

AI Assisted coding

Assignment-6.5

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

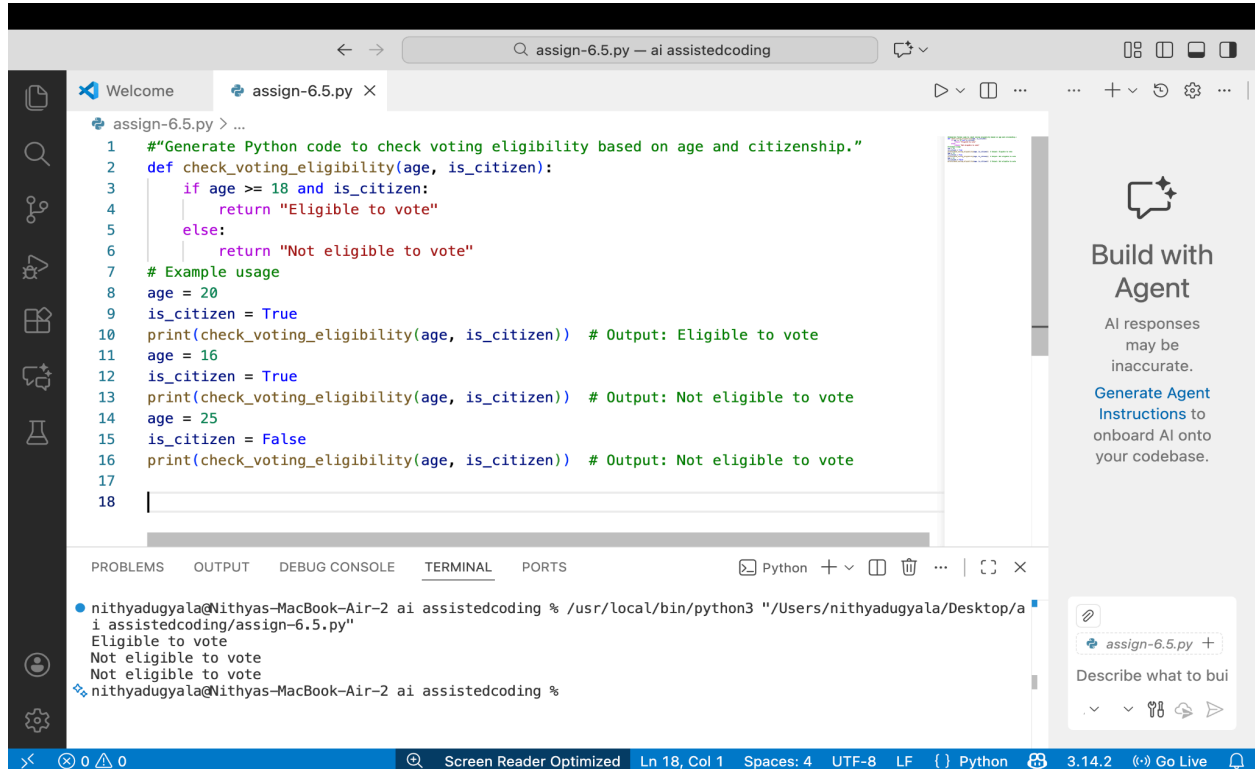
Task: Use an AI tool to generate eligibility logic.

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.



The screenshot displays an AI-assisted coding interface. The main editor window shows a Python script named `assign-6.5.py` with the following code:

```
1  """Generate Python code to check voting eligibility based on age and citizenship."""
2  def check_voting_eligibility(age, is_citizen):
3      if age >= 18 and is_citizen:
4          return "Eligible to vote"
5      else:
6          return "Not eligible to vote"
7  # Example usage
8  age = 20
9  is_citizen = True
10 print(check_voting_eligibility(age, is_citizen)) # Output: Eligible to vote
11 age = 16
12 is_citizen = True
13 print(check_voting_eligibility(age, is_citizen)) # Output: Not eligible to vote
14 age = 25
15 is_citizen = False
16 print(check_voting_eligibility(age, is_citizen)) # Output: Not eligible to vote
17
18
```

The terminal window at the bottom shows the execution of the script using `python3`, resulting in the following output:

```
nithyadugyala@Nithyas-MacBook-Air-2 ai assistedcoding % /usr/local/bin/python3 "/Users/nithyadugyala/Desktop/a
i assistedcoding/assign-6.5.py"
Eligible to vote
Not eligible to vote
Not eligible to vote
nithyadugyala@Nithyas-MacBook-Air-2 ai assistedcoding %
```

On the right side of the interface, there is a sidebar with the text "Build with Agent" and a warning: "AI responses may be inaccurate." Below this, there is a link to "Generate Agent Instructions to onboard AI onto your codebase." At the bottom right, there is a small chat window with the text "Describe what to bui" and a "Go Live" button.

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

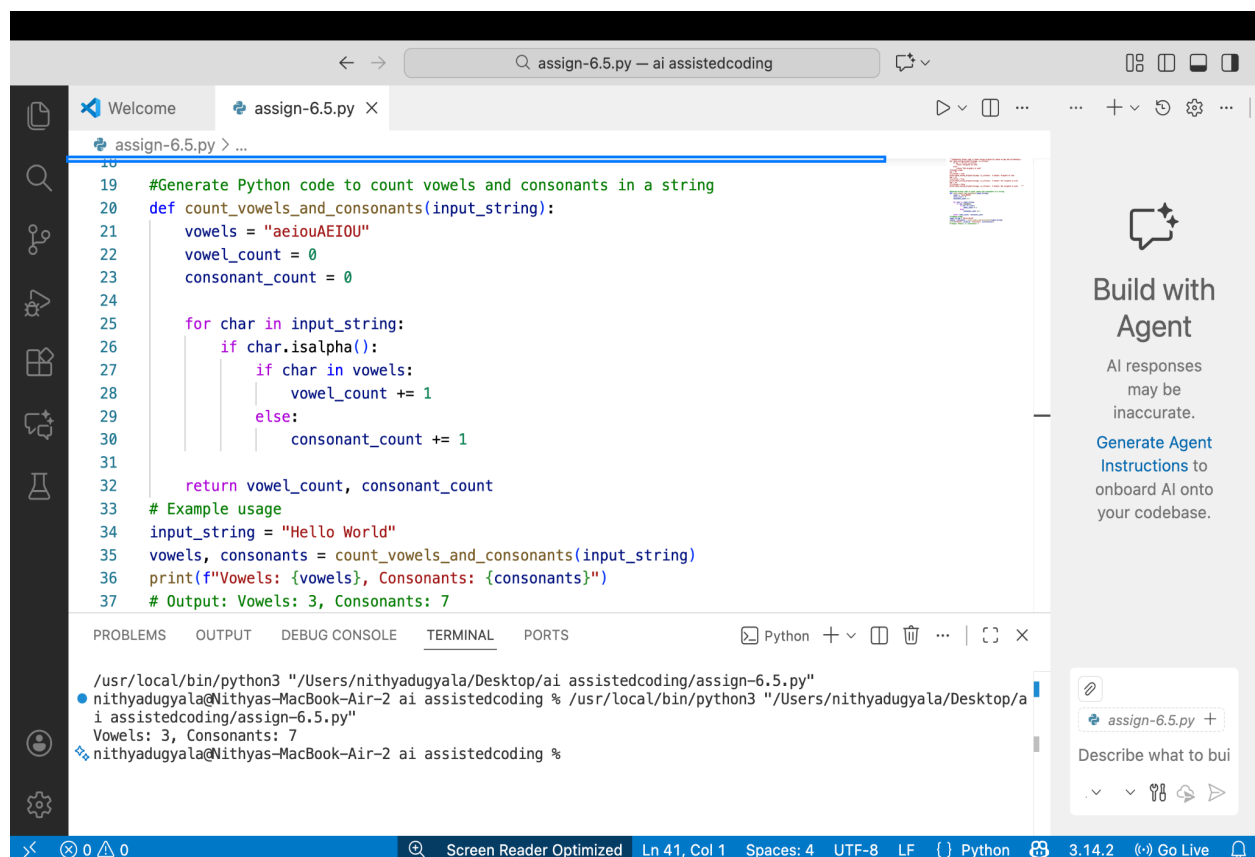
Task: Use an AI tool to process strings using loops.

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 displays an AI-assisted coding interface. The main editor window shows a Python script named `assign-6.5.py` with the following code:

```
19 #Generate Python code to count vowels and consonants in a string
20 def count_vowels_and_consonants(input_string):
21     vowels = "aeiouAEIOU"
22     vowel_count = 0
23     consonant_count = 0
24
25     for char in input_string:
26         if char.isalpha():
27             if char in vowels:
28                 vowel_count += 1
29             else:
30                 consonant_count += 1
31
32     return vowel_count, consonant_count
33
34 # Example usage
35 input_string = "Hello World"
36 vowels, consonants = count_vowels_and_consonants(input_string)
37 print(f"Vowels: {vowels}, Consonants: {consonants}")
38 # Output: Vowels: 3, Consonants: 7
```

Below the code editor, the terminal window shows the execution of the script:

```
/usr/local/bin/python3 "/Users/nithyadugyala/Desktop/ai assistedcoding/assign-6.5.py"
nithyadugyala@Nithyas-MacBook-Air-2 ai assistedcoding % /usr/local/bin/python3 "/Users/nithyadugyala/Desktop/a
i assistedcoding/assign-6.5.py"
Vowels: 3, Consonants: 7
nithyadugyala@Nithyas-MacBook-Air-2 ai assistedcoding %
```

On the right side of the interface, there is a sidebar with the text "Build with Agent" and a note: "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase." Below this, there is a search bar with the text "Describe what to bui" and a button labeled "assign-6.5.py +".

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

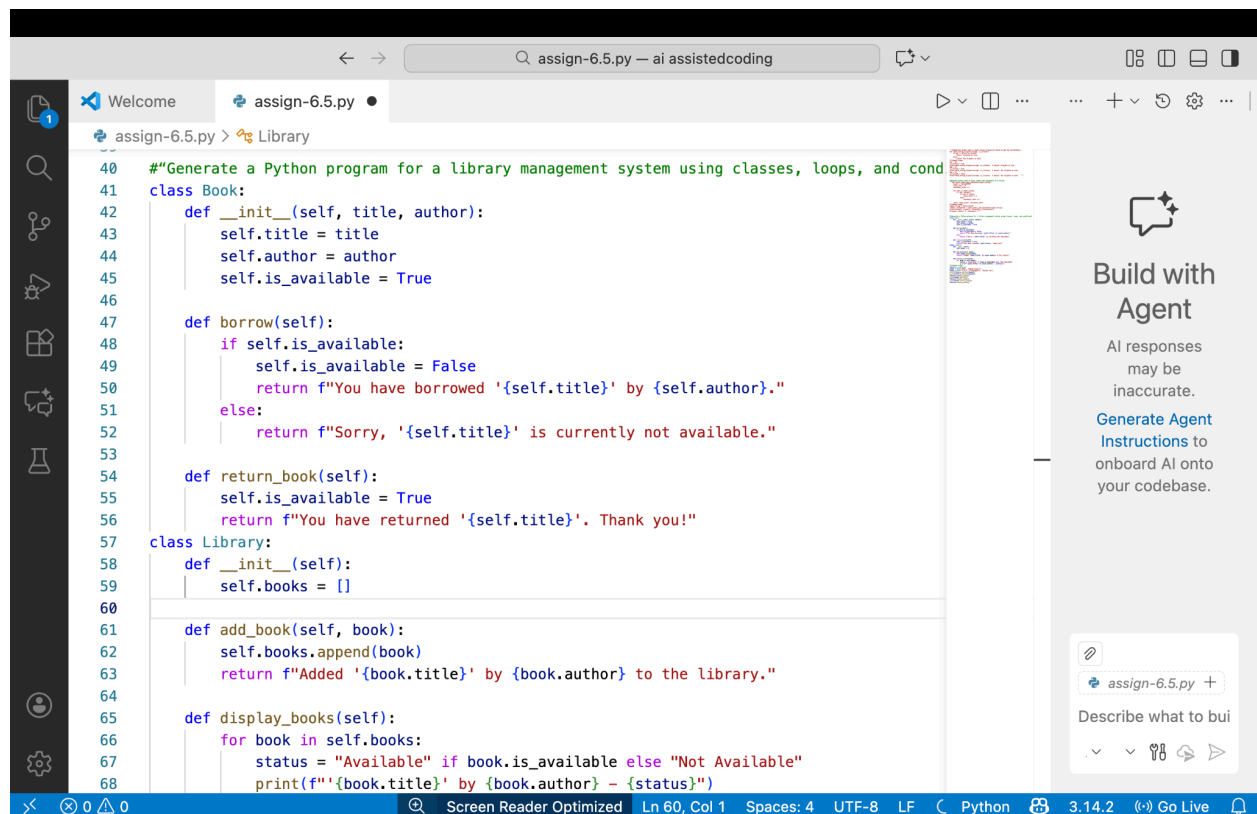
Task: Use an AI tool to generate a complete program using classes, loops, and conditionals.

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.



```
40  #Generate a Python program for a library management system using classes, loops, and cond
41  class Book:
42      def __init__(self, title, author):
43          self.title = title
44          self.author = author
45          self.is_available = True
46
47      def borrow(self):
48          if self.is_available:
49              self.is_available = False
50              return f"You have borrowed '{self.title}' by {self.author}."
51          else:
52              return f"Sorry, '{self.title}' is currently not available."
53
54      def return_book(self):
55          self.is_available = True
56          return f"You have returned '{self.title}'. Thank you!"
57  class Library:
58      def __init__(self):
59          self.books = []
60
61      def add_book(self, book):
62          self.books.append(book)
63          return f"Added '{book.title}' by {book.author} to the library."
64
65      def display_books(self):
66          for book in self.books:
67              status = "Available" if book.is_available else "Not Available"
68              print(f"'{book.title}' by {book.author} - {status}")
```

Build with Agent

AI responses may be inaccurate.

[Generate Agent Instructions](#) to onboard AI onto your codebase.

assign-6.5.py +

Describe what to build

The screenshot shows a code editor with a file named `assign-6.5.py`. The code defines a `Library` class with methods `add_book`, `display_books`, `borrow`, and `return_book`. It also includes example usage code. The terminal output shows the execution of the script, displaying the results of the library operations.

```
57 class Library:
58     def __init__(self):
59         self.books = []
60
61     def add_book(self, book):
62         self.books.append(book)
63         return f"Added '{book.title}' by {book.author} to the library."
64
65     def display_books(self):
66         for book in self.books:
67             status = "Available" if book.is_available else "Not Available"
68             print(f"'{book.title}' by {book.author} - {status}")
69
70 # Example usage
71 library = Library()
72 book1 = Book("1984", "George Orwell")
73 book2 = Book("To Kill a Mockingbird", "Harper Lee")
74 print(library.add_book(book1))
75 print(library.add_book(book2))
76 library.display_books()
77 book1.borrow()
78 library.display_books()
79 book1.return_book()
80 library.display_books()
```

```
Python 3.14.2
nithyadugyala@Nithyas-MacBook-Air: ~ % python3 assign-6.5.py
Added '1984' by George Orwell to the library.
Added 'To Kill a Mockingbird' by Harper Lee to the library.
'1984' by George Orwell - Available
'To Kill a Mockingbird' by Harper Lee - Available
You have borrowed '1984' by George Orwell.
'1984' by George Orwell - Not Available
'To Kill a Mockingbird' by Harper Lee - Available
You have returned '1984'. Thank you!
'1984' by George Orwell - Available
'To Kill a Mockingbird' by Harper Lee - Available
```

