

Varshini Girugula

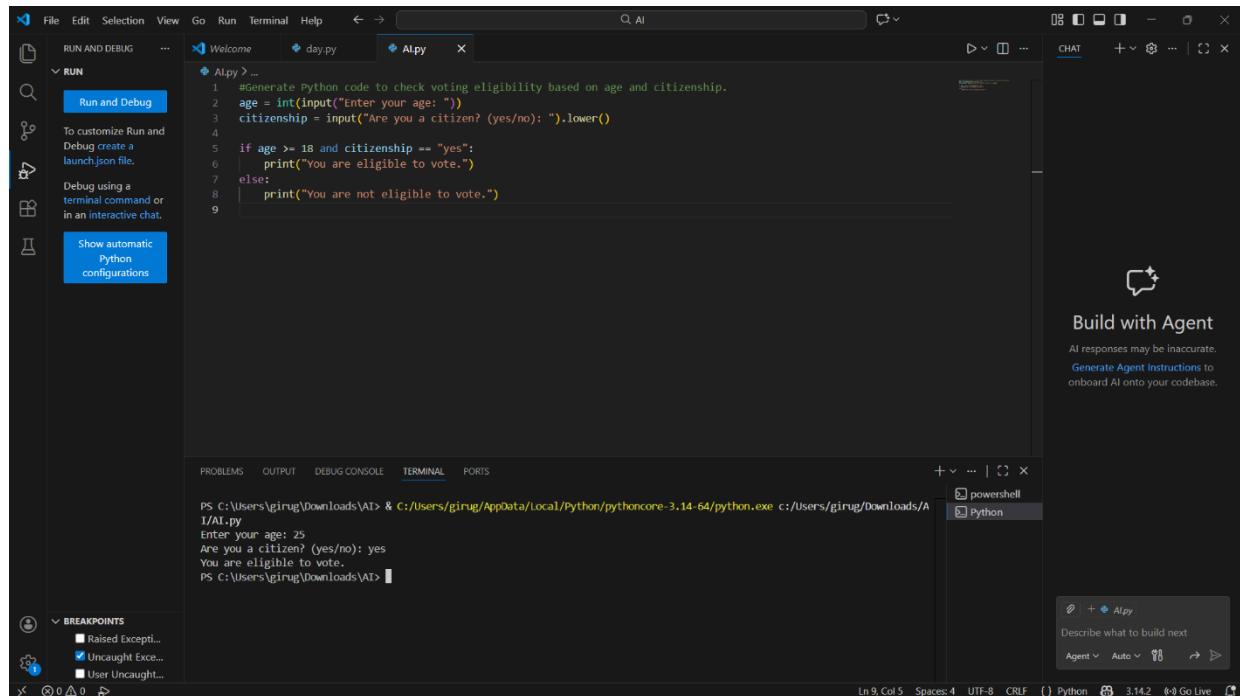
2403a51l14

Batch-51

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

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

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



```
#Generate Python code to check voting eligibility based on age and citizenship.
age = int(input("Enter your age: "))
citizenship = input("Are you a citizen? (yes/no): ").lower()

if age >= 18 and citizenship == "yes":
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

### Justification:

It checks voting eligibility using conditional statements. It verifies whether the user's age is 18 or above and if the person is a citizen. Based on these conditions, it displays eligibility status.

age  $\geq$  18 → Checks minimum voting age.

citizen == "yes" → Ensures citizenship.

and → Both conditions must be true.

if-else → Makes the eligibility decision.

## Task Description – 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.

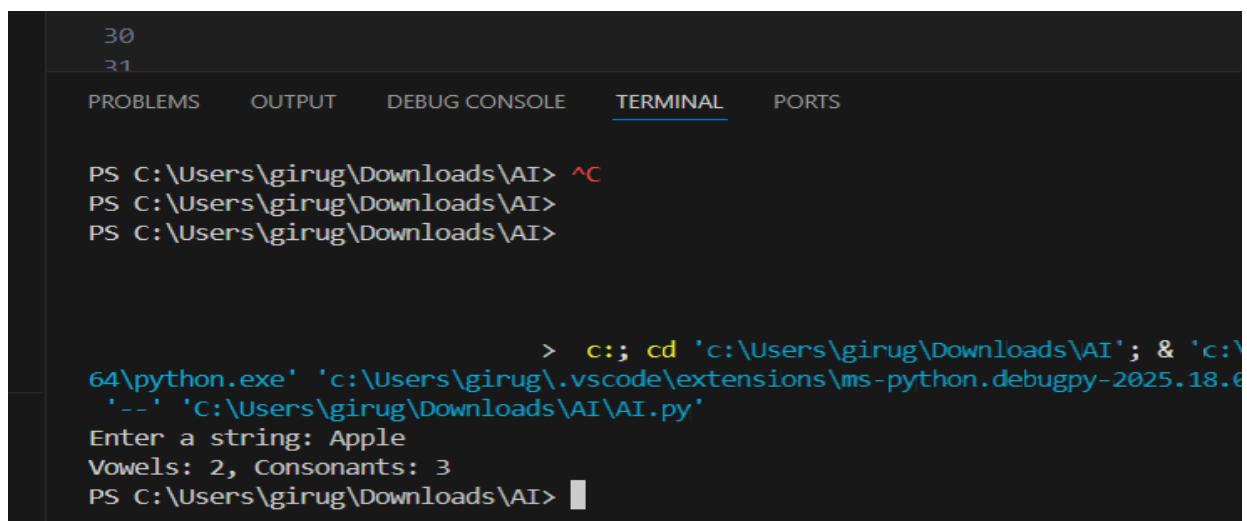
### AI-Generated Code:

```
#task2
#“Generate Python code to count vowels and consonants in a string using a loop.”
def count_vowels_consonants(s):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in s:
        if char.isalpha(): # Check if the character is a letter
            if char in vowels:
                vowel_count += 1 # Increment vowel count
            else:
                consonant_count += 1 # Increment consonant count

    return vowel_count, consonant_count
input_string = input("Enter a string: ")
vowels, consonants = count_vowels_consonants(input_string)
print(f"Vowels: {vowels}, Consonants: {consonants}")
```

### OUTPUT:



```
PS C:\Users\girug\Downloads\AI> ^C
PS C:\Users\girug\Downloads\AI>
PS C:\Users\girug\Downloads\AI>

                                     > c;; cd 'c:\users\girug\downloads\AI'; & 'c:\v
64\python.exe' 'c:\users\girug\.vscode\extensions\ms-python.debugpy-2025.18.6
' '--' 'c:\users\girug\downloads\AI\AI.py'
Enter a string: Apple
Vowels: 2, Consonants: 3
PS C:\Users\girug\Downloads\AI>
```

### Justification:

This task uses AI-generated Python code to count vowels and consonants in a given string using a loop. The program reads a string input from the user and

converts it to lowercase to avoid case sensitivity. A for loop iterates through each character in the string. If it is a vowel, the vowel count is incremented; otherwise, the consonant count is increased. This task demonstrates the use of loops and nested conditionals for string processing and character analysis.

## **Task Description – 3: AI-Assisted Code Completion Reflection Task**

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

```
File Edit Selection View Go Run Terminal Help < > Q AI RUN AND DEBUG ... Welcome day.py Al.py X RUN To customize Run and Debug create a launch.json file. Debug using a terminal command or in an interactive chat. Show automatic Python configurations CHAT + + Build with Agent AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.
```

```
29 #task3
30 #Generate a Python program for a library management system using classes, loops, and conditional statements.
31 class Book:
32     def __init__(self, title, author):
33         self.title = title
34         self.author = author
35         self.is_borrowed = False
36 
37 class Library:
38     def __init__(self):
39         self.books = []
40 
41     def add_book(self, book):
42         self.books.append(book)
43 
44     def display_books(self):
45         for book in self.books:
46             status = "Borrowed" if book.is_borrowed else "Available"
47             print(f"{book.title} by {book.author} - {status}")
48 
49     def borrow_book(self, title):
50         for book in self.books:
51             if book.title == title and not book.is_borrowed:
52                 book.is_borrowed = True
53                 print("You have borrowed '{book.title}'")
54             return
55             print("Sorry, '{title}' is not available.")
56 
57     def return_book(self, title):
58         for book in self.books:
59             if book.title == title and book.is_borrowed:
60                 book.is_borrowed = False
61                 print("You have returned '{book.title}'")
62             return
63             print("'{title}' was not borrowed.")
64 
# Example usage
65 library = Library()
66 library.add_book(Book("1984", "George Orwell"))
67 library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
68 library.display_books()
```

```
File Edit Selection View Go Run Terminal Help < > Q AI RUN AND DEBUG ... Welcome day.py Al.py X RUN To customize Run and Debug create a launch.json file. Debug using a terminal command or in an interactive chat. Show automatic Python configurations CHAT + + Build with Agent AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.
```

```
36 class Library:
37     def return_book(self, title):
38         print("'{title}' was not borrowed.")
39 
# Example usage
40 library = Library()
41 library.add_book(Book("1984", "George Orwell"))
42 library.add_book(Book("To Kill a Mockingbird", "Harper Lee"))
43 library.display_books()
44 library.borrow_book("1984")
45 library.display_books()
46 library.return_book("1984")
47 library.display_books()
48 
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
49 
50 
51 
52 
53 
54 
55 
56 
57 
58 
59 
60 
61 
62 
63 
64 
65 
66 
67 
68 
69 
70 
71 
72 
73 
74 
75 
76 
77 
78 
79 
80 
81 
82 
83 
84 
85 
86 
87 
88 
89 
90 
91 
92 
93 
94 
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 
260 
261 
262 
263 
264 
265 
266 
267 
268 
269 
270 
271 
272 
273 
274 
275 
276 
277 
278 
279 
279 
280 
281 
282 
283 
284 
285 
286 
287 
288 
289 
289 
290 
291 
292 
293 
294 
295 
296 
297 
298 
299 
299 
300 
301 
302 
303 
304 
305 
306 
307 
308 
309 
309 
310 
311 
312 
313 
314 
315 
316 
317 
318 
319 
319 
320 
321 
322 
323 
324 
325 
326 
327 
328 
329 
329 
330 
331 
332 
333 
334 
335 
336 
337 
338 
339 
339 
340 
341 
342 
343 
344 
345 
346 
347 
348 
349 
349 
350 
351 
352 
353 
354 
355 
356 
357 
358 
359 
359 
360 
361 
362 
363 
364 
365 
366 
367 
368 
369 
369 
370 
371 
372 
373 
374 
375 
376 
377 
378 
379 
379 
380 
381 
382 
383 
384 
385 
386 
387 
388 
389 
389 
390 
391 
392 
393 
394 
395 
396 
397 
398 
399 
399 
400 
401 
402 
403 
404 
405 
406 
407 
408 
409 
409 
410 
411 
412 
413 
414 
415 
416 
417 
418 
419 
419 
420 
421 
422 
423 
424 
425 
426 
427 
428 
429 
429 
430 
431 
432 
433 
434 
435 
436 
437 
438 
439 
439 
440 
441 
442 
443 
444 
445 
446 
447 
448 
449 
449 
450 
451 
452 
453 
454 
455 
456 
457 
458 
459 
459 
460 
461 
462 
463 
464 
465 
466 
467 
468 
469 
469 
470 
471 
472 
473 
474 
475 
476 
477 
478 
479 
479 
480 
481 
482 
483 
484 
485 
486 
487 
488 
489 
489 
490 
491 
492 
493 
494 
495 
496 
497 
498 
499 
499 
500 
501 
502 
503 
504 
505 
506 
507 
508 
509 
509 
510 
511 
512 
513 
514 
515 
516 
517 
518 
519 
519 
520 
521 
522 
523 
524 
525 
526 
527 
528 
529 
529 
530 
531 
532 
533 
534 
535 
536 
537 
538 
539 
539 
540 
541 
542 
543 
544 
545 
546 
547 
548 
549 
549 
550 
551 
552 
553 
554 
555 
556 
557 
558 
559 
559 
560 
561 
562 
563 
564 
565 
566 
567 
568 
569 
569 
570 
571 
572 
573 
574 
575 
576 
577 
578 
579 
579 
580 
581 
582 
583 
584 
585 
586 
587 
588 
589 
589 
590 
591 
592 
593 
594 
595 
596 
597 
598 
599 
599 
600 
601 
602 
603 
604 
605 
606 
607 
608 
609 
609 
610 
611 
612 
613 
614 
615 
616 
617 
618 
619 
619 
620 
621 
622 
623 
624 
625 
626 
627 
628 
629 
629 
630 
631 
632 
633 
634 
635 
636 
637 
638 
639 
639 
640 
641 
642 
643 
644 
645 
646 
647 
648 
649 
649 
650 
651 
652 
653 
654 
655 
656 
657 
658 
659 
659 
660 
661 
662 
663 
664 
665 
666 
667 
668 
669 
669 
670 
671 
672 
673 
674 
675 
676 
677 
678 
679 
679 
680 
681 
682 
683 
684 
685 
686 
687 
688 
689 
689 
690 
691 
692 
693 
694 
695 
696 
697 
698 
699 
699 
700 
701 
702 
703 
704 
705 
706 
707 
708 
709 
709 
710 
711 
712 
713 
714 
715 
716 
717 
718 
719 
719 
720 
721 
722 
723 
724 
725 
726 
727 
728 
729 
729 
730 
731 
732 
733 
734 
735 
736 
737 
738 
739 
739 
740 
741 
742 
743 
744 
745 
746 
747 
748 
749 
749 
750 
751 
752 
753 
754 
755 
756 
757 
758 
759 
759 
760 
761 
762 
763 
764 
765 
766 
767 
768 
769 
769 
770 
771 
772 
773 
774 
775 
776 
777 
778 
779 
779 
780 
781 
782 
783 
784 
785 
786 
787 
788 
789 
789 
790 
791 
792 
793 
794 
795 
796 
797 
798 
799 
799 
800 
801 
802 
803 
804 
805 
806 
807 
808 
809 
809 
810 
811 
812 
813 
814 
815 
816 
817 
818 
819 
819 
820 
821 
822 
823 
824 
825 
826 
827 
828 
829 
829 
830 
831 
832 
833 
834 
835 
836 
837 
838 
839 
839 
840 
841 
842 
843 
844 
845 
846 
847 
848 
849 
849 
850 
851 
852 
853 
854 
855 
856 
857 
858 
859 
859 
860 
861 
862 
863 
864 
865 
866 
867 
868 
869 
869 
870 
871 
872 
873 
874 
875 
876 
877 
878 
879 
879 
880 
881 
882 
883 
884 
885 
886 
887 
888 
889 
889 
890 
891 
892 
893 
894 
895 
896 
897 
898 
899 
899 
900 
901 
902 
903 
904 
905 
906 
907 
908 
909 
909 
910 
911 
912 
913 
914 
915 
916 
917 
918 
919 
919 
920 
921 
922 
923 
924 
925 
926 
927 
928 
929 
929 
930 
931 
932 
933 
934 
935 
936 
937 
938 
939 
939 
940 
941 
942 
943 
944 
945 
946 
947 
948 
949 
949 
950 
951 
952 
953 
954 
955 
956 
957 
958 
959 
959 
960 
961 
962 
963 
964 
965 
966 
967 
968 
969 
969 
970 
971 
972 
973 
974 
975 
976 
977 
978 
979 
979 
980 
981 
982 
983 
984 
985 
986 
987 
988 
989 
989 
990 
991 
992 
993 
994 
995 
996 
997 
998 
999 
999 
1000
```

## OUTPUT:

The screenshot shows the Visual Studio Code interface with an AI-assisted development extension. The left sidebar has sections for RUN AND DEBUG, RUN, and Python configurations. The main editor shows a Python file named 'day.py' with the following code:

```
36     class Library:
37         def return_book(self, title):
38             print(f"'{title}' was not borrowed.")
39
40             # Example usage
41             library = Library()
42             library.add_book(book("1984", "George Orwell"))
43             library.display_books()
44             library.borrow_book("1984")
45             library.display_books()
46             library.return_book("1984")
47             library.display_books()
```

The terminal below shows the execution of the code, displaying the state of the library:

```
'--> C:\Users\girug\Downloads\AI\AI.py'
1984 by George Orwell - Available
To Kill a Mockingbird by Harper Lee - Available
You have borrowed '1984'.
1984 by George Orwell - Borrowed
To Kill a Mockingbird by Harper Lee - Available
You have returned '1984'.
1984 by George Orwell - Available
1984 by George Orwell - Available
To Kill a Mockingbird by Harper Lee - Available
You have borrowed '1984'.
1984 by George Orwell - Borrowed
To Kill a Mockingbird by Harper Lee - Available
You have returned '1984'.
1984 by George Orwell - Available
1984 by George Orwell - Borrowed
To Kill a Mockingbird by Harper Lee - Available
You have returned '1984'.
1984 by George Orwell - Available
You have returned '1984'.
1984 by George Orwell - Available
To Kill a Mockingbird by Harper Lee - Available
PS C:\Users\girug\Downloads\AI> []
```

The status bar at the bottom indicates the code is in Python mode, with line 72, column 1, and other standard status information.

**Justification:** AI-assisted coding helped speed up development and provided a structured solution. However, human review was necessary to improve input validation, error handling, and real-world usability. Responsible AI use requires understanding and modifying generated code rather than copying blindly.

### Task Description – 4: AI-Assisted Code Completion for Class-

Based Attendance System

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

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder structure with files like `Alpy`, `day.py`, and `Alpy >...`.
- Code Editor:** Displays Python code for a `Student` class with methods for marking attendance and displaying results.
- Terminal:** Shows the output of running the code, which prints the attendance status for a student named Alice over three days.
- Run View:** Shows a breakdown of run configurations: `powershell`, `Python`, and `Python Deb...`.
- Breakpoints:** A sidebar showing breakpoints: `Raised Excepti...`, `Uncaught Excepti...` (which is checked), and `User Uncaught...`.
- AI Panel:** An AI panel titled "Build with Agent" is visible on the right, with a message about AI responses being inaccurate and a button to "Generate Agent Instructions".
- Status Bar:** Shows the current line (Ln 94, Col 29), spaces (Spaces: 4), encoding (UTF-8), and file type (Python).

## OUTPUT:

This screenshot is similar to the previous one but with the AI panel expanded:

- The AI panel now includes a text input field: "Describe what to build next".
- The "Agent" dropdown menu is open, showing options: "Agent", "Auto", and "Off".
- The "Go Live" button is also visible in the status bar.

## Observations:

- Dictionary stores student name and attendance

- Loop iterates through records
- Simple and efficient design

## Task Description – 5: AI-Based Code Completion for Conditional

### Menu Navigation

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

The screenshot shows a code editor window with the following details:

- File:** Welcome / day.py
- Editor Content:**

```

95     #task5
96     #generate a Python program using loops and conditionals to simulate an ATM menu
97     def atm_menu():
98         balance = 1000 # Initial balance
99         while True:
100             print("\nATM Menu:")
101             print("1. Check Balance")
102             print("2. Deposit Money")
103             print("3. Withdraw Money")
104             print("4. Exit")
105
106             choice = input("Choose an option (1-4): ")
107
108             if choice == '1':
109                 print(f"Your current balance is: ${balance}")
110             elif choice == '2':
111                 amount = float(input("Enter amount to deposit: "))
112                 if amount > 0:
113                     balance += amount
114                     print(f"${amount} deposited successfully.")
115                 else:
116                     print("Invalid amount. Please enter a positive number.")
117             elif choice == '3':
118                 amount = float(input("Enter amount to withdraw: "))
119                 if 0 < amount <= balance:
120                     balance -= amount
121                     print(f"${amount} withdrawn successfully.")
122                 else:
123                     print("Invalid amount or insufficient funds.")
124             elif choice == '4':
125                 print("Thank you for using the ATM. Goodbye!")
126                 break
127             else:
128                 print("Invalid choice. Please select a valid option.")
129
atm_menu()

```
- SIDE BAR:** Shows sections for VARIABLES, WATCH, CALL STACK (Running), and BREAKPOINTS. It also includes a "Build with Agent" button with instructions: "Al responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase."
- Bottom Bar:** Includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected. A status bar at the bottom shows: Line 130, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python | powershell | Go Live.

## OUTPUT:

```
ATM Menu:
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
choose an option (1-4): 2
Enter amount to deposit: 20
$20.0 deposited successfully.

ATM Menu:
1. Check Balance
2. Deposit Money
choose an option (1-4): 2
Enter amount to deposit: 20
$20.0 deposited successfully.

ATM Menu:
1. Check Balance
2. Deposit Money
ATM Menu:
1. Check Balance
2. Deposit Money
1. Check Balance
2. Deposit Money
2. Deposit Money
3. Withdraw Money
4. Exit
Choose an option (1-4): 4
Thank you for using the ATM. Goodbye!
PS C:\Users\girug\Downloads\ATI>
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder structure with files like `Alpy`, `__init__.py`, and `day.py`.
- Run and Debug View:** Displays the `Alpy` configuration.
- Terminal:** Shows the execution of the `Alpy` script, which runs an ATM menu. It performs a deposit of \$20.00 twice and then exits the program.
- Breakpoints:** A sidebar shows breakpoints for the `Alpy` file, with one uncaught exception breakpoint selected.
- AI Assistant:** A sidebar titled "Build with Agent" is visible, with a message about AI responses being inaccurate and a button to "Generate Agent Instructions".
- Status Bar:** Shows the current line (Ln 130), column (Col 1), spaces (Spaces: 4), encoding (UTF-8), and file (C:\Users\girug\Downloads\ATI).

## Justification:

- Correct balance update
- Prevents overdraft
- Loop exits safely