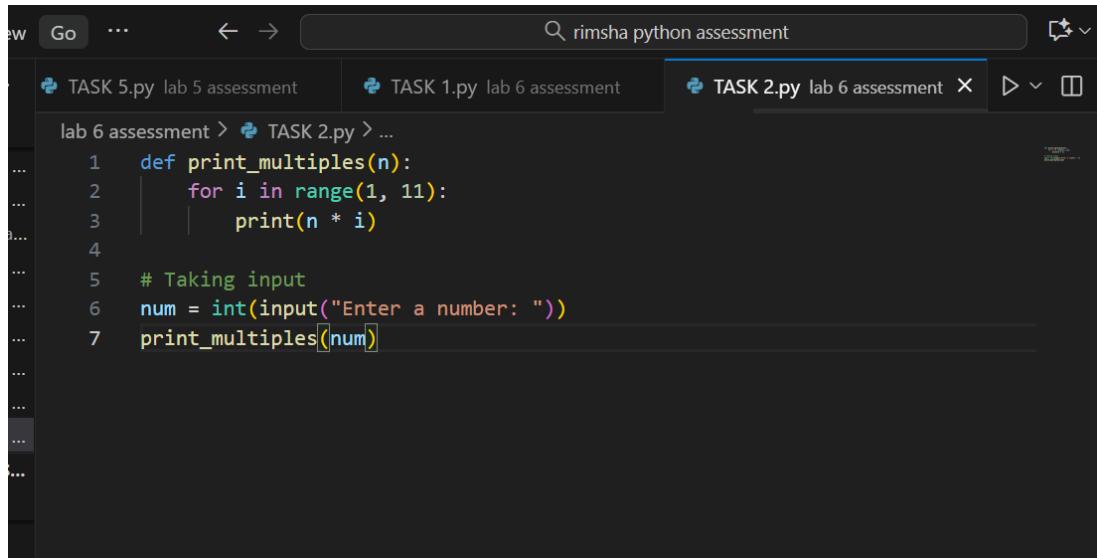


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... | [ ] X
> < TERMINAL
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneDrive\Desktop\rimsha python assessment\tempCodeRunnerFile.py"
Enter Student Name: Rimsha
Student Name: Rimsha
Roll Number: 123
Branch: ECE
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment>
```

## Task Description#2 (Loops)

- Prompt AI to complete a function that prints the first 10 multiples of a number using a loop.
- Analyze the generated code
- Ask AI to generate code using other controlled looping

**PROMPT:** Prompt AI to complete a function that prints the first 10 multiples of a number using a loop.



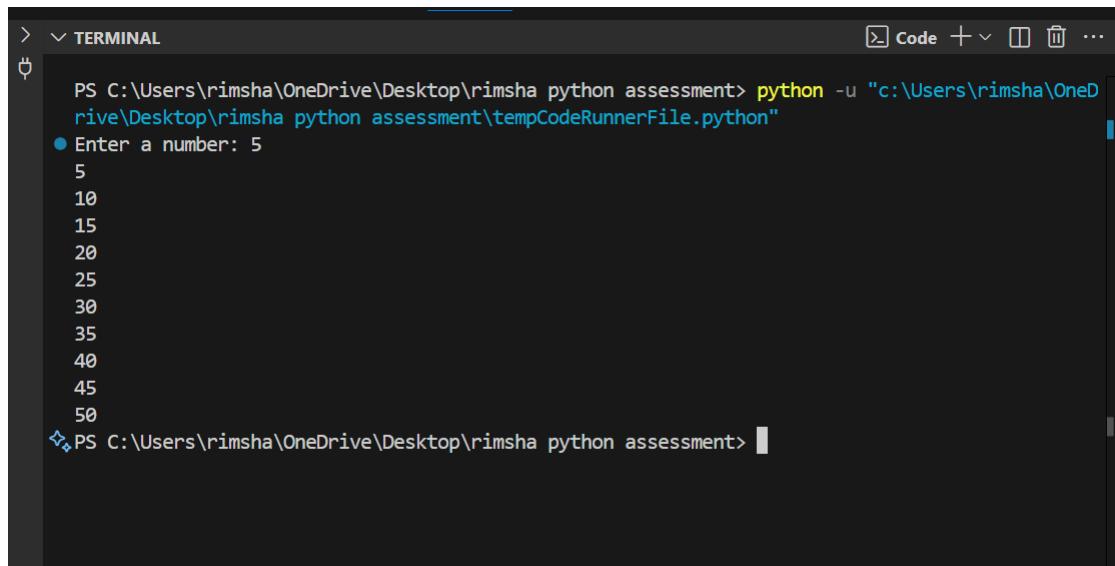
```
def print_multiples(n):
    for i in range(1, 11):
        print(n * i)