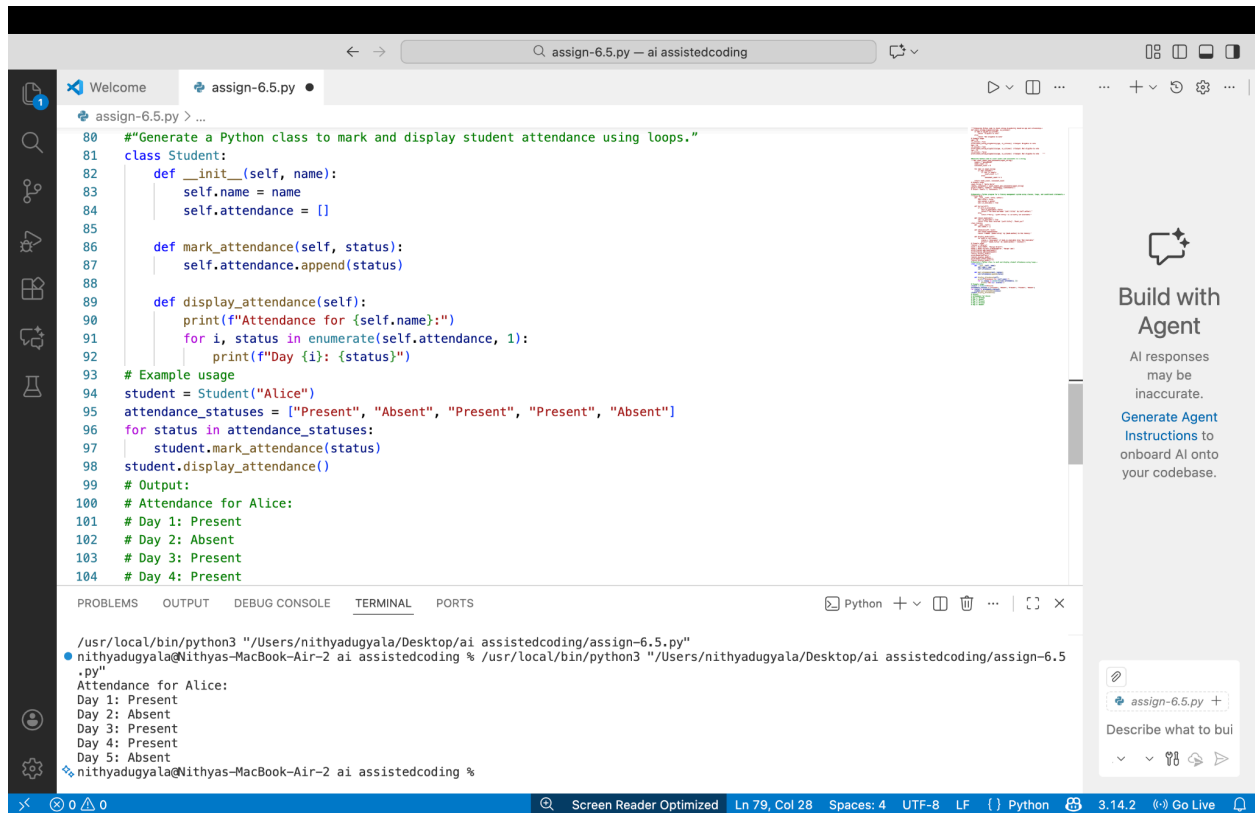
Task Description #4 (AI-Assisted Code Completion for Class-Based Attendance System)

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.



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

