

AI-Assisted Coding

Assignment-6.3

Name: A.Sai Shrehan

Rollno: 2303A52299

Batch: 45

Task1: Classes (Student Class)

=>Use an AI tool (GitHub Copilot) to complete a Student class.

=>The class should include attributes such as name, roll number, and branch.

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

- EXPLORER:** Shows files like `2299_AI_AC_assignment_4.5.py`, `2303A52299 AI A-C Ass-4.3.docx`, and several PDF files.
- CODE EDITOR:** Displays the `Asignment_6.3.py` file with the following code:

```
class student:  
    """Represents a single student with basic information."""  
  
    def __init__(self, student_id: int, name: str, email: str, gpa: float):  
        """  
        Initialize a Student object.  
  
        Args:  
            student_id (int): Unique identifier for the student  
            name (str): Student's full name  
            email (str): Student's email address  
            gpa (float): Student's Grade Point Average (0.0 - 4.0)  
        """  
        self.student_id = student_id  
        self.name = name  
        self.email = email  
        self.gpa = gpa  
  
    def __str__(self) -> str:  
        """Return a string representation of the student."""  
        return f"ID: {self.student_id} | Name: {self.name} | Email: {self.email} | GPA: {self.gpa}"  
  
    def __repr__(self) -> str:  
        """Return a detailed string representation of the student."""  
        return f"Student(id={self.student_id}, name='{self.name}', email='{self.email}', gpa={self.gpa})"
```

- TERMINAL:** Shows the command `python.exe "c:/Users/sushri/Documents/3-2/AI-Assisted coding/Asignment_6.3.py"` and its output:

```
ID: 101 | Name: Alice Johnson | Email: alice@university.edu | GPA: 3.95  
ID: 102 | Name: Bob Smith | Email: bob@university.edu | GPA: 3.5  
ID: 103 | Name: Charlie Brown | Email: charlie@university.edu | GPA: 3.9  
ID: 104 | Name: Diana Prince | Email: diana@university.edu | GPA: 3.6
```

=>Add a method `display_details()` to print student information.

The screenshot shows the VS Code interface with the code editor open to `Assignment_6.3.py`. The code defines a `StudentManager` class with methods `update_student` and `list_all_students`. A new method `display_details` is added to the class. The code uses string interpolation and f-strings to format student information. The terminal shows the output of running the script, which includes the successful addition of a student named Eve Wilson. The status bar at the bottom indicates the file is 24 columns long and 5 spaces wide, and the Python extension is active.

```
assignment_3.5.py Assignment_6.3.py U X
Assignment_6.3.py > StudentManager > get_student
53 class StudentManager:
106     def update_student(self, student_id: int, name: str = None,
107                         email: str = None, gpa: float = None) -> bool:
108         """
109             Update a student's information.
110
111         Args:
112             student_id (int): The ID of the student to update
113             name (str, optional): New name
114             email (str, optional): New email
115             gpa (float, optional): New GPA
116
117         Returns:
118             bool: True if updated successfully, False if student not found
119
120         student = self.get_student(student_id)
121         if student is None:
122             print(f"Error: Student with ID {student_id} not found.")
123             return False
124         student.update_info(name, email, gpa)
125         print(f"Student {student.name} updated successfully.")
126         return True
127
128     def list_all_students(self) -> None:
129         """Display all students in the system."""
130         if not self.students:
131             print("No students in the system.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

PS C:\Users\Sushri\Documents\3-2\AI-Assisted coding & C:\Users\Sushri\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/Sushri/Documents/3-2/AI-Assisted coding/Assignment_6.3.py"

Student Eve Wilson added successfully.

CHAT STUDENT INFORMATION MANAGEMENT MODULE DEVELOPMENT

Perfect! ✅ The code has been successfully updated and executed. Here's a summary of what was implemented:

Updates Made:

- Modified Student Class Attributes:
 - Replaced email and GPA with `roll_number` and `branch`
 - Now includes: name, `roll_number`, and `branch` as required
- Added `display_details()` Method:
 - Prints formatted student information in a detailed box format
 - Shows ID, Name, Roll Number, and Branch clearly
- Updated Methods:
 - `update_info()` and `update_student()` now work with `roll_number` and `branch`
 - Added `filter_by_branch()` to find students by their branch
 - Updated string representations (`__str__` and `__repr__`)
- Demo Execution Results:
 - Created 5 students with different branches (Computer Science, Electronics, Mechanical)
 - Listed all students successfully
 - Displayed detailed information

0 + Assignment_6.3.py

=>Execute the code and verify the output.

The screenshot shows the VS Code interface with the code editor open to `Assignment_6.3.py`. The code defines a `StudentManager` class with methods `update_student`, `list_all_students`, and the newly added `display_details`. The code uses string interpolation and f-strings to format student information. The terminal shows the output of running the script, which includes the successful addition of a student named Eve Wilson and the execution of the `display_details` method for three students. The status bar at the bottom indicates the file is 24 columns long and 5 spaces wide, and the Python extension is active.

