

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.”

```
⚡ lab6task1.py > ...
1  ## Task 1 in Lab 6: Voting Eligibility Checker
2  # Generate Python code to check voting eligibility based on age and citizenship using conditional logic
3  def check_voting_eligibility(age, is_citizen):
4      if age >= 18 and is_citizen:
5          return "Eligible to vote"
6      elif age < 18 and is_citizen:
7          return "Not eligible: underage"
8      elif not is_citizen:
9          return "Not eligible: not a citizen"
10     else:
11         return "Invalid input"
12
13 # Example usage
14 print(check_voting_eligibility(20, True))  # Eligible
15 print(check_voting_eligibility(16, True))  # Underage
16 print(check_voting_eligibility(25, False)) # Not a citizen
17 print(check_voting_eligibility(15, False)) # Underage and not a citizen
18
19
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding> python lab6task1.py
Eligible to vote
Not eligible: underage
Not eligible: not a citizen
Not eligible: not a citizen
⚡ PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding>
```

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.”

```
lab6task2.py > count_vowels_and_consonants
1  # Generate Python code to count vowels and consonants in a string using a loop.
2  def count_vowels_and_consonants(text):
3      vowels = "aeiouAEIOU"
4      vowel_count = 0
5      consonant_count = 0
6
7      for char in text:
8          if char.isalpha():
9              if char in vowels:
10                  vowel_count += 1
11              else:
12                  consonant_count += 1
13
14      return vowel_count, consonant_count
15
16 # Example usage
17 text = "Hello World"
18 vowels, consonants = count_vowels_and_consonants(text)
19 print(f"Vowels: {vowels}, Consonants: {consonants}")
20
21
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding> python lab6task2.py
Vowels: 3, Consonants: 7
✿ PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding>
```

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.”

```
lab6task3.py > Book > __init__  
1 # Generate a Python program for a library management system using classes, loops, and conditional statements.  
2 class Book:  
3     def __init__(self, title, author):  
4         self.title = title  
5         self.author = author  
6         self.is_available = True  
7  
8     class Library:  
9         def __init__(self):  
10            self.books = []  
11  
12        def add_book(self, book):  
13            self.books.append(book)  
14  
15        def display_books(self):  
16            for book in self.books:  
17                status = "Available" if book.is_available else "Checked Out"  
18                print(f"{book.title} by {book.author} - {status}")  
19  
20        def borrow_book(self, title):  
21            for book in self.books:  
22                if book.title.lower() == title.lower() and book.is_available:  
23                    book.is_available = False  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding> python lab6task2.py  
Vowels: 3, Consonants: 7  
● PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding> python lab6task3.py  
1984 by George Orwell - Available  
Python Basics by John Doe - Available  
You borrowed '1984'  
1984 by George Orwell - Checked Out  
Python Basics by John Doe - Available  
You returned '1984'  
1984 by George Orwell - Available  
Python Basics by John Doe - Available  
◆ PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding>
```

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.

