

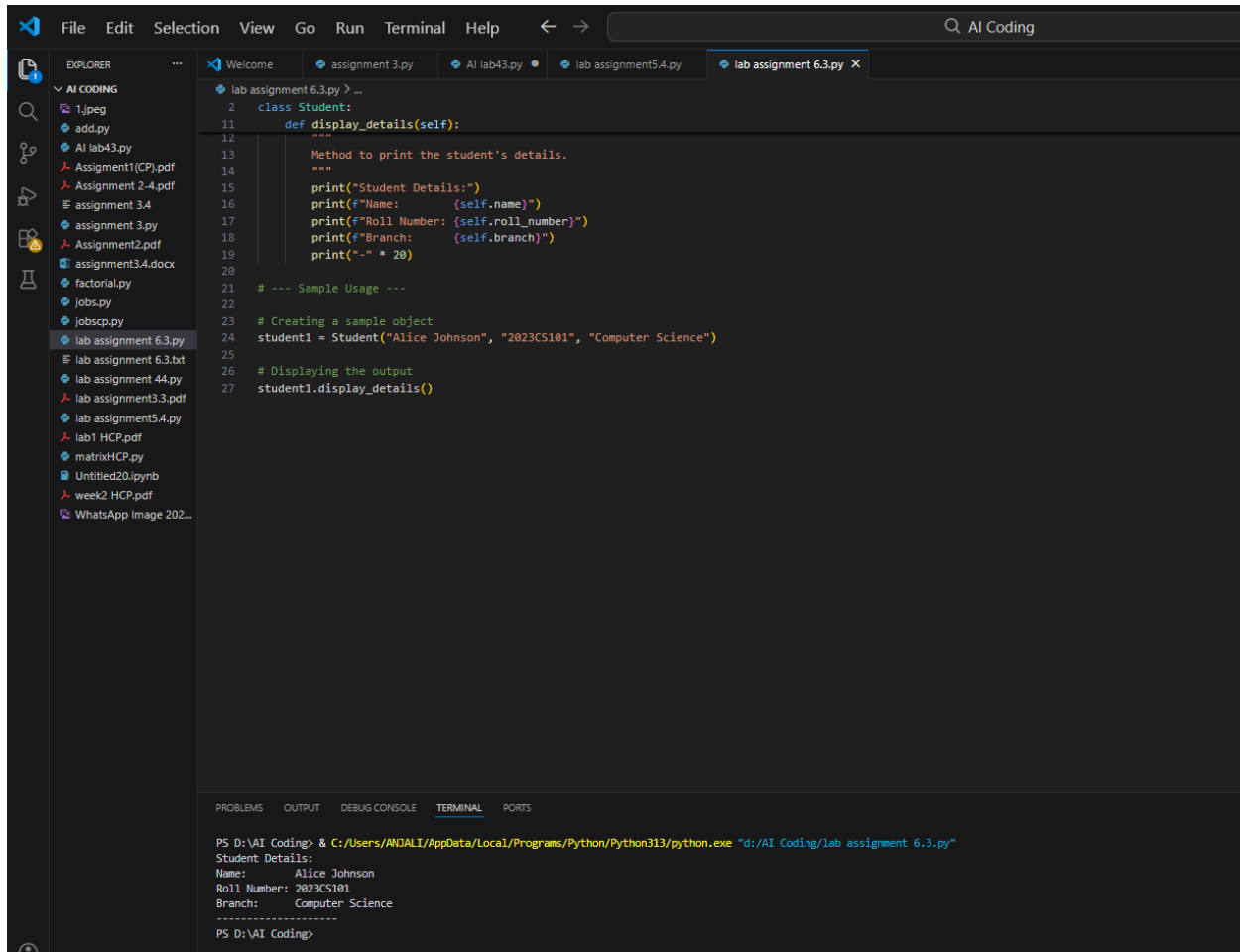
2303a51924

## Lab Assignment 6.3

### Task Description #1: Classes (Student Class)

#### Scenario

You are developing a simple student information management module.



```
File Edit Selection View Go Run Terminal Help
lab assignment 6.3.py > ...
2 class Student:
11     def display_details(self):
12         """
13         Method to print the student's details.
14         """
15         print("Student Details:")
16         print(f"Name: {self.name}")
17         print(f"Roll Number: {self.roll_number}")
18         print(f"Branch: {self.branch}")
19         print("-" * 20)
20
21 # --- Sample Usage ---
22
23 # Creating a sample object
24 student1 = Student("Alice Johnson", "2023CS101", "Computer Science")
25
26 # Displaying the output
27 student1.display_details()
```