```
File Edit Selection View Go Run Terminal Help <- > Q: AI assisted coding
EXPLORER AI ASSISTED CODING
assignment_6.3.py x
assignment_6.3.py > StudentManager
56     # Create student objects
57     student1 = Student("Alice Johnson", 101, "Computer Science")
58     student2 = Student("Bob Smith", 102, "Mechanical Engineering")
59     student3 = Student("Carol White", 103, "Electrical Engineering")
60
61     # Display student information
62     print("Student Information:")
63     print("." * 50)
64     print(student1)
65     print(student2)
66     print(student3)
67
68     # Display detailed information using display_details method
69     print("\nDetailed Student Information:")
70     student1.display_details()
71     student2.display_details()
72     student3.display_details()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

PS D:\AI assisted coding> cd "d:\AI assisted coding" ; python assignment_6.3.py

Student Information:

Name: Alice Johnson, Roll Number: 101, Branch: Computer Science
Name: Bob Smith, Roll Number: 102, Branch: Mechanical Engineering
Name: Carol White, Roll Number: 103, Branch: Electrical Engineering

Detailed Student Information:

Student Details:

Name: Alice Johnson
Roll Number: 101
Branch: Computer Science

Execution Results - Verified Output:

1. Basic student objects are created with name, roll number, and branch.

Describe what to build next

Agent Auto

In 24, Col 5 Spaces: 4 UTF-8 CRLF {} Python 3.14.2 ⓘ Go Live

=>Analyze the code generated by the AI tool for correctness and clarity.

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

- EXPLORER** panel: Shows files like assignment_3.5.py, assignment_4.3.py, assignment_4.5.py, and assignment_6.3.py.
- CODE** panel: Displays Python code for a `Student` class and its `display_details` method.
- TERMINAL** panel: Shows the command `PS D:\AI assited coding> cd "d:\AI assited coding" ; python assignment_6.3.py` and its output, which includes creating student objects and displaying their details.
- CHAT** panel: Provides feedback on the code implementation.
- EXECUTION RESULTS**:
 - New Method Added:** `display_details()` : Prints student information in a formatted, easy-to-read layout with separators.
 - Execution Results - Verified Output:** The code runs successfully with all features working.
 - Summary:** Basic student objects are created with name, roll number, and branch. String representation displays all students in a single line. Detailed display shows each student's information in a formatted box with:
 - Student Details header
 - Name, Roll Number, and Branch on separate lines
 - Clear visual separators (= =)
 - Dictionary conversion works correctly
 - Update methods successfully modify student data
- OUTPUT** panel: Shows the output of the executed code.
- STATUS BAR**: Shows file path, spaces, encoding, Python version, and line numbers.

Task Description #2: Loops (Multiples of a Number)

=>generate a function to print first 10 multiple of a given number

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

- EXPLORER** panel: Shows files like assignment_6.3.py.
- CODE** panel: Displays Python code for a `Student` class and a `print_multiples` function.
- TERMINAL** panel: Shows the command `PS D:\AI assited coding> cd "d:\AI assited coding" ; python assignment_6.3.py` and its output, which includes creating student objects and calling the `print_multiples` function.
- CHAT** panel: Provides feedback on the code implementation.
- EXECUTION RESULTS**:
 - Initial Feedback:** You are writing a utility function to display multiples of a given number. generate a function to print first 10 multiple of a given number.
 - Plan:** I'll create a short plan, then add the `print_multiples` utility to `assignment_6.3.py` and run the file to verify output.
 - Implementation:** Now I'll add `print_multiples()` to the file and include a demo call in `__main__`.
 - Test:** Next I'll add a demo call to `print_multiples()` in `__main__` and run the script to show output.
 - Completion:** Done — I added `print_multiples(number, count=10)` to `assignment_6.3.py`, added a demo call `print_multiples(7)`, and ran the script. Output verified showing the first 10 multiples of 7. Would you like a version that returns the multiples as a list instead of printing?
- OUTPUT** panel: Shows the output of the executed code.
- STATUS BAR**: Shows file path, spaces, encoding, Python version, and line numbers.

=>Ask the AI to generate the same functionality using another controlled looping structure (e.g.,while instead of for).

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows files `assignment_6.3.py` and `assignment_6.3.ipynb`.
- Code Editor:** Displays three functions:
 - `def generate_multiples(number, count=10):`
 - `def generate_multiples_while(number, count=10):`
 - `def print_multiples_compact(number, count=10):`
- Terminal:** Shows the command `PS D:\AI assited coding> cd "d:\AI assited coding"; python assignment_6.3.py` and its output:
 - For `generate_multiples`:
Demo: Generate multiples using for loop (return as list)
Multiples of 5: [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
 - For `generate_multiples_while`:
Demo: Generate multiples using while loop
Multiples of 3: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30]
 - For `print_multiples_compact`:
First 10 multiples of 6 (using list comprehension):
6 12 18 24 30 36 42 48 54 60
- Output:** Shows the output of the terminal command.
- Debug Console:** Shows the command `ps D:\AI assited coding`.
- Terminal:** Shows the command `PS D:\AI assited coding>`.
- Chat:** Shows a conversation about the assignment, including a demo section and a note to update the demo section.
- Bottom Status Bar:** Shows file paths, line numbers (116, Col 23), and other status information.

Task Description #3: Conditional Statements (Age Classification)

You are building a basic classification system based on age. nested if-elif-else conditional statements to classify age groups(e.g., child, teenager, adult, senior).

The screenshot shows a Python script named `assignment_6.3.py` in a code editor. The code defines a class `AgeClassifier` with a static method `classify_age` that uses nested if-elif-else statements to classify ages into groups: Child, Teenager, Adult, or Senior. The code editor interface includes tabs for File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar at the top. Below the code, the terminal window shows the script being run in a Python environment, displaying the first 10 multiples of 6 and the classification results for various ages.

```
File Edit Selection View Go Run Terminal Help < > Q AI assisted coding
assignment_6.3.py X
assignment_6.3.py > ...
# You are building a basic classification system based on age. Nested if-elif-else conditional statements to classify age groups(e.g., child, teenager, adult, senior).
class AgeClassifier:
    """A class to classify age groups."""
    @staticmethod
    def classify_age(age):
        """
        Classify the age into different groups.

        Args:
            age (int): The age to classify.

        Returns:
            str: The age group classification.
        """
        if age < 0:
            return "Invalid age"
        elif age <= 12:
            return "Child"
        elif age <= 19:
            return "Teenager"
        elif age <= 59:
            return "Adult"
        else:
            return "Senior"
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
PS D:\AI assisted coding> & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assisted coding/assignment_6.3.py"
First 10 multiples of 6 (using list comprehension):
[6, 12, 18, 24, 30, 36, 42, 48, 54, 60]
Age Classification using nested if-elif-else:
Age: 5 - Classification: Child
Age: 15 - Classification: Teenager
Age: 30 - Classification: Adult
Age: 70 - Classification: Senior
Age: -3 - Classification: Invalid age
Ln 200, Col 55 Spaces: 4 UTF-8 CRLF { } Python 3.14.2 ⓘ Go Live

```

#generate generate the same classification using alternative conditional structures (simplified conditions or dictionary-based logic).