# Taking input
num = int(input("Enter a number: "))
print_multiples(num)
```

## Expected Output#2

- Correct loop-based implementation

**Practical output:**

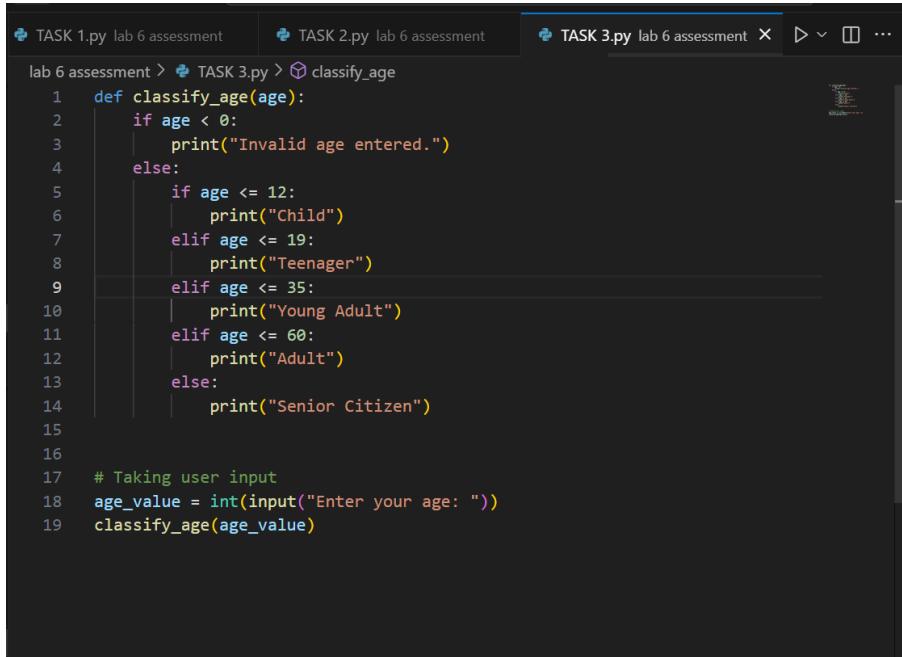


```
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneDrive\Desktop\rimsha python assessment\tempCodeRunnerFile.python"
● Enter a number: 5
5
10
15
20
25
30
35
40
45
50
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment>
```

### Task Description#3 (Conditional Statements)

- Ask AI to write nested if-elif-else conditionals to classify age groups.
- Analyze the generated code
- Ask AI to generate code using other conditional statements

**PROMPT:** Write Python code to classify age using nested if-elif-else, then explain

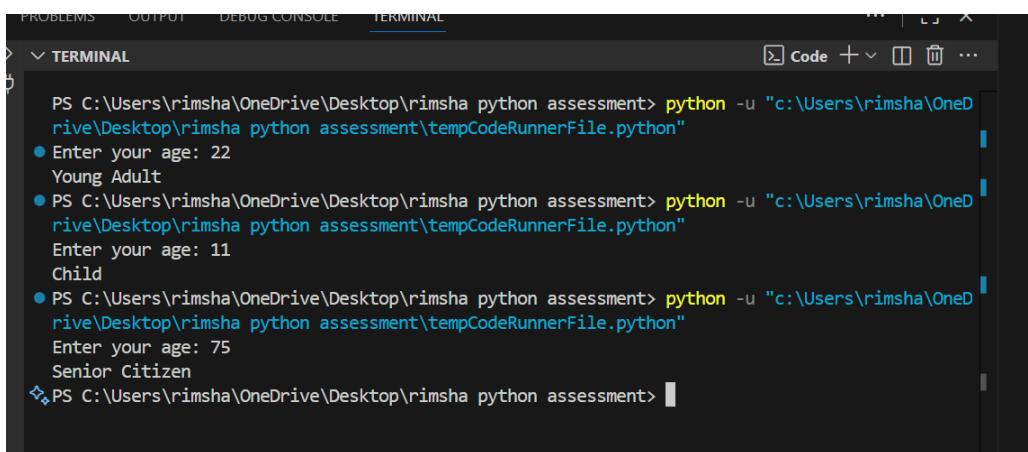


