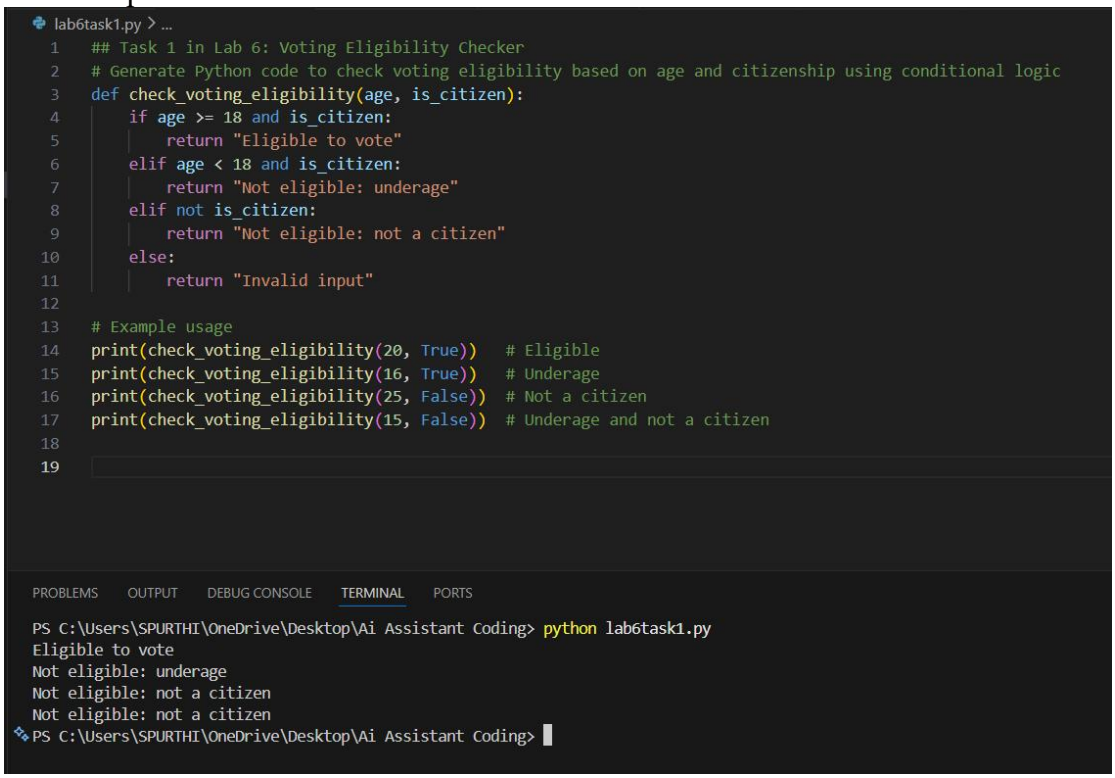
AI Assisted Coding
2303A51802
Billa Spurthi
Batch 29

Assignment Number: 6.5

Task 1 Prompt:
"Generate Python code to check voting eligibility based on age and citizenship."



Explanation

`def check_voting_eligibility(age, is_citizen):` → Defines a function with two inputs: age (number) and citizenship (boolean).

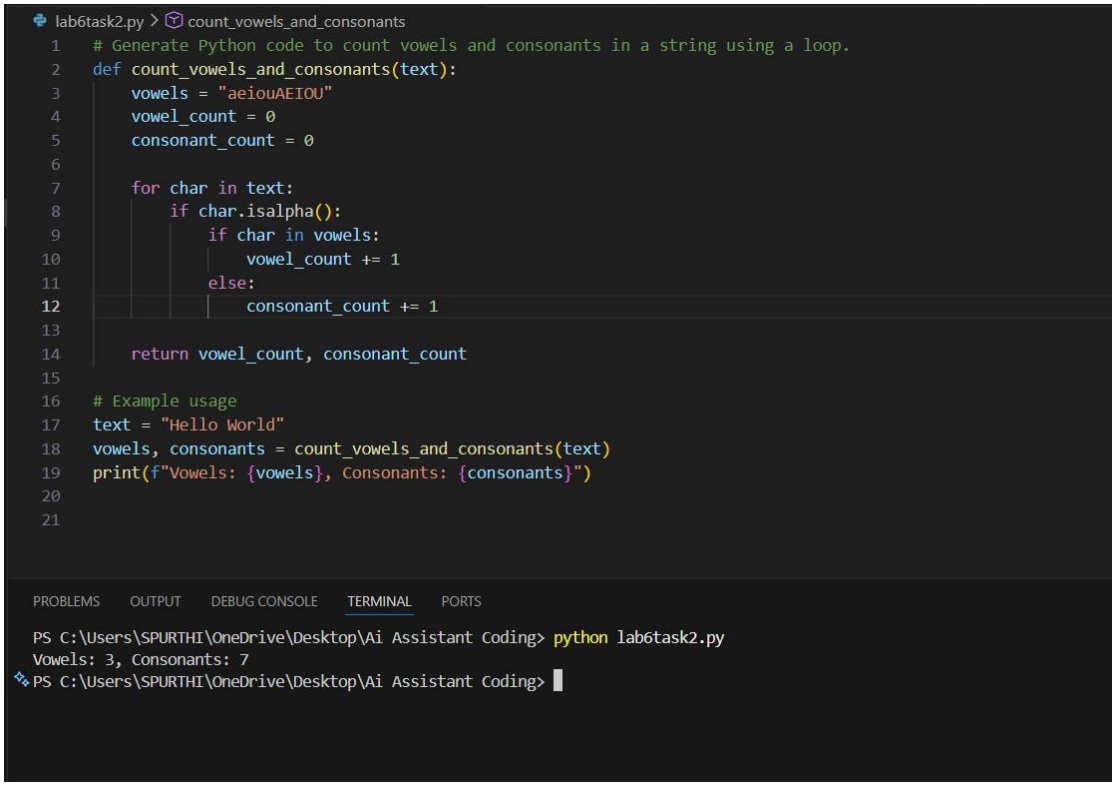
`if age >= 18 and is_citizen:` → Checks if the person is at least 18 **and** a citizen. If true, they can vote.