The screenshot shows the PyCharm IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and Run.
- Code Editor:** The main window displays the Python code for `assignment_6.3.py`. The code defines a class `AgeClassifier` with methods `classify_age_dict` and `display_classification_dict`, and a `main` function.
- Terminal:** The bottom terminal window shows the execution of the script and its output:

```
First 10 multiples of 6 (using list comprehension):  
[6, 12, 18, 24, 30, 36, 42, 48, 54, 60]  
Age Classification using nested if-elif-else:  
Age: 5 - Classification: Child  
Age: 15 - Classification: Teenager  
Age: 30 - Classification: Adult  
Age: 70 - Classification: Senior  
Age: -3 - Classification: Invalid age
```
- Status Bar:** Shows the current file path as `D:\AI assisted coding & C:/Users/vaish/AppData/Local/Programs/Python/Python314/python.exe "d:/AI assisted coding/assignment_6.3.py"`.
- Bottom Icons:** Includes icons for problem, output, debug console, terminal, ports, and postman console.

Task Description #4: For and While Loops (Sum of First n Numbers)

```
# generate a sum_to_n() function using a for loop.
```

#suggest an alternative implementation using the formula $n(n + 1)/2$.

```
File Edit Selection View Go Run Terminal Help < → Q AI assisted coding
assignment_6.3.py X
assignment_6.3.py > display_sum.formula
210 #suggest an alternative implementation using the formula n(n + 1)/2.
211 def sum_to_n_formula(n):
212     """Calculate the sum of the first n natural numbers using the formula."""
213     if n <= 0:
214         return 0
215     return n * (n + 1) // 2
216 def display_sum(n):
217     """Display the sum of the first n natural numbers."""
218     total = sum_to_n(n)
219     print(f"the sum of the first {n} natural numbers is: {total}")
220 def display_sum_formula(n):
221     """Display the sum of the first n natural numbers using the formula."""
222     total = sum_to_n_formula(n)
223     print(f"the sum of the first {n} natural numbers (using formula) is: {total}")
224 if __name__ == "__main__":
225     n = 10
226     display_sum(n)
227     display_sum_formula(n)
228
229
```

Task Description #5: Classes (Bank Account Class)

#you are designing a banking application generate a Bank Account class with methods such as deposit(), withdraw(),and check_balance()

#Add meaningful comments and explain the working of the code

```
File Edit Selection View Go Run Terminal Help ← → Q AI assisted coding

assignment_6.3.py (1)
assignment_6.3.py 1 BankAccount
class BankAccount:
    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
    260
    261
    262
    263
    264
    265
    266
    267
    268
    269
    270
    271
    272
    273
    274
    275
    276
    277
    278
    279
    280
    281
    282
    283
    284
    285
    286
    287
    288
    289
    290
    291
    292
    293
    294
    295

def __str__(self):
    """Return a readable string representation of the bank account"""
    return f"Account Holder: {self.account_holder}, Balance: ${self.balance:.2f}"

def display_account_details(self):
    """Print account details in a formatted way."""
    print("-----")
    print(f"Initial Account Details:")
    print(f"Name: {self.name}")
    print(f"Balance: {self.balance:.2f}")
    print(f"Balance: ${self.balance:.2f}")

if __name__ == "__main__":
    # Create a bank account
    account = BankAccount("John Doe", 1000)

    # Display account details
    print("Initial Account Details:")
    account.display_account_details()

    # Deposit and withdraw money
    account.deposit(500)
    account.withdraw(200)

    # Check balance
    account.check_balance()

    # Display updated account details
    print("Updated Account Details:")
    account.display_account_details()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

PS D:\AI assisted coding & C:/Users/valish/AppData/Local/Programs/Python/Python311/python.exe "d:/AI assisted coding/assignment_6.3.py"
Updated Account Details:
-----
Bank Account Details:
-----
Account Holder: John Doe
Balance: $1300.00
-----
PS D:\AI assisted coding> []
x main() 0 ⌂ ⌂ 1
Snipping Tool
```