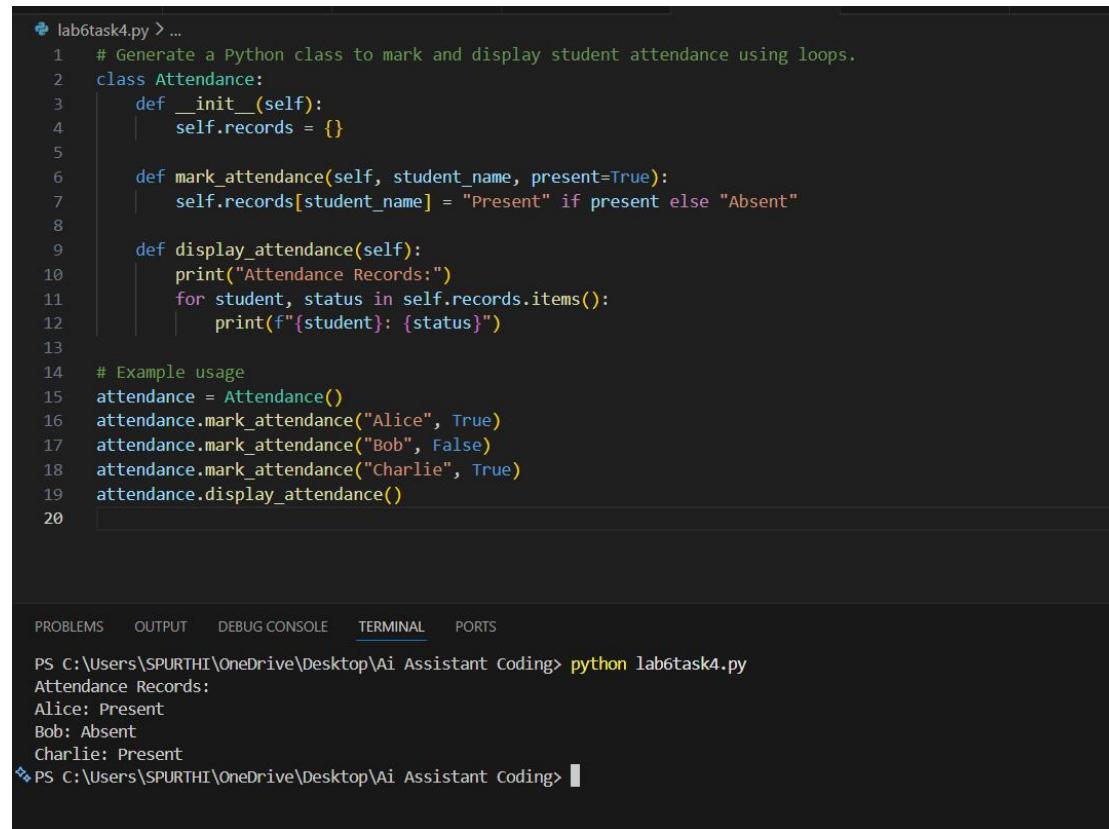
`display_books()` → Loops through all books and prints their status.

`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.



```
lab6task4.py > ...
1 # Generate a Python class to mark and display student attendance using loops.
2 class Attendance:
3     def __init__(self):
4         self.records = {}
5
6     def mark_attendance(self, student_name, present=True):
7         self.records[student_name] = "Present" if present else "Absent"
8
9     def display_attendance(self):
10        print("Attendance Records:")
11        for student, status in self.records.items():
12            print(f"{student}: {status}")
13
14 # Example usage
15 attendance = Attendance()
16 attendance.mark_attendance("Alice", True)
17 attendance.mark_attendance("Bob", False)
18 attendance.mark_attendance("Charlie", True)
19 attendance.display_attendance()
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding> python lab6task4.py
Attendance Records:
Alice: Present
Bob: Absent
Charlie: Present
PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding>
```

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
1  # Generate a Python program using loops and conditionals to simulate an ATM menu.
2  def atm_menu():
3      balance = 1000 # initial balance
4      while True:
5          print("\n--- ATM Menu ---")
6          print("1. Check Balance")
7          print("2. Deposit Money")
8          print("3. Withdraw Money")
9          print("4. Exit")
10
11         choice = input("Enter your choice: ")
12
13         if choice == "1":
14             print(f"Your balance is: {balance}")
15         elif choice == "2":
16             amount = float(input("Enter amount to deposit: "))
17             balance += amount
18             print(f"Deposited {amount}. New balance: {balance}")
19         elif choice == "3":
20             amount = float(input("Enter amount to withdraw: "))
21             if amount <= balance:
22                 balance -= amount
23                 print(f"Withdrew {amount}. New balance: {balance}")
24             else:
25                 print("Insufficient funds!")
26         elif choice == "4":
27             print("Thank you for using the ATM. Goodbye!")
28             break
29         else:
30             print("Invalid choice. Please try again.")
31     atm_menu()
```

```
● PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding> python lab6task5.py

--- ATM Menu ---
1. Check Balance
2. Deposit Money
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 1
Your balance is: 1000

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 2
Enter amount to deposit: 100000
Deposited 100000.0. New balance: 101000.0

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 3
Enter amount to withdraw: 50005
Withdrew 50005.0. New balance: 50995.0

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 4
Thank you for using the ATM. Goodbye!
◆ PS C:\Users\SPURTHI\OneDrive\Desktop\Ai Assistant Coding>
```

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.