```
lab 6 assessment > TASK 3.py > classify_age
1 def classify_age(age):
2     if age < 0:
3         print("Invalid age entered.")
4     else:
5         if age <= 12:
6             print("Child")
7         elif age <= 19:
8             print("Teenager")
9         elif age <= 35:
10            print("Young Adult")
11        elif age <= 60:
12            print("Adult")
13        else:
14            print("Senior Citizen")
15
16
17 # Taking user input
18 age_value = int(input("Enter your age: "))
19 classify_age(age_value)
```

### Expected Output#3

- Age classification function with appropriate conditions and with explanation

**Practical output:**

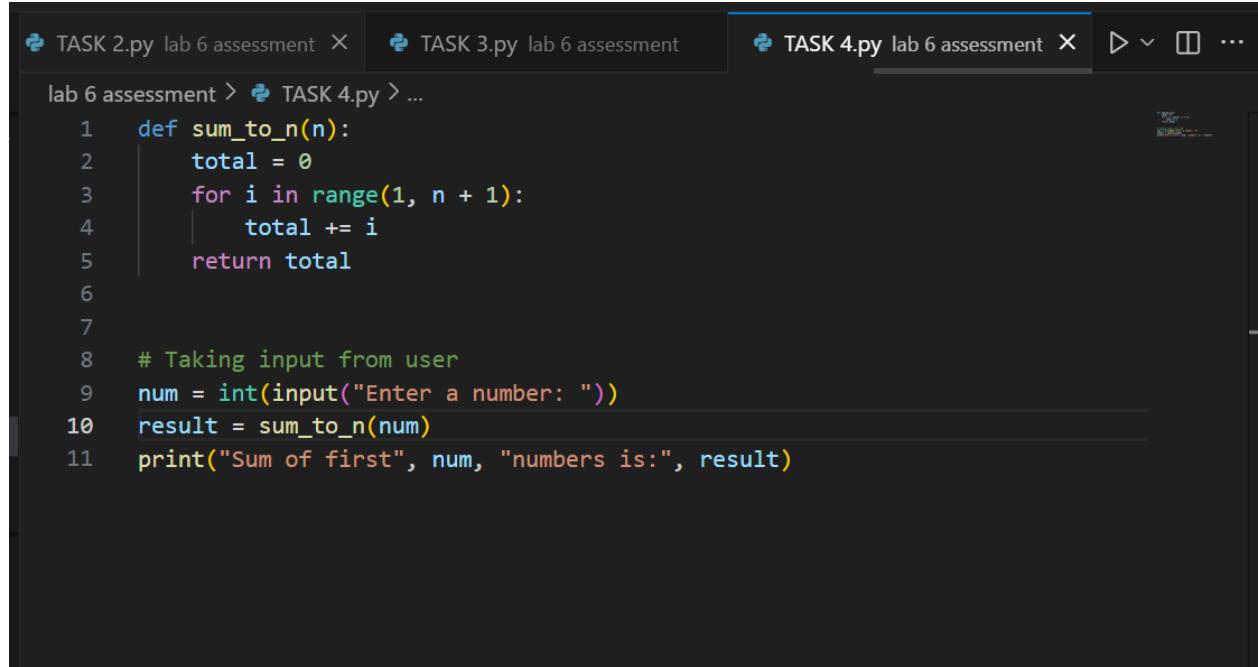


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
> ▾ TERMINAL
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneDrive\Desktop\rimsha python assessment\tempCodeRunnerFile.python"
● Enter your age: 22
Young Adult
● PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneDrive\Desktop\rimsha python assessment\tempCodeRunnerFile.python"
Enter your age: 11
Child
● PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneDrive\Desktop\rimsha python assessment\tempCodeRunnerFile.python"
Enter your age: 75
Senior Citizen
❖ PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment>
```