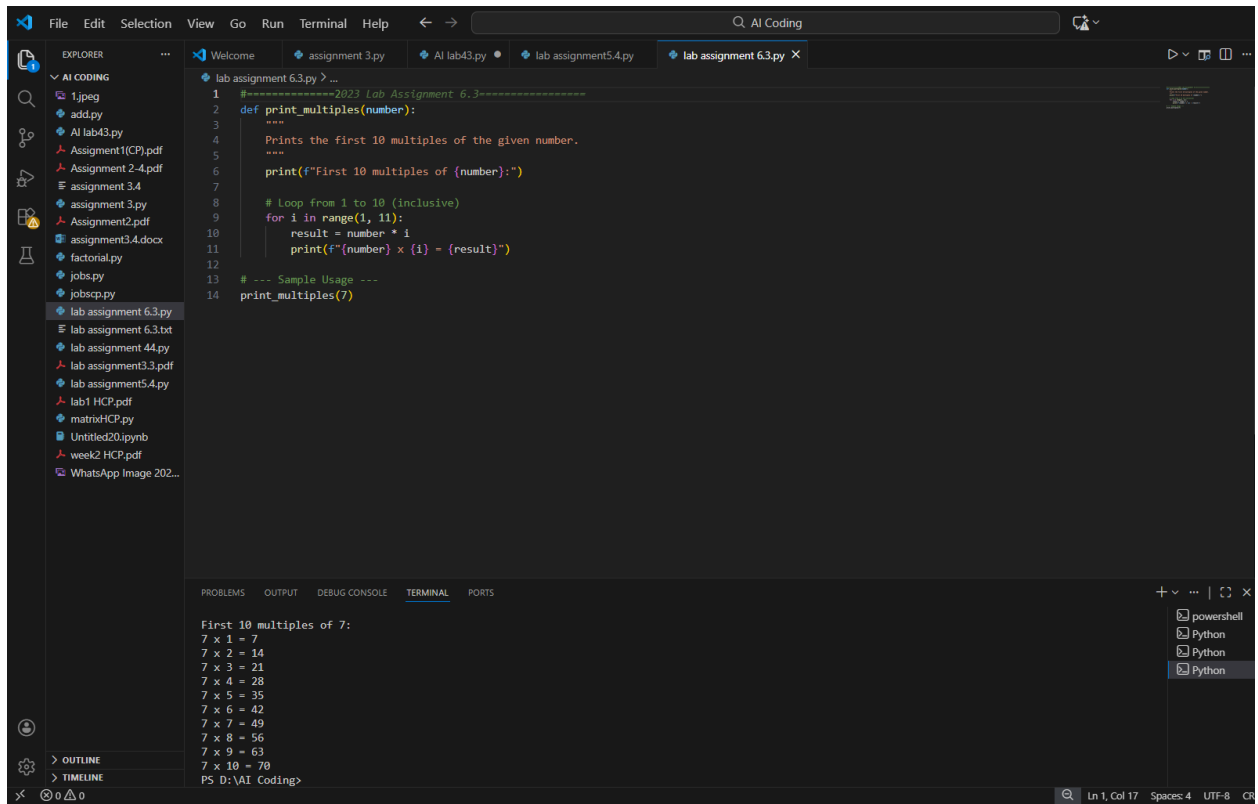
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 6.3.py"
Student Details:
Name:      Alice Johnson
Roll Number: 2023CS101
Branch:    Computer Science
-----
PS D:\AI Coding>
```

## Task Description #2: Loops (Multiples of a Number)

### Scenario

You are writing a utility function to display multiples of a given number.



The screenshot shows a Visual Studio Code editor with a Python file named 'lab assignment 6.3.py' open. The code defines a function 'print\_multiples' that prints the first 10 multiples of a given number. The function uses a for loop to iterate from 1 to 10, calculating the product of the number and the loop index, and printing the result. A sample usage is provided, calling the function with the number 7. The terminal at the bottom shows the output of the program, displaying the first 10 multiples of 7.

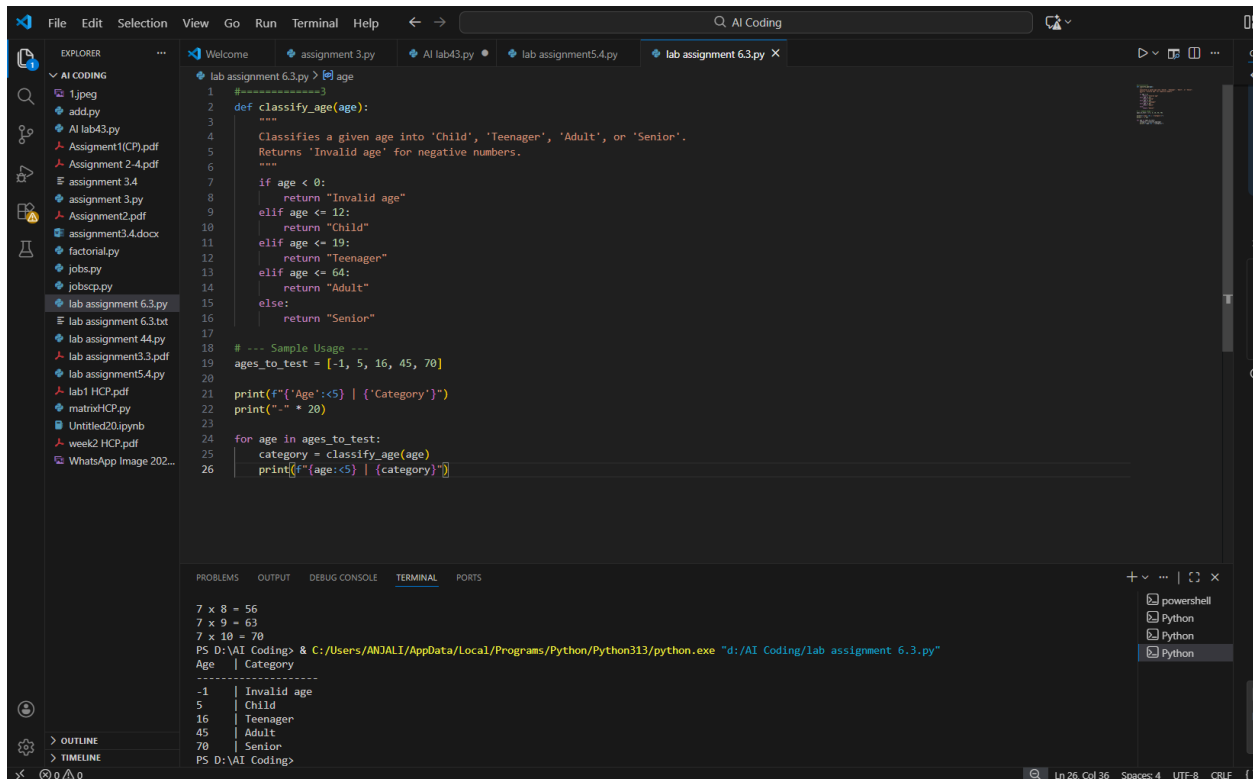
```
1 #-----2023 Lab Assignment 6.3-----
2 def print_multiples(number):
3     """
4     Prints the first 10 multiples of the given number.
5     """
6     print(f"First 10 multiples of {number}:")
7
8     # Loop from 1 to 10 (inclusive)
9     for i in range(1, 11):
10         result = number * i
11         print(f"{number} x {i} = {result}")
12
13 # --- Sample Usage ---
14 print_multiples(7)
```

First 10 multiples of 7:  
7 x 1 = 7  
7 x 2 = 14  
7 x 3 = 21  
7 x 4 = 28  
7 x 5 = 35  
7 x 6 = 42  
7 x 7 = 49  
7 x 8 = 56  
7 x 9 = 63  
7 x 10 = 70

### Task Description #3: Conditional Statements (Age Classification)

#### Scenario

You are building a basic classification system based on age.



The screenshot shows a Visual Studio Code editor with a Python file named 'lab assignment 6.3.py' open. The code defines a function 'classify\_age' that takes an age as input and returns a classification string based on the age. The function uses conditional statements to handle different age ranges: 'Invalid age' for negative numbers, 'Child' for ages 1-12, 'Teenager' for ages 13-19, 'Adult' for ages 20-64, and 'Senior' for ages 65 and above. Below the function definition, there is a sample usage section that defines a list of ages to test and prints the results of the classification function for each age.

