

AI ASSISTED CODING

Name:- J.Sai Kumar || Batch:09 || Ht.no:2303A51562

Task 1: Student Performance Evaluation System.

```
Lab-6.4.py U X
1 #Task:-1 Student Performance Evaluation System
2 #Create a class named Student with attributes for name, roll number, and marks.
3
4 class Student:
5     def __init__(self, name, roll_number, marks):
6         self.name = name
7         self.roll_number = roll_number
8         self.marks = marks
9
10    # Method to display student details
11    def display_student_details(self):
12        print(f"Name: {self.name}")
13        print(f"Roll Number: {self.roll_number}")
14        print(f"Marks: {self.marks}")
15
16    # Method to check if student marks are above class average
17    def check_performance(self, class_average):
18        if self.marks > class_average:
19            return f"{self.name} is performing above the class average."
20        elif self.marks == class_average:
21            return f"{self.name} is performing at the class average."
22        else:
23            return f"{self.name} is performing below the class average."
24
25    # Sample usage
26    if __name__ == "__main__":
27        student1 = Student("Alice", "S001", 85)
28        student2 = Student("Bob", "S002", 72)
29        student3 = Student("Charlie", "S003", 90)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

Python + v [Icons] [Refresh] [Close]

```
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/saiku/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/saiku/OneDrive/Desktop/AI ASSISTANT CODING/Lab-6.4.py"
Name: Alice
Roll Number: S001
Marks: 85

Alice is performing above the class average.
Bob is performing below the class average.
Charlie is performing above the class average.
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING>
```

Task 2: Data Processing in a Monitoring System.

The image shows a Python IDE window titled "Lab-6.4.py U". The code editor contains a script for processing sensor readings. Below the editor is a terminal window showing the execution of the script. The output displays the square of each even number from the provided list.

```
1 #Task 2: Data Processing in a Monitoring System.  
2 # Sensor readings data  
3 sensor_readings = [12, 7, 24, 15, 36, 9, 48, 3, 18, 22]  
4  
5 # Iterate over sensor readings and process even numbers  
6 # For each reading, identify even numbers using modulus operator,  
7 # calculate their square, and print in format "Reading: X, Square: Y"  
8 for reading in sensor_readings:  
9     if reading % 2 == 0:  
10         square = reading ** 2  
11         print(f"Reading: {reading}, Square: {square}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\saiiku\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/saiiku/AppData/Local/Programs/Python/Python314/python.exe /Users/saiiku/OneDrive/Desktop/AI ASSISTANT CODING/Lab-6.4.py"

Reading: 12, Square: 144
Reading: 24, Square: 576
Reading: 36, Square: 1296
Reading: 48, Square: 2304
Reading: 18, Square: 324
Reading: 22, Square: 484

PS C:\Users\saiiku\OneDrive\Desktop\AI ASSISTANT CODING>

Task 3: Banking Transaction Simulation.

```
Lab-6.4.py U X
Lab-6.4.py > ...
1 #Task 3: Banking Transaction Simulation
2 #create a structure of a Python class named BankAccount with attributes:account_holder and balance.
3 class BankAccount:
4     def __init__(self, account_holder, balance=0):
5         self.account_holder = account_holder
6         self.balance = balance
7
8     def deposit(self, amount):
9         if amount > 0:
10             self.balance += amount
11             print(f"Deposit successful! Current balance: ₹{self.balance}")
12         else:
13             print("Deposit amount must be greater than zero.")
14
15     def withdraw(self, amount):
16         if amount <= 0:
17             print("Withdrawal amount must be greater than zero.")
18         elif amount <= self.balance:
19             self.balance -= amount
20             print(f"Withdrawal successful! Remaining balance: ₹{self.balance}")
21         else:
22             print("Insufficient balance. Withdrawal denied.")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\saiiku\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/saiku/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/saiiku/OneDrive/Desktop/AI ASSISTANT CODING/Lab-6.4.py"

● Enter account holder name: sai kumar

--- Banking Menu ---
1. Deposit
2. Withdraw
3. Exit
Enter your choice: 2
Enter withdrawal amount: 20000
Insufficient balance. Withdrawal denied.

--- Banking Menu ---
1. Deposit
2. Withdraw
3. Exit
Enter your choice: 3
Thank you for using the banking system.

PS C:\Users\saiiku\OneDrive\Desktop\AI ASSISTANT CODING>

Task 4: Student Scholarship Eligibility Check.

Lab-6.4.py U X

Lab-6.4.py > ...

```
1  # Task 4: Student Scholarship Eligibility Check
2
3  # Initialize list of students
4  students = [
5      {"name": "Rahul", "score": 85},
6      {"name": "Anita", "score": 72},
7      {"name": "Suresh", "score": 90},
8      {"name": "Meena", "score": 78},
9      {"name": "Kiran", "score": 65}
10 ]
11
12 print("Eligible students for merit-based scholarship:")
13
14 # Iterate through the list using a while loop
15 index = 0
16 while index < len(students):
17     if students[index]["score"] > 75:
18         print(students[index]["name"])
19     index += 1
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING> & C:/Users/saiku/AppData/Local/Programs/Python/Python38/Python38.exe C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING/Lab-6.4.py

Eligible students for merit-based scholarship:

Rahul

○ Suresh

Meena

PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING> █

Task 5: Online Shopping Cart Module.

```
Lab-6.4.py U X
Lab-6.4.py > ...
1  # Task 5: Online Shopping Cart Module
2
3  class ShoppingCart:
4      def __init__(self):
5          # Empty list to store cart items
6          # Each item is a dictionary: name, price, quantity
7          self.items = []
8
9      def add_item(self, name, price, quantity):
10         # Add item details to the cart
11         item = {
12             "name": name,
13             "price": price,
14             "quantity": quantity
15         }
16         self.items.append(item)
17         print(f"{name} added to cart.")
18
19     def remove_item(self, name):
20         # Remove an item from the cart by name
21         for item in self.items:
22             if item["name"].lower() == name.lower():

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
--- Shopping Cart Menu ---
1. Add Item
2. Remove Item
3. View Total
4. Exit
Enter your choice: 2
Enter item name to remove: eggs
Item not found in cart.

--- Shopping Cart Menu ---
1. Add Item
2. Remove Item
3. View Total
4. Exit
Enter your choice: 4
Thank you for shopping with us!
PS C:\Users\saiku\OneDrive\Desktop\AI ASSISTANT CODING>
```

0 Ln 74, Col 1 Spaces: