

# **AJ ASSISTED CODING**

2303A52454

B-34

**LAB-6.5**

## **Task Description #1 (AI-Based Code Completion for Conditional Eligibility Check)**

*Task: Use an AI tool to generate eligibility logic.*

*Prompt:*

*“Generate Python code to check voting eligibility based on age and citizenship.”*

### *Expected Output:*

- AI-generated conditional logic.
  - Correct eligibility decisions.
  - Explanation of conditions.

C: > Users > MUTHOJU PRAVALIKA > Downloads > AI ASS6.5.py > ...

```
1 #Generate Python code to check voting eligibility based on age and citizenship."
2 age = int(input("Enter your age: "))
3 citizenship = input("Are you a citizen? (yes/no): ").lower()
4
5 if age >= 18 and citizenship == "yes":
6     print("You are eligible to vote.")
7 else:
8     print("You are not eligible to vote.")
9
10 #Generate Python code to count vowels and consonants in a string using a loop."
11 text = input("Enter a string: ")
12 vowels = "aeiouAEIOU"
13 vowel_count = 0
14 consonant_count = 0
15 for char in text:
16     if char.isalpha():
17         if char in vowels:
18             vowel_count += 1
+
... | []
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell Python
```

PS C:\Users\Users\Downloads> & "C:/Users/MUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/MUTHOJU PRAVALIKA/Downloads/AI ASS6.5.py"

Enter your age: 20  
Are you a citizen? (yes/no): yes  
You are eligible to vote.

## Task Description #2(AI-Based Code Completion for Loop-Based String Processing)

Task: Use an AI tool to process strings using loops.

### Prompt:

"Generate Python code to count vowels and consonants in a string using a loop."

Expected Output:

- AI-generated string processing logic.
- Correct counts.
- Output verification.

The screenshot shows a terminal window in a dark-themed code editor. The code in the editor is:

```
AI ASS6.5.py
C: > Users > MUTHOJU PRAVALIKA > Downloads > AI ASS6.5.py > ...
9
C:\Users\{MUTHOJU PRAVALIKA\Downloads\AI ASS6.5.py
n code to count vowels and consonants in a string using a loop."
a string: "
older.
I currently open
add a folder
ning a .NET
a new .NET
ject

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\{MUTHOJU PRAVALIKA> & "C:/Users/MUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/n.exe" "c:/Users/MUTHOJU PRAVALIKA/Downloads/AI ASS6.5.py"
● Enter your age: 20
Are you a citizen? (yes/no): yes
You are eligible to vote.
Enter a string: abasarnt
Vowels: 3, Consonants: 5
○ PS C:\Users\{MUTHOJU PRAVALIKA>
```

The terminal output shows the program running and printing the count of vowels and consonants for the string "abasarnt".

### Task Description #3 (AI-Assisted Code Completion Reflection Task)

Task: Use an AI tool to generate a complete program using classes, loops, and conditionals.

Prompt:

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

Expected Output:

- Complete AI-generated program.
- Review of AI suggestions quality.
- Short reflection on AI-assisted coding experience.

```

Go Run Terminal Help ← → ⌂ Search
... AI ASS6.5.py ●
C > Users > MUTHOJU PRAVALIKA > Downloads > AI ASS6.5.py > Book
PRAVALIKA...
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"Title: {book.title}, Author: {book.author}, Status: {status}")
19
20     def check_out_book(self, title):
21         for book in self.books:
22             if book.title == title and book.is_available:
23                 book.is_available = False
24                 print(f"You have checked out '{title}'")
25             return
26
27         print(f"Sorry, '{title}' is not available.")
28
29     def return_book(self, title):
30         for book in self.books:
31             if book.title == title and not book.is_available:
32                 book.is_available = True
33                 print(f"You have returned '{title}'")
34             return
35
36         print(f"'{title}' was not checked out.")
37
38 library = Library()
39 library.add_book(Book("1984", "George Orwell"))
40 library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
41 library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
42 while True:
43     print("\nLibrary Menu:")
44     print("1. Display Books")
45     print("2. Check Out Book")
46     print("3. Return Book")
47     print("4. Exit")
48     choice = input("Enter your choice (1-4): ")
49
50     if choice == '1':
51         library.display_books()
52     elif choice == '2':
53         title = input("Enter the title of the book to check out: ")
54

```

Ln 2, Col 1 Spaces: 4 UTF-8

```

42     if choice == '1':
43         library.display_books()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Library Menu:
1. Display Books
2. Check Out Book
3. Return Book
4. Exit
Enter your choice (1-4): 1
Title: 1984, Author: George Orwell, Status: Available
Title: To Kill a Mockingbird, Author: Harper Lee, Status: Available
Title: The Great Gatsby, Author: F. Scott Fitzgerald, Status: Available

```

## Task Description #4 (AI-Assisted Code Completion for Class-Based Attendance System)

Task: Use an AI tool to generate an attendance management class.

Prompt: “Generate a Python class to mark and display student attendance using loops.”

Expected Output:

- AI-generated attendance logic.

- Correct display of attendance.

- Test cases.

```

AI ASS6.5.py 3
C: > Users > MUTHOJU PRAVALIKA > Downloads > AI ASS6.5.py > ...
1
2
3
4
5
6  #Generate a Python class to mark and display student attendance using loops
7  class Attendance:
8      def __init__(self):
9          self.records = {}
10
11     def mark(self, name, status):
12         status = status.capitalize()
13         if status in ["Present", "Absent"]:
14             self.records[name] = status
15         else:
16             print("Invalid status")
17
18     def display(self):
19         for name, status in self.records.items():
20             print(f"{name}: {status}")
21
22 # Example usage
23 attendance = Attendance()
24 attendance.mark("Alice", "Present")
25 attendance.mark("Bob", "Absent")
26 attendance.display()
27
28
29
30
31

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MUTHOJU PRAVALIKA> & "C:/Users/MUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/MUTHOJU PRAVALIKA/Downloads/AI ASS6.5.py"
Alice: Present
Bob: Absent
PS C:\Users\MUTHOJU PRAVALIKA>

## Task Description #5 (AI-Based Code Completion for Conditional Menu Navigation)

### Menu Navigation

Task: Use an AI tool to complete a navigation menu.

Prompt: “Generate a Python program using loops and conditionals

to simulate an ATM menu.” Expected Output:

- AI-generated menu logic.
- Correct option handling.
- Output verification.

The screenshot shows a code editor interface with a terminal window below it. The terminal window displays the output of a Python program. The code in the editor is a Python script named 'AI ASS6.5.py' which simulates an ATM menu. The terminal output shows the program running and prompting the user for a choice, with the current balance displayed as \$1000.

```
View Go Run Terminal Help < > Search
...
AI ASS6.5.py X
C:\Users\UMUTHOJU PRAVALIKA> Downloads > AI ASS6.5.py > ...
3
4 #Generate a Python program using loops and conditionalsto simulate an ATM menu.
5 def atm_menu():
6     balance=1000 # Initial balance
7     while True:
8         print("\nWelcome to the ATM")
9         print("1. Check Balance")
10        print("2. Deposit Money")
11        print("3. Withdraw Money")
12        print("4. Exit")
13
14        choice = input("Please select an option (1-4): ")
15
16        if choice == '1':
17            print(f"Your current balance is: ${balance}")
18
19        elif choice == '2':
20            deposit = float(input("Enter amount to deposit: $"))
21            if deposit > 0:
22                balance += deposit
23                print(f"${deposit} deposited successfully.")
24            else:
25                print("Invalid amount. Please try again.")
26
27        elif choice == '3':
28            withdraw = float(input("Enter amount to withdraw: $"))
29            if 0 < withdraw <= balance:
30                balance -= withdraw
31                print(f"${withdraw} withdrawn successfully.")
32            else:
33                print("Insufficient funds or invalid amount. Please try again.")
34
35    elif choice == '4':
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\UMUTHOJU PRAVALIKA> & "C:/Users/UMUTHOJU PRAVALIKA/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/UMUTHOJU PRAVALIKA/Downloads/AI ASS6.5.py"
3. Withdraw Money
4. Exit
Please select an option (1-4): 1
Your current balance is: $1000

Welcome to the ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
```