```
1 #-----3
2
3 def classify_age(age):
4     """
5     Classifies a given age into 'Child', 'Teenager', 'Adult', or 'Senior'.
6     Returns 'Invalid age' for negative numbers.
7     """
8     if age < 0:
9         return "Invalid age"
10    elif age <= 12:
11        return "Child"
12    elif age <= 19:
13        return "Teenager"
14    elif age <= 64:
15        return "Adult"
16    else:
17        return "Senior"
18
19 # --- Sample Usage ---
20 ages_to_test = [-1, 5, 16, 45, 70]
21
22 print(f"{'Age':<5} | {'Category':>10}")
23 print("-" * 20)
24
25 for age in ages_to_test:
26     category = classify_age(age)
27     print(f"{'Age':<5} | {category:>10}")
```

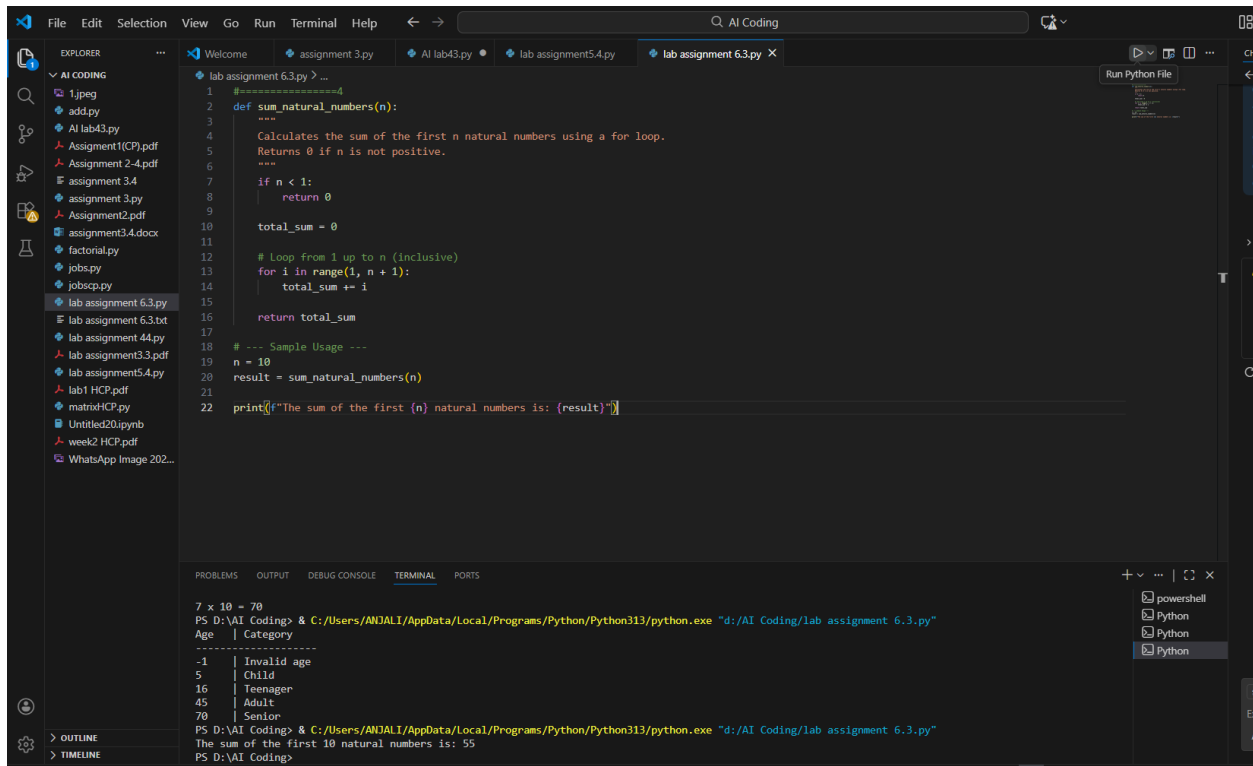
The terminal output shows the execution of the script, displaying the classification results for each age in the test list:

```
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
PS D:\VAI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 6.3.py"
Age | Category
-----
-1 | Invalid age
5 | Child
16 | Teenager
45 | Adult
70 | Senior
PS D:\VAI Coding>
```

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

### Scenario

You need to calculate the sum of the first n natural numbers.



The screenshot shows a Visual Studio Code editor with a Python file named 'lab assignment 6.3.py' open. The code defines a function 'sum\_natural\_numbers(n)' that calculates the sum of the first n natural numbers using a for loop. The function returns 0 for n < 1. A sample usage is provided with n = 10, and the result is printed. The terminal output shows the execution of the script, which prints 'The sum of the first 10 natural numbers is: 55'.

```
1 #####
2 def sum_natural_numbers(n):
3     """
4     Calculates the sum of the first n natural numbers using a for loop.
5     Returns 0 if n is not positive.
6     """
7     if n < 1:
8         return 0
9
10    total_sum = 0
11
12    # Loop from 1 up to n (inclusive)
13    for i in range(1, n + 1):
14        total_sum += i
15
16    return total_sum
17
18 # --- Sample Usage ---
19 n = 10
20 result = sum_natural_numbers(n)
21
22 print(f"The sum of the first {n} natural numbers is: {result}")
```

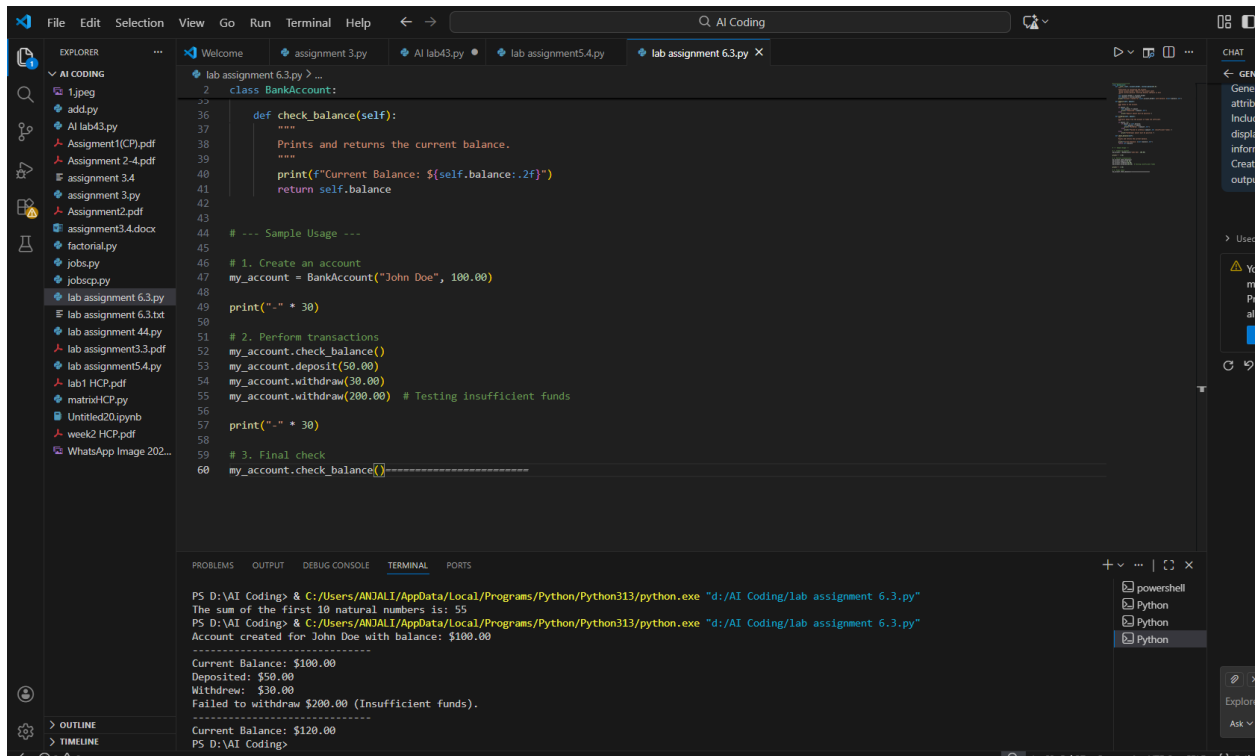
Terminal Output:

```
7 x 10 = 70
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 6.3.py"
Age | Category
-----
-1 | Invalid age
5 | Child
16 | Teenager
45 | Adult
70 | Senior
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 6.3.py"
The sum of the first 10 natural numbers is: 55
PS D:\AI Coding>
```

## Task Description #5: Classes (Bank Account Class)

### Scenario

You are designing a basic banking application.



The screenshot shows a Visual Studio Code editor with a Python file named 'lab assignment 6.3.py'. The code defines a 'BankAccount' class with a 'check\_balance' method. The terminal output shows the execution of the script, which creates an account for 'John Doe' with a balance of \$100.00, performs a deposit of \$50.00, and a withdrawal of \$30.00, resulting in a final balance of \$120.00.

```
2 class BankAccount:
3     def check_balance(self):
4         """
5         Prints and returns the current balance.
6         """
7         print(f"Current Balance: ${self.balance:.2f}")
8         return self.balance
9
10 # --- Sample Usage ---
11 # 1. Create an account
12 my_account = BankAccount("John Doe", 100.00)
13
14 print("-" * 30)
15
16 # 2. Perform transactions
17 my_account.check_balance()
18 my_account.deposit(50.00)
19 my_account.withdraw(30.00)
20 my_account.withdraw(200.00) # Testing insufficient funds
21
22 print("-" * 30)
23
24 # 3. Final check
25 my_account.check_balance()
```

Terminal Output:

```
PS D:\AI Coding> C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 6.3.py"
The sum of the first 10 natural numbers is: 55
PS D:\AI Coding> C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment 6.3.py"
-----
Current Balance: $100.00
Deposited: $50.00
Withdraw: $30.00
Failed to withdraw $200.00 (Insufficient funds).
-----
Current Balance: $120.00
PS D:\AI Coding>
```