

2303A510B7 (Harini)

Batch - 14

AI - Assisted Coding

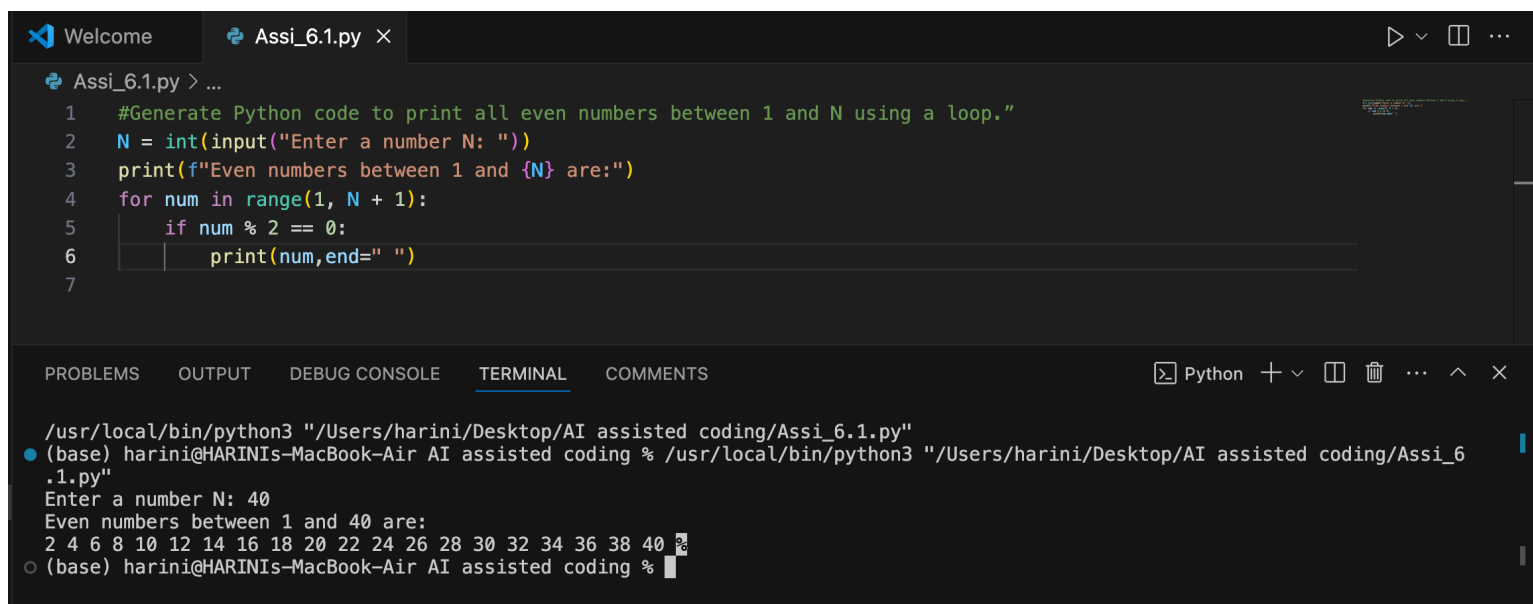
Assignment - 6.1

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

Task: Use an AI code completion tool to generate a loop-based program.

Prompt 1 : # Generate Python code to print all even numbers between 1 and N using a loop.”

Code & OutPut:



The screenshot shows a code editor with a file named 'Assi_6.1.py'. The code is a Python program that prompts the user for a number N and prints all even numbers between 1 and N. The terminal output shows the program being executed, the user entering 40, and the program printing the even numbers from 2 to 40.

```
Assi_6.1.py > ...
1 #Generate Python code to print all even numbers between 1 and N using a loop."
2 N = int(input("Enter a number N: "))
3 print(f"Even numbers between 1 and {N} are:")
4 for num in range(1, N + 1):
5     if num % 2 == 0:
6         print(num,end=" ")
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

Python + - [] [X] [Y] [Z] [W] [V] [U] [T] [S] [R] [Q] [P] [O] [N] [M] [L] [K] [J] [I] [H] [G] [F] [E] [D] [C] [B] [A]

```
/usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
(base) harini@HARINIs-MacBook-Air AI assisted coding % /usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
Enter a number N: 40
Even numbers between 1 and 40 are:
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 %
(base) harini@HARINIs-MacBook-Air AI assisted coding %
```

Code & OutPut:

```
Assi_6.1.py > ...
8 # Generate Python code to count how many numbers in a list are even and odd user defined inputs
9 N = int(input("Enter the number of elements in the list: "))
10 numbers = []
11 for i in range(N):
12     num = int(input(f"Enter number {i+1}: "))
13     numbers.append(num)
14 even_count = 0
15 odd_count = 0
16 for num in numbers:
17     if num % 2 == 0:
18         even_count += 1
19     else:
20         odd_count += 1
21 print(f"Total even numbers: {even_count}")
22 print(f"Total odd numbers: {odd_count}")
23
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

```
/usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
(base) harini@HARINIs-MacBook-Air AI assisted coding % /usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
Enter the number of elements in the list: 6
Enter number 1: 6
Enter number 2: 8
Enter number 3: 9
Enter number 4: 10
Enter number 5: 30
Enter number 6: 22
Total even numbers: 5
Total odd numbers: 1
(base) harini@HARINIs-MacBook-Air AI assisted coding %
```

Prompt 3 : Generate a Python class User that validates age and email using conditional statements.

Code :

```
Assi_6.1.py > ...
24 # Generate a Python class User that validates age and email using conditional statements
25 class User:
26     def __init__(self, name, age, email):
27         self.name = name
28         self.age = age
29         self.email = email
30
31     def validate_age(self):
32         if 0 < self.age < 120:
33             return True
34         else:
35             return False
36
37     def validate_email(self):
38         if "@" in self.email and "." in self.email.split("@")[-1]:
39             return True
40         else:
41             return False
42
43 # Example usage
44 n=int(input("Enter number of users to validate: "))
45 for i in range(n):
46     name = input(f"Enter name of user {i+1}: ")
47     age = int(input(f"Enter age of user {i+1}: "))
48     email = input(f"Enter email of user {i+1}: ")
49     user = User(name, age, email)
50     if user.validate_age():
51         print(f"{name}'s age is valid.")
52     else:
53         print(f"{name}'s age is invalid.")
54     if user.validate_email():
55         print(f"{name}'s email is valid.")
56     else:
57         print(f"{name}'s email is invalid.")
```

OutPut :

✓ TERMINAL

```
/usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
● (base) harini@HARINIs-MacBook-Air AI assisted coding % /usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
Enter number of users to validate: 2
Enter name of user 1: Harini
Enter age of user 1: 21
Enter email of user 1: hani@gmail.com
Harini's age is valid.
Harini's email is valid.
Enter name of user 2: Vishnu
Enter age of user 2: 21
Enter email of user 2: vishnu@gmail.com
Vishnu's age is valid.
Vishnu's email is valid.
```

Prompt 4: # Generate a Python class Student with attributes (name, roll number, marks) and methods to calculate total and average marks

Code:

```
# Generate a Python class Student with attributes (name, roll number, marks) and methods
# to calculate total and average marks."
class Student:
    def __init__(self, name, roll_number, marks):
        self.name = name
        self.roll_number = roll_number
        self.marks = marks # marks should be a list of integers

    def calculate_total(self):
        return sum(self.marks)

    def calculate_average(self):
        if len(self.marks) == 0:
            return 0
        return self.calculate_total() / len(self.marks)

# Example usage
n=int(input("Enter number of students: "))
for i in range(n):
    name = input(f"Enter name of student {i+1}: ")
    roll_number = input(f"Enter roll number of student {i+1}: ")
    marks_input = input(f"Enter marks of student {i+1} separated by spaces: ")
    marks = list(map(int, marks_input.split()))
    student = Student(name, roll_number, marks)
    total_marks = student.calculate_total()
    average_marks = student.calculate_average()
    print(f"Student: {name}, Roll Number: {roll_number}, Total Marks: {total_marks}, Average Marks: {average_marks:.2f}")
```

Output:

```
▼ TERMINAL Python + - [] ...
/usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
• (base) harini@HARINIs-MacBook-Air AI assisted coding % /usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
Enter number of students: 2
Enter name of student 1: Harini
Enter roll number of student 1: 10B7
Enter marks of student 1 separated by spaces: 89 90 75 80 99
Student: Harini, Roll Number: 10B7, Total Marks: 433, Average Marks: 86.60
Enter name of student 2: Vishnu
Enter roll number of student 2: 10A0
Enter marks of student 2 separated by spaces: 100 100 100 100 100
Student: Vishnu, Roll Number: 10A0, Total Marks: 500, Average Marks: 100.00
○ (base) harini@HARINIs-MacBook-Air AI assisted coding %
```

Prompt 5: # Generate