`elif age < 18 and is_citizen:` → If they are a citizen but under 18, they're not eligible.

`elif not is_citizen:` → If they're not a citizen, they're not eligible regardless of age.

`else:` → Handles invalid inputs (e.g., negative age).

Task 2 Promt
“Generate Python code to count vowels and consonants in a string using a loop.”



Explanation

`vowels = "aeiouAEIOU"` → Defines which letters are vowels.

`for char in text:` → Loops through each character in the string.

`if char.isalpha():` → Ensures only letters are counted (ignores spaces, punctuation).

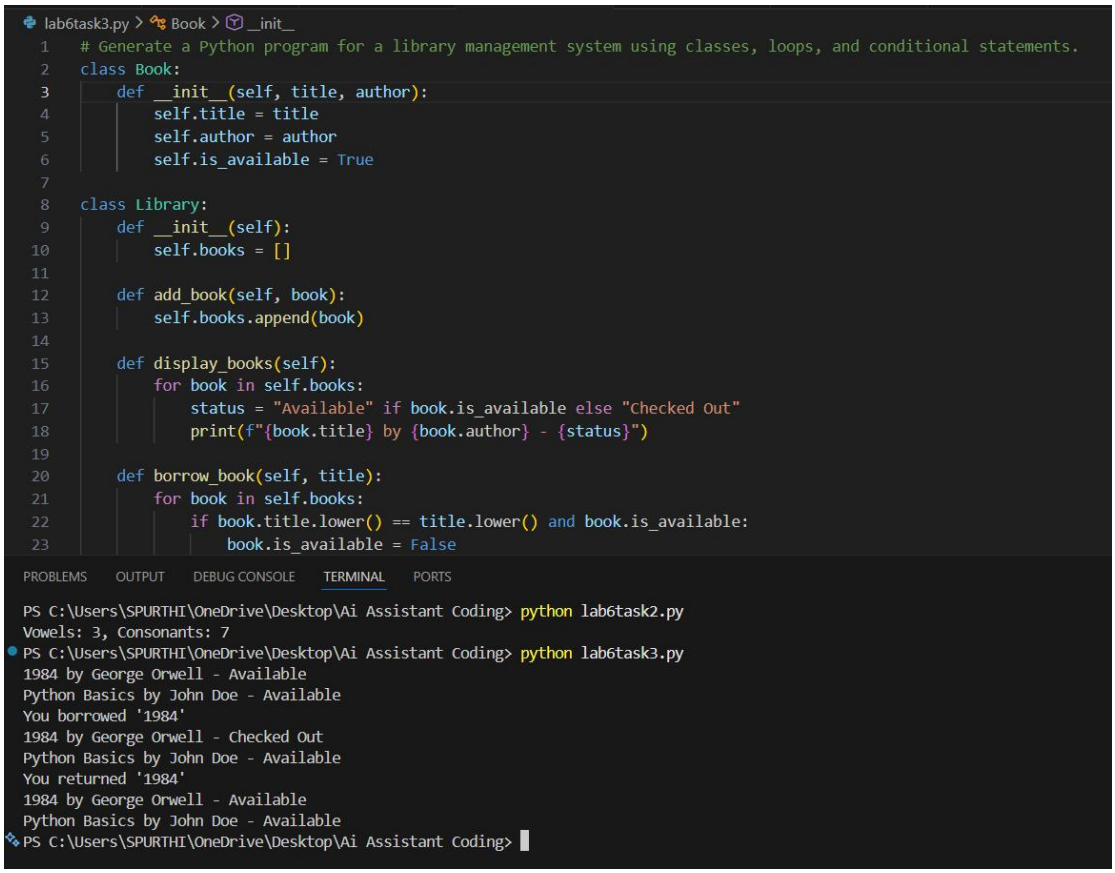
`if char in vowels:` → If the character is a vowel, increment `vowel_count`.

`else:` → Otherwise, it’s a consonant, so increment `consonant_count`.

Returns both counts.

Task 3 Prompt

“Generate a Python program for a library management system using classes, loops, and conditional statements.”



Explanation

Class `Book` → Represents a book with attributes: title, author, and availability.

Class `Library` → Manages a collection of books.

`add_book()` → Adds a book to the library.

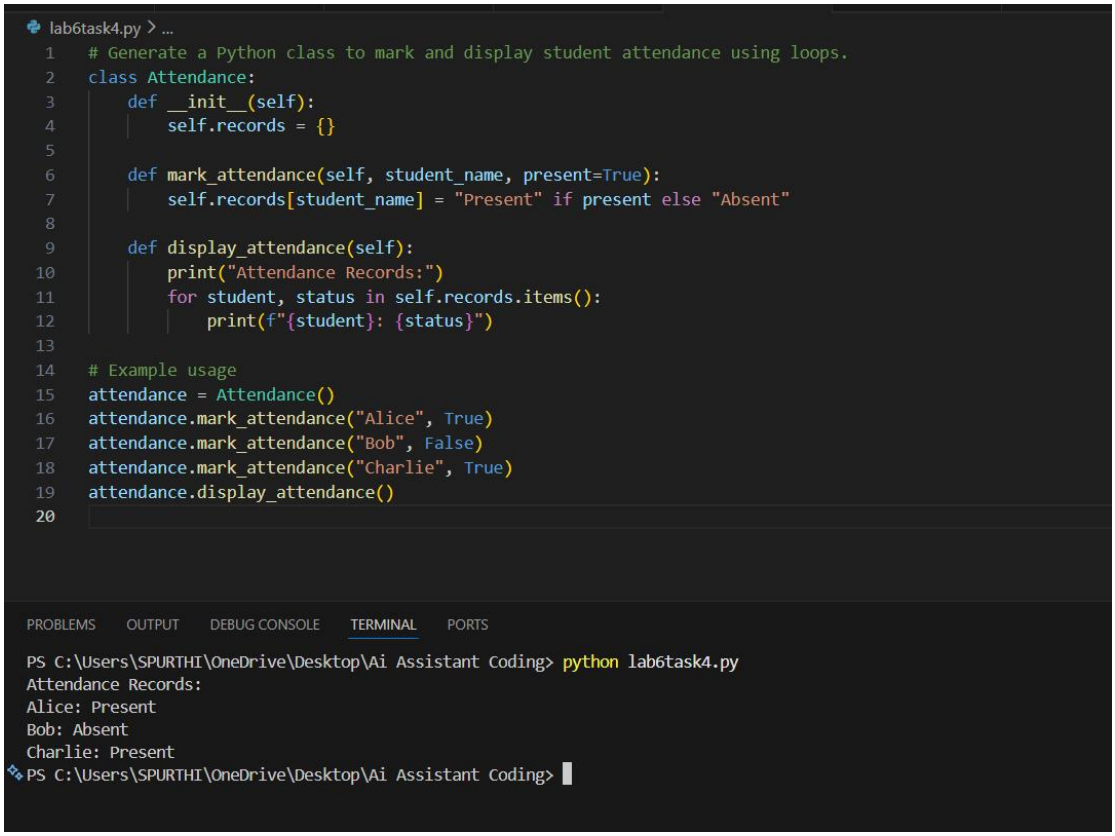
`display_books()` → Loops through all books and prints their status.

`borrow_book()` → Uses conditionals to check if a book is available before borrowing.

`return_book()` → Marks a borrowed book as available again.

Task 4 Prompt

Generate a Python class to mark and display student attendance using loops.



Explanation

`self.records = {}` → A dictionary to store student names and their status.

`mark_attendance()` → Adds or updates a student's attendance (default is "Present").

`display_attendance()` → Loops through the dictionary and prints each student's status.

Task 5 Prompt

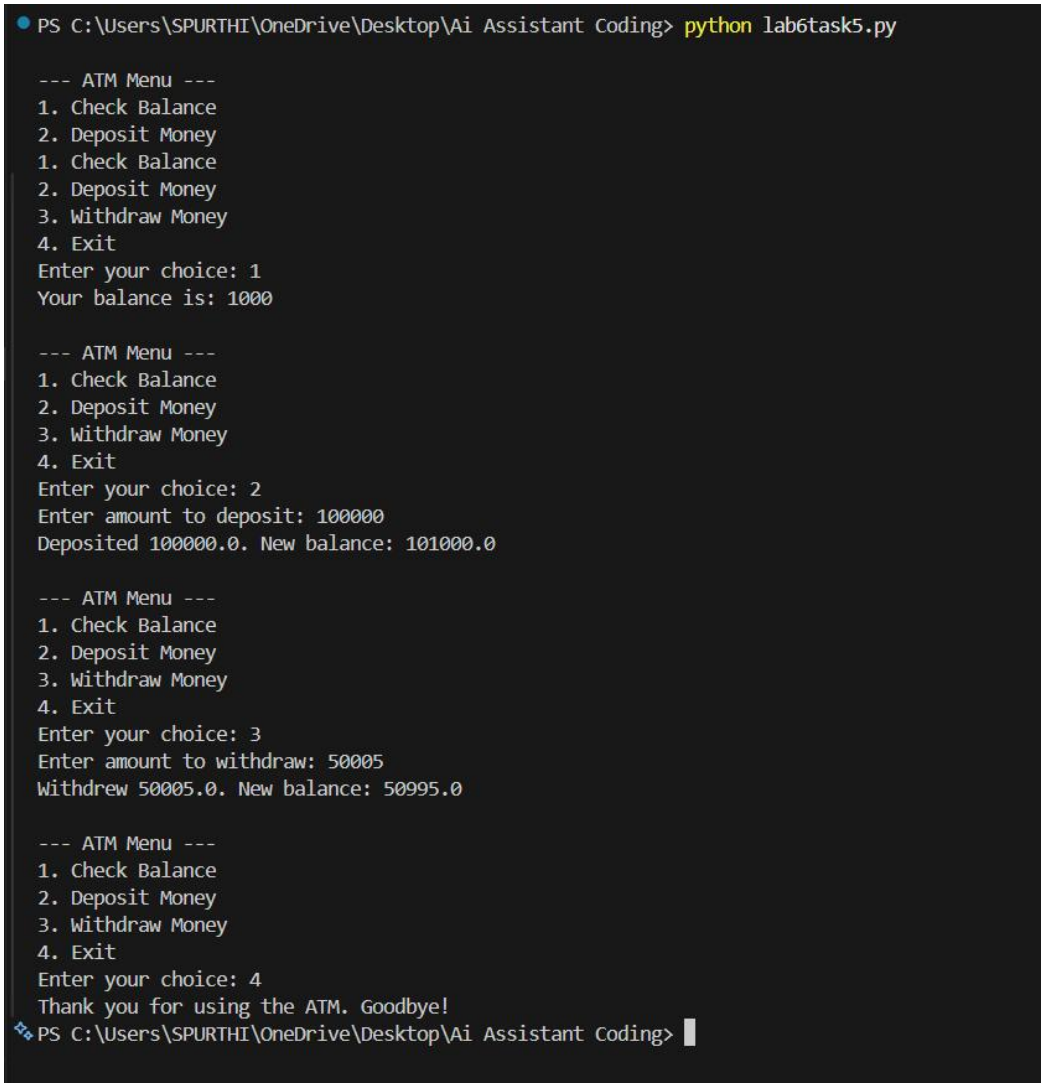
“Generate a Python program using loops and conditionals to simulate an ATM menu.”

lab6task5.py > atm_menu

```python
# Generate a Python program using loops and conditionals to simulate an ATM menu.
def atm_menu():
    balance = 1000  # initial balance
    while True:
        print("\n--- ATM Menu ---")
        print("1. Check Balance")
        print("2. Deposit Money")
        print("3. Withdraw Money")
        print("4. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            print(f"Your balance is: {balance}")
        elif choice == "2":
            amount = float(input("Enter amount to deposit: "))
            balance += amount
            print(f"Deposited {amount}. New balance: {balance}")
        elif choice == "3":
            amount = float(input("Enter amount to withdraw: "))
            if amount <= balance:
                balance -= amount
                print(f"Withdrew {amount}. New balance: {balance}")
            else:
                print("Insufficient funds!")
        elif choice == "4":
            print("Thank you for using the ATM. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")
atm_menu()
```



Explanation

`balance = 1000` → Starts with a default balance.

`while True:` → Creates an infinite loop until the user chooses to exit.

Menu options are printed each time.

`choice = input(...)` → Reads user input.

`if choice == "1":` → Shows balance.

`elif choice == "2":` → Deposits money (adds to balance).

`elif choice == "3":` → Withdraws money (checks if sufficient funds).

`elif choice == "4":` → Breaks the loop and exits.

`else:` → Handles invalid input.