## Task Description#4 (For and While loops)

- Generate a sum\_to\_n() function to calculate sum of first n numbers
- Analyze the generated code
- Get suggestions from AI with other controlled looping

**PROMPT:** Write a sum\_to\_n() function in Python to sum first n numbers using a loop.

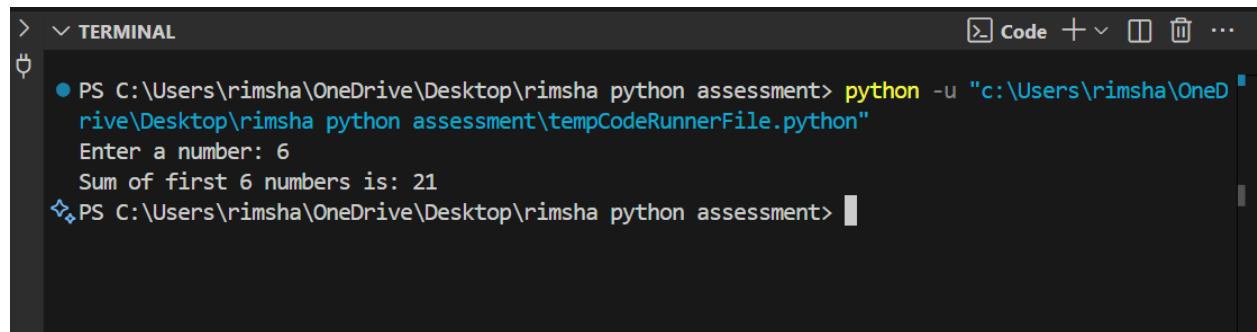


```
lab 6 assessment > TASK 4.py > ...
1  def sum_to_n(n):
2      total = 0
3      for i in range(1, n + 1):
4          total += i
5      return total
6
7
8  # Taking input from user
9 num = int(input("Enter a number: "))
10 result = sum_to_n(num)
11 print("Sum of first", num, "numbers is:", result)
```

## Expected Output#4

- Python code with explanation

**Practical output:**

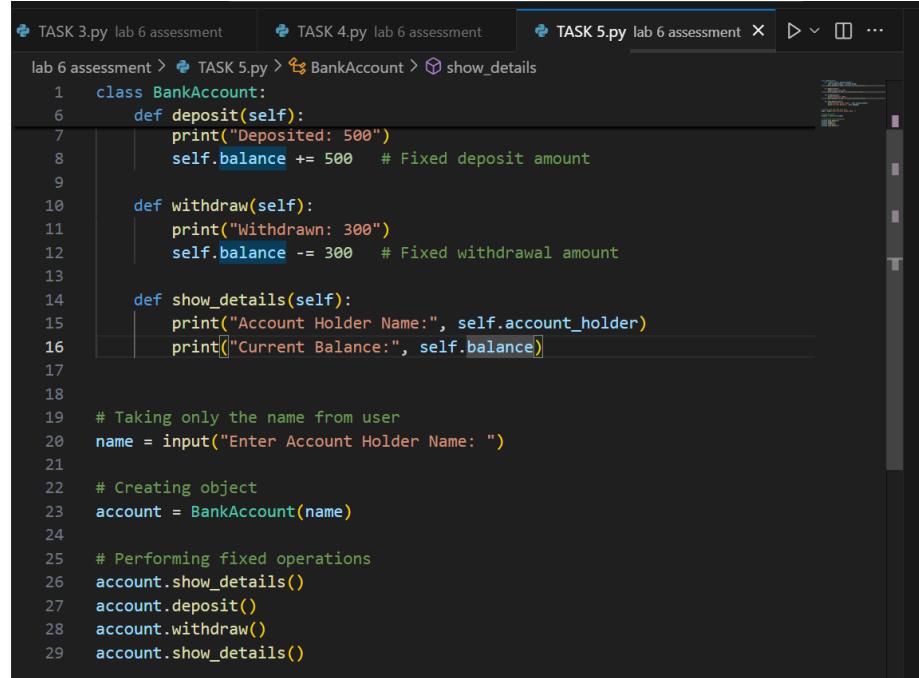


```
> TERMINAL
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneD
rive\Desktop\rimsha python assessment\tempCodeRunnerFile.python"
Enter a number: 6
Sum of first 6 numbers is: 21
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment>
```

## Task Description#5 (Class)

- Use AI to build a BankAccount class with deposit, withdraw, and balance methods.
- Analyze the generated code
- Add comments and explain code

**PROMPT:** Create a Python BankAccount class with deposit, withdraw, and check\_balance methods.

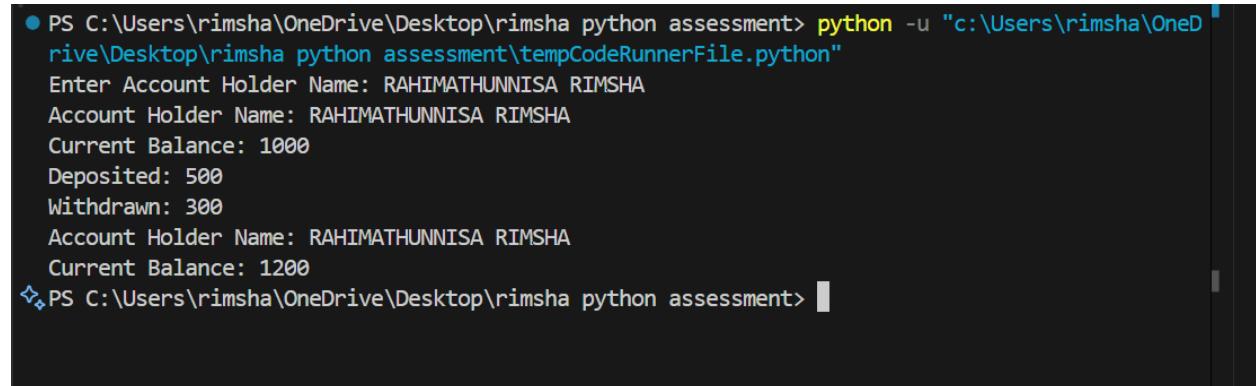


```
1  class BankAccount:
2      def deposit(self):
3          print("Deposited: 500")
4          self.balance += 500 # Fixed deposit amount
5
6      def withdraw(self):
7          print("Withdrawn: 300")
8          self.balance -= 300 # Fixed withdrawal amount
9
10     def show_details(self):
11         print("Account Holder Name:", self.account_holder)
12         print("Current Balance:", self.balance)
13
14
15 # Taking only the name from user
16 name = input("Enter Account Holder Name: ")
17
18
19 # Creating object
20 account = BankAccount(name)
21
22 # Performing fixed operations
23 account.show_details()
24 account.deposit()
25 account.withdraw()
26 account.show_details()
```

## Expected Output#5

- Python code with explanation

**Practical output:**



```
● PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneDrive\Desktop\rimsha python assessment\tempCodeRunnerFile.python"
Enter Account Holder Name: RAHIMATHUNNISA RIMSHA
Account Holder Name: RAHIMATHUNNISA RIMSHA
Current Balance: 1000
Deposited: 500
Withdrawn: 300
Account Holder Name: RAHIMATHUNNISA RIMSHA
Current Balance: 1200
↳ PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment>
```