

# Experiment 6: AI-Based Code Completion: Working with suggestions for classes, loops, conditionals

Name: Syed Murtaza

Hall ticket no: 2303A51259

## Task 1: AI-Based Code Completion for Conditional Eligibility Check

**Prompt:**

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

**Code & Output:**

The screenshot shows a code editor interface with two panes. The left pane displays a Python file named 'Assignment\_6.5.py' containing the following code:

```
Assignment_6.5.py
1  "Task 1: AI-Based Code Completion for Conditional Eligibility Check"
2  # Generate Python code to check voting eligibility based on age and
3  # citizenship.
4  age = int(input("Enter your age: "))
5  citizenship = input("Are you a citizen? (yes/no): ").lower()
6
7  if age >= 18 and citizenship == "yes":
8      print("You are eligible to vote.")
9  else:
10     print("You are not eligible to vote.")
```

The right pane shows the terminal output of running the script:

```
Are you a citizen? (yes/no): yes
Are you a citizen? (yes/no): yes
You are not eligible to vote.
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5.py"
Enter your age: 18
Are you a citizen? (yes/no): no
You are not eligible to vote.
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5.py"
Enter your age: 18
Are you a citizen? (yes/no): yes
You are eligible to vote.
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

**Explanation:**

The AI-generated code uses conditional statements to check voting eligibility. It verifies whether the age is 18 or above and whether the user is a citizen. Both conditions must be true for eligibility. This demonstrates correct use of conditional logic generated through AI-based code completion.

## Task 2: AI-Based Code Completion for Loop-Based String Processing

## Prompt:

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

## Code & Output:

```
Assignment_6.5.py
```

```
Assignment_6.5 > Assignment_6.5.py > ...
```

```
11 "Task 2: AI-Based Code Completion for Loop-Based String Processing"
12 # Generate Python code to count vowels and consonants in a string using a loop
13 input_string = input("Enter a string: ")
14 vowel_count = 0
15 consonant_count = 0
16 vowels = "aeiouAEIOU"
17 for char in input_string:
18     if char.isalpha():
19         if char in vowels:
20             vowel_count += 1
21         else:
22             consonant_count += 1
23 print(f"Vowels: {vowel_count}, Consonants: {consonant_count}")
24
```

```
All Assisted Coding
```

```
Code
```

```
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5Assignment_6.5.py"
Enter a string: Hello World
Vowels: 3, Consonants: 7
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

## Explanation:

The AI-generated code processes the input string using a loop. Each character is checked to determine whether it is a vowel or a consonant. Alphabetic characters are counted correctly, while non-letter characters are ignored. The output verifies that the logic works as expected.

## Task 3: AI-Assisted Code Completion Reflection Task

## Prompt:

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

## Code & Output:

```
Assignment_6.5.py
Assignment_6.5 > Assignment_6.5.py > ...
24
25     # Task 3: AI-Assisted Code Completion Reflection Task"
26     # Generate a Python program for a library management system using classes,
27     # loops, and conditional statements.
28
29     class Book:
30         def __init__(self, title, author):
31             self.title = title
32             self.author = author
33             self.is_borrowed = False
34
35         def borrow(self):
36             if not self.is_borrowed:
37                 self.is_borrowed = True
38             return True
39         return False
40
41         def return_book(self):
42             if self.is_borrowed:
43                 self.is_borrowed = False
44             return True
45         return False
46
47     class Library:
48         def __init__(self):
49             self.books = []
50
51         def add_book(self, book):
52             self.books.append(book)
53
54         def display_books(self):
55             for idx, book in enumerate(self.books):
56                 status = "borrowed" if book.is_borrowed else "Available"
57                 print(f"{idx + 1}. {book.title} by {book.author} - {status}")
58
59         def borrow_book(self, index):
60             if 0 <= index < len(self.books):
61                 if self.books[index].borrow():
62                     print(f"You have borrowed '{self.books[index].title}'")
63
64         def return_book(self, index):
65             if 0 <= index < len(self.books):
66                 if self.books[index].return_book():
67                     print(f"You have returned '{self.books[index].title}'")
68                 else:
69                     print(f"'{self.books[index].title}' was not borrowed.")
70             else:
71                 print("invalid book index.")
72
73     def main():
74         library = Library()
75         library.add_book(Book("1984", "George Orwell"))
76         library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
77         library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
78
79         while True:
80             print("\nLibrary Menu:")
81             print("1. Display Books")
82             print("2. Borrow Book")
83             print("3. Return Book")
84             print("4. Exit")
85             choice = input("Enter your choice: ")
86
87             if choice == '1':
88                 library.display_books()
89             elif choice == '2':
90                 index = int(input("Enter the book index to borrow: ")) - 1
91                 library.borrow_book(index)
92             elif choice == '3':
93                 index = int(input("Enter the book index to return: ")) - 1
94                 library.return_book(index)
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
```

PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment\_6.5\Assignment\_6.5.py"

1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 2  
Enter the book index to borrow: 3  
You have borrowed 'The Great Gatsby'.

Library Menu:  
1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 1  
1. 1984 by George Orwell - Available  
2. To Kill a Mockingbird by Harper Lee - Available  
3. The Great Gatsby by F. Scott Fitzgerald - Borrowed

Library Menu:  
1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 3  
Enter the book index to return: 3  
You have returned 'The Great Gatsby'.

Library Menu:  
1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 1  
1. 1984 by George Orwell - Available  
2. To Kill a Mockingbird by Harper Lee - Available  
1. 1984 by George Orwell - Available  
2. To Kill a Mockingbird by Harper Lee - Available  
3. The Great Gatsby by F. Scott Fitzgerald - Available

Library Menu:  
1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 1  
1. 1984 by George Orwell - Available  
2. To Kill a Mockingbird by Harper Lee - Available  
3. The Great Gatsby by F. Scott Fitzgerald - Available

```
Assignment_6.5.py
Assignment_6.5 > Assignment_6.5.py > main
44
45     class Library:
46
47         def borrow_book(self, index):
48
49             else:
50                 print(f"'{self.books[index].title}' is already borrowed.")
51             else:
52                 print("invalid book index.")
53
54         def return_book(self, index):
55
56             if 0 <= index < len(self.books):
57                 if self.books[index].return_book():
58                     print(f"You have returned '{self.books[index].title}'")
59                 else:
60                     print(f"'{self.books[index].title}' was not borrowed.")
61             else:
62                 print("invalid book index.")
63
64     def main():
65         library = Library()
66         library.add_book(Book("1984", "George Orwell"))
67         library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
68         library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
69
70
71         while True:
72             print("\nLibrary Menu:")
73             print("1. Display Books")
74             print("2. Borrow Book")
75             print("3. Return Book")
76             print("4. Exit")
77             choice = input("Enter your choice: ")
78
79             if choice == '1':
80                 library.display_books()
81             elif choice == '2':
82                 index = int(input("Enter the book index to borrow: ")) - 1
83                 library.borrow_book(index)
84             elif choice == '3':
85                 index = int(input("Enter the book index to return: ")) - 1
86                 library.return_book(index)
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
109
```

PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment\_6.5\Assignment\_6.5.py"

Enter the book index to borrow: 3  
You have borrowed 'The Great Gatsby'.

Library Menu:  
1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 1  
1. 1984 by George Orwell - Available  
2. To Kill a Mockingbird by Harper Lee - Available  
3. The Great Gatsby by F. Scott Fitzgerald - Available

Library Menu:  
1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 3  
Enter the book index to return: 3  
You have returned 'The Great Gatsby'.

Library Menu:  
1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 1  
1. 1984 by George Orwell - Available  
2. To Kill a Mockingbird by Harper Lee - Available  
1. 1984 by George Orwell - Available  
2. To Kill a Mockingbird by Harper Lee - Available  
3. The Great Gatsby by F. Scott Fitzgerald - Available

Library Menu:  
1. Display Books  
2. Borrow Book  
3. Return Book  
4. Exit  
Enter your choice: 1  
1. 1984 by George Orwell - Available  
2. To Kill a Mockingbird by Harper Lee - Available  
3. The Great Gatsby by F. Scott Fitzgerald - Available

The screenshot shows a code editor with a dark theme. On the left is the code file `Assignment_6.5.py`, which contains Python code for a library system. The code defines a `Library` class with methods for adding books, displaying books, borrowing books, returning books, and exiting. It uses a menu loop to interact with the user. On the right, the terminal window shows the execution of the program. The user enters choices 1 through 4, and the program displays the current state of the library, including book titles, authors, and availability.

```

File Edit Selection View Go Run Terminal Help ⏪ ⏩ ...
Assignment_6.5.py X AI Assisted Coding
Assignment_6.5 > Assignment_6.5.py > main
73 def main():
74     library = Library()
75     library.add_book(Book("1984", "George Orwell"))
76     library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
77     library.add_book(Book("The Great Gatsby", "F. Scott Fitzgerald"))
78
79     while True:
80         print("\nLibrary Menu:")
81         print("1. Display Books")
82         print("2. Borrow Book")
83         print("3. Return Book")
84         print("4. Exit")
85         choice = input("Enter your choice: ")
86
87         if choice == '1':
88             library.display_books()
89         elif choice == '2':
90             index = int(input("Enter the book index to borrow: ")) - 1
91             library.borrow_book(index)
92         elif choice == '3':
93             index = int(input("Enter the book index to return: ")) - 1
94             library.return_book(index)
95         elif choice == '4':
96             print("Exiting the library system.")
97             break
98         else:
99             print("Invalid choice. Please try again.")
100 if __name__ == "__main__":
101     main()
102
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5\Assignment_6.5.py"
1. Display Books
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 3
Enter the book index to return: 3
You have returned 'The Great Gatsby'.

Library Menu:
1. Display Books
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 1
1. 1984 by George Orwell - Available
2. To Kill a Mockingbird by Harper Lee - Available
1. 1984 by George Orwell - Available
2. To Kill a Mockingbird by Harper Lee - Available
3. The Great Gatsby by F. Scott Fitzgerald - Available
PS E:\3rd Year\2nd Sem\AI Assisted Coding>

```

### Explanation:

The AI-generated program uses a class to represent a library and includes loops and conditional statements for menu-driven interaction. The loop allows continuous user input, and conditionals control program flow. The program correctly demonstrates AI-assisted use of object-oriented programming concepts.

### Reflection on AI-Assisted Coding:

The AI tool generated a complete and functional program quickly. While the logic is correct, the code can be further improved with input validation and advanced features. This task shows that AI is useful for speeding up development but still requires human review and optimization.

## Task 4: AI-Assisted Code Completion for Class-Based Attendance System

### Prompt:

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

### Code & Output:

The screenshot shows a code editor interface with a dark theme. On the left is the code editor pane containing a file named 'Assignment\_6.5.py'. The code defines a class 'AttendanceSystem' with methods for marking attendance and displaying records. It includes a main loop for user interaction. On the right is the terminal pane where the code is run. The terminal shows the output of the program, which asks for student names and marks them as 'Present', then displays the attendance record for both 'Vineeth' and 'Koushik'.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ AI Assisted Coding
Assignment_6.5.py > Assignment_6.5.py > ...
102
103 "Task 4: AI-Assisted Code Completion for Class-Based Attendance System"
104 # Generate a Python class to mark and display student attendance using loops.
105
106 class AttendanceSystem:
107     def __init__(self):
108         self.attendance = {}
109
110     def mark_attendance(self, student_name):
111         self.attendance[student_name] = "Present"
112
113     def display_attendance(self):
114         print("Attendance Record:")
115         for student, status in self.attendance.items():
116             print(f"{student}: {status}")
117
118     def main():
119         attendance_system = AttendanceSystem()
120         while True:
121             name = input("Enter student name to mark attendance (or 'exit' to finish): ")
122             if name.lower() == "exit":
123                 break
124             attendance_system.mark_attendance(name)
125         attendance_system.display_attendance()
126
127 if __name__ == "__main__":
128     main()

PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5\Assignment_6.5.py"
Enter student name to mark attendance (or 'exit' to finish): Vineeth
Enter student name to mark attendance (or 'exit' to finish): Koushik
Enter student name to mark attendance (or 'exit' to finish): exit
Attendance Record:
Vineeth: Present
Koushik: Present
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

### Explanation:

The AI-generated attendance system uses a class to store attendance data. A loop is used to take multiple student entries, and another loop displays the attendance records. The code works correctly and demonstrates class-based AI code completion.

## Task 5: AI-Based Code Completion for Conditional Menu Navigation

### Prompt:

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

### Code & Output:

```
Assignment_6.5.py
Assignment_6.5 > Assignment_6.5.py > ...
127  "Task 5: AI-Based Code Completion for Conditional Menu Navigation"
128  # Generate a Python program using loops and conditionals to simulate an ATM
129  menu.
130  balance = 1000.0
131  while True:
132      print("\nATM Menu:")
133      print("1. Check Balance")
134      print("2. Deposit Money")
135      print("3. Withdraw Money")
136      print("4. Exit")
137      choice = input("Enter your choice: ")
138
139      if choice == '1':
140          print(f"Your current balance is: ${balance:.2f}")
141      elif choice == '2':
142          amount = float(input("Enter amount to deposit: "))
143          if amount > 0:
144              balance += amount
145              print(f"${amount:.2f} deposited successfully.")
146              print(f"Your current balance is: ${balance:.2f}")
147          else:
148              print("Invalid amount. Please enter a positive value.")
149      elif choice == '3':
150          amount = float(input("Enter amount to withdraw: "))
151          if 0 < amount <= balance:
152              balance -= amount
153              print(f"${amount:.2f} withdrawn successfully.")
154              print(f"Your current balance is: ${balance:.2f}")
155          else:
156              print("Invalid amount or insufficient balance.")
157      elif choice == '4':
158          print("Exiting the ATM. Thank you!")
159          break
160      else:
161          print("Invalid choice. Please try again.")

PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_6.5\Assignment_6.5.py"
ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 1
Your current balance is: $1000.00

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 2
Enter amount to deposit: 14500
$14500.00 deposited successfully.
Your current balance is: $15500.00

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 3
Enter amount to withdraw: 1720
$1720.00 withdrawn successfully.
Your current balance is: $13780.00

ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 4
Exiting the ATM. Thank you!
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

## Explanation:

The AI-generated ATM program uses a loop to display the menu repeatedly and conditional statements to handle user choices. The logic correctly updates the balance and prevents invalid withdrawals. This task demonstrates effective AI-based code completion for menu-driven programs.

## Final Conclusion:

This experiment shows how AI-based code completion tools can generate useful Python code involving classes, loops, and conditionals. While AI speeds up development, developers must still review logic, handle edge cases, and ensure ethical and responsible use of AI-generated code.