

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)	
		Dr. T. Sampath Kumar	
		Dr. Pramoda Patro	
		Dr. Brij Kishor Tiwari	
		Dr.J.Ravichander	
		Dr. Mohammand Ali Shaik	
		Dr. Anirodh Kumar	
		Mr. S.Naresh Kumar	
		Dr. RAJESH VELDULA	
		Mr. Kundhan Kumar	
		Ms. Ch.Rajitha	
		Mr. M Prakash	
		Mr. B.Raju	
		Intern 1 (Dharma teja)	
		Intern 2 (Sai Prasad)	
		Intern 3 (Sowmya)	
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	Applicable to Batches	
AssignmentNumber: 6.3(Present assignment number)/24(Total number of assignments)			
Q.No.	Question	Expected Time to complete	
1	Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals Lab Objectives: <ul style="list-style-type: none"> To explore AI-powered auto-completion features for core Python constructs. To analyze how AI suggests logic for class definitions, loops, and 	Week3 - Wednesday	

	<p>conditionals.</p> <ul style="list-style-type: none"> To evaluate the completeness and correctness of code generated by AI assistants. <p>Lab Outcomes (LOs): After completing this lab, students will be able to:</p> <ul style="list-style-type: none"> Use AI tools to generate and complete class definitions and methods. Understand and assess AI-suggested loops for iterative tasks. Generate conditional statements through prompt-driven suggestions. Critically evaluate AI-assisted code for correctness and clarity. <p>Task Description#1 (Classes)</p> <ul style="list-style-type: none"> Use AI to complete a Student class with attributes and a method. Check output Analyze the code generated by AI tool <p>Expected Output#1</p> <ul style="list-style-type: none"> Class with constructor and display_details() method <p>Task Description#2 (Loops)</p> <ul style="list-style-type: none"> 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 <p>Expected Output#2</p> <ul style="list-style-type: none"> Correct loop-based implementation <p>Task Description#3 (Conditional Statements)</p> <ul style="list-style-type: none"> 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 <p>Expected Output#3</p> <ul style="list-style-type: none"> Age classification function with appropriate conditions and with explanation <p>Task Description#4 (For and While loops)</p> <ul style="list-style-type: none"> 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 <p>Expected Output#4</p> <ul style="list-style-type: none"> Python code with explanation <p>Task Description#5 (Class)</p> <ul style="list-style-type: none"> Use AI to build a BankAccount class with deposit, withdraw, and 	
--	--	--

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

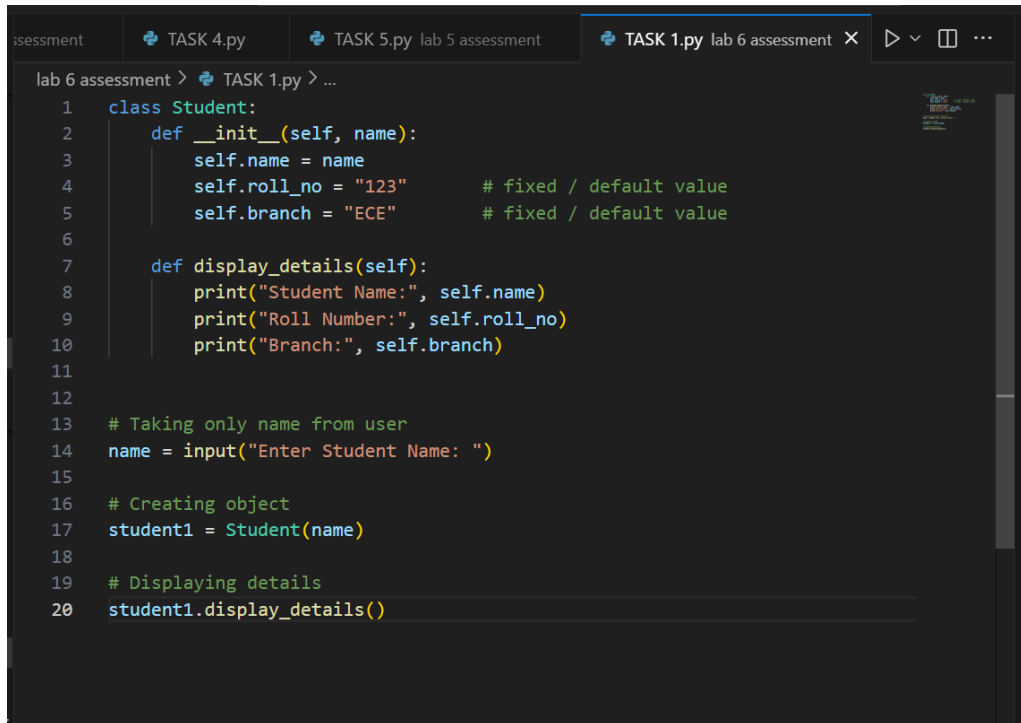
Evaluation Criteria:

Criteria	Max Marks
Class	1.0
Loops	1.0
Conditional Statements	0.5
Total	2.5 Marks

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.

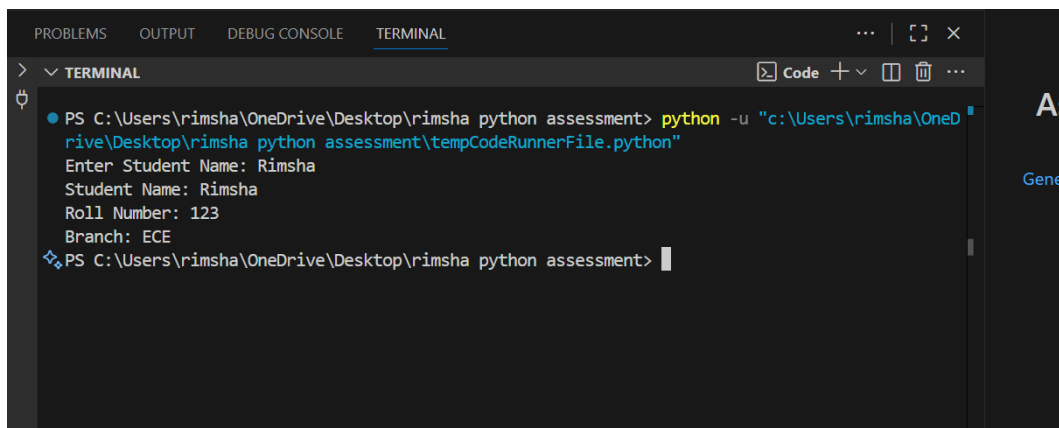


```
lab 6 assessment > TASK 1.py > ...
1  class Student:
2      def __init__(self, name):
3          self.name = name
4          self.roll_no = "123"      # fixed / default value
5          self.branch = "ECE"      # fixed / default value
6
7      def display_details(self):
8          print("Student 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:

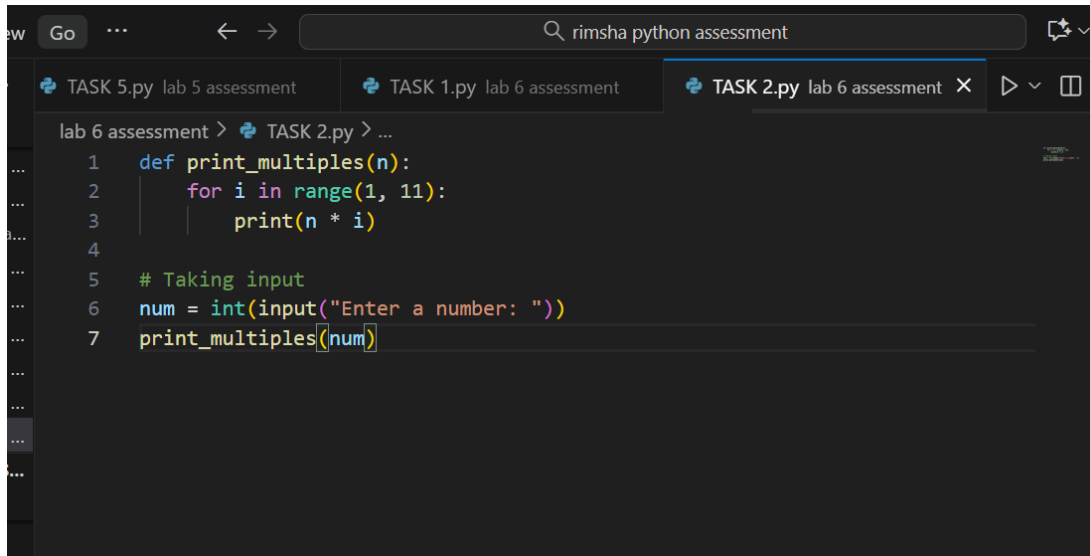


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
> TERMINAL
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneD
rive\Desktop\rimsha python assessment\tempCodeRunnerFile.python"
Enter Student Name: Rimsha
Student Name: Rimsha
Roll Number: 123
Branch: ECE
PS C:\Users\rimsha\OneDrive\Desktop\rimsha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



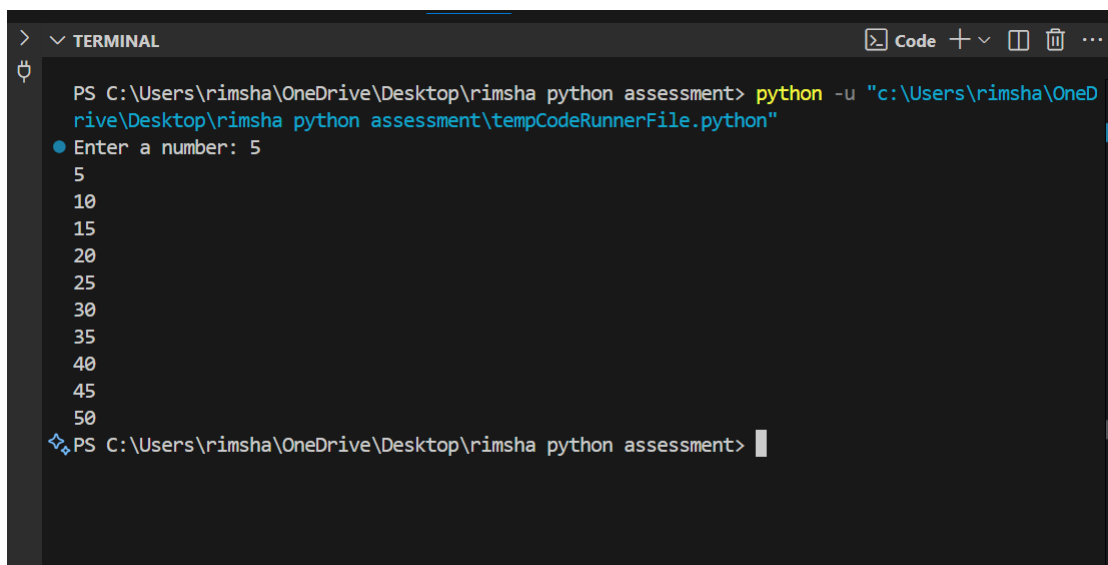
The screenshot shows a code editor with three tabs: 'TASK 5.py lab 5 assessment', 'TASK 1.py lab 6 assessment', and 'TASK 2.py lab 6 assessment'. The active tab is 'TASK 2.py lab 6 assessment'. The code in the editor is as follows:

```
lab 6 assessment > TASK 2.py > ...
1  def print_multiples(n):
2      for i in range(1, 11):
3          print(n * i)
4
5  # Taking input
6  num = int(input("Enter a number: "))
7  print_multiples(num)
```

Expected Output#2

- Correct loop-based implementation

Practical output:



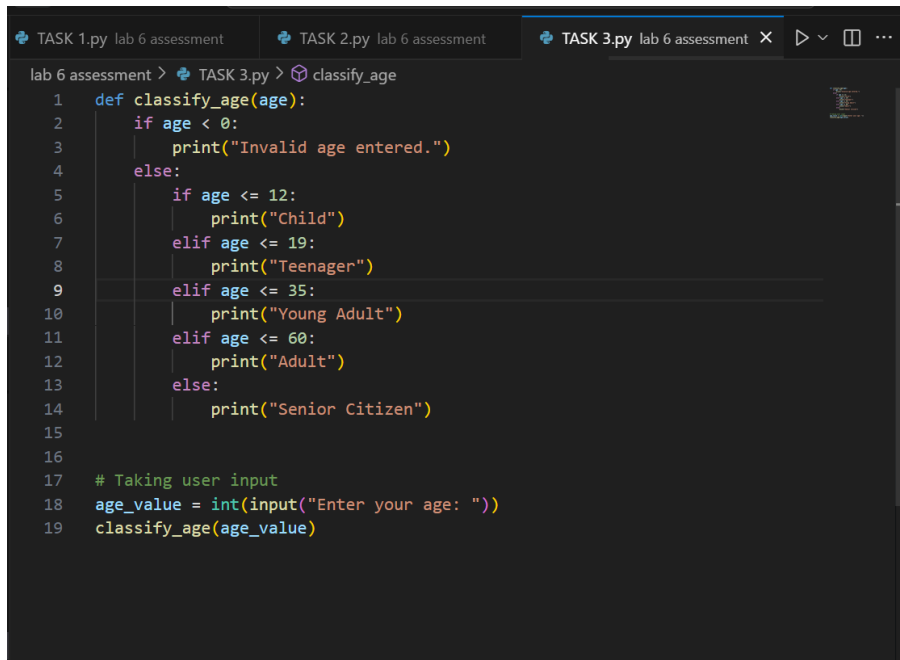
The screenshot shows a terminal window with the following output:

```
> TERMINAL
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

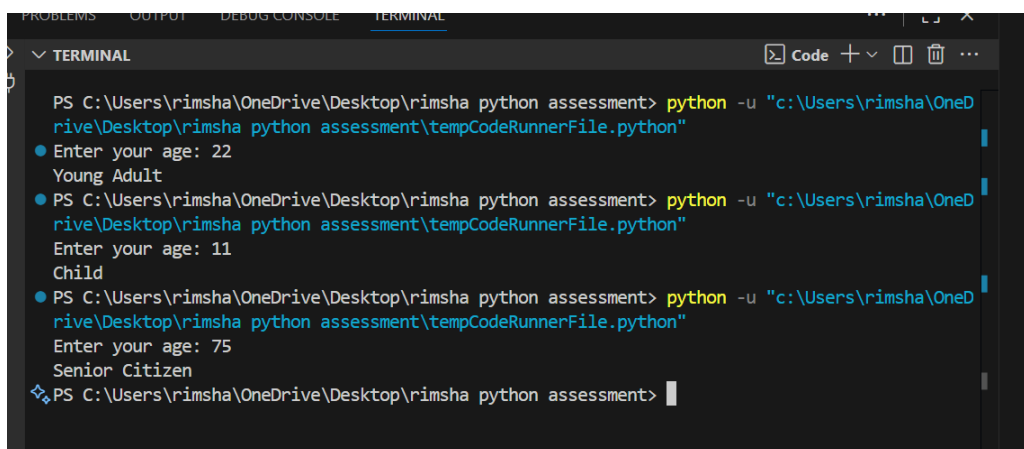


```
lab 6 assessment > TASK 3.py lab 6 assessment x TASK 3.py lab 6 assessment x TASK 3.py lab 6 assessment x
1 def classify_age(age):
2     if age < 0:
3         print("Invalid age entered.")
4     else:
5         if age <= 12:
6             print("Child")
7         elif age <= 19:
8             print("Teenager")
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:

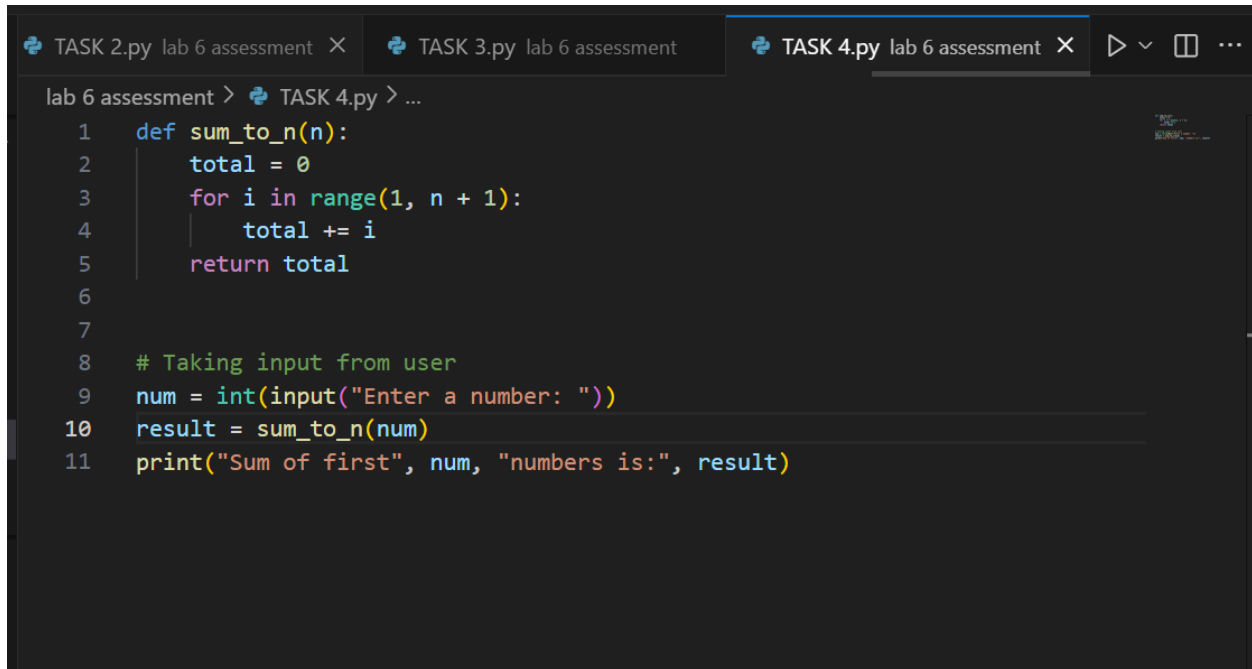


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code + - [ ] [ ] ...
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneDrive\Desktop\rimsha python assessment\tempCodeRunnerFile.python"
Enter your age: 22
Young Adult
PS C:\Users\rimsha\OneDrive\Desktop\rimsha python 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.

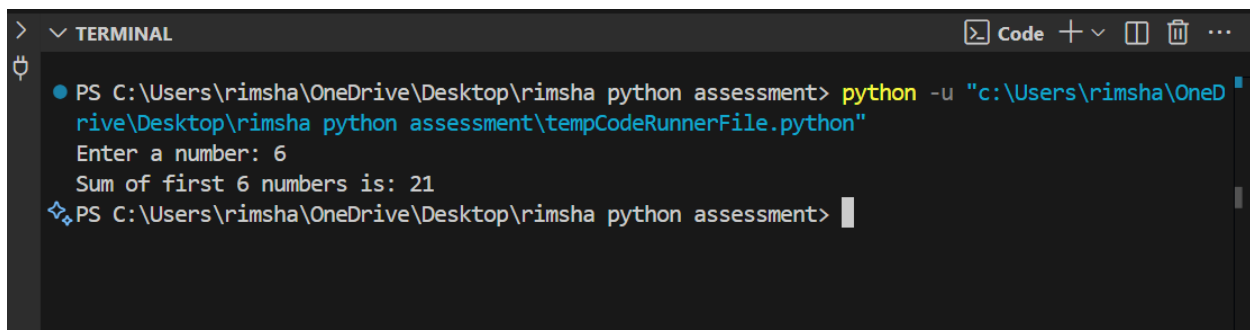


```
lab 6 assessment > TASK 4.py > ...
1  def sum_to_n(n):
2      total = 0
3      for i in range(1, n + 1):
4          total += i
5      return total
6
7
8  # Taking input from user
9  num = int(input("Enter a number: "))
10 result = sum_to_n(num)
11 print("Sum of first", num, "numbers is:", result)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

Expected Output#4

- Python code with explanation

Practical output:



```
> ▾ TERMINAL
● PS C:\Users\rimsha\OneDrive\Desktop\rimsha python assessment> python -u "c:\Users\rimsha\OneDrive\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.

```
lab 6 assessment > TASK 5.py > BankAccount > show_details
1  class BankAccount:
6      def deposit(self):
7          print("Deposited: 500")
8          self.balance += 500  # Fixed deposit amount
9
10     def withdraw(self):
11         print("Withdrawn: 300")
12         self.balance -= 300  # Fixed withdrawal amount
13
14     def show_details(self):
15         print("Account Holder Name:", self.account_holder)
16         print("Current Balance:", self.balance)
17
18
19 # Taking only the name from user
20 name = input("Enter Account Holder Name: ")
21
22 # Creating object
23 account = BankAccount(name)
24
25 # Performing fixed operations
26 account.show_details()
27 account.deposit()
28 account.withdraw()
29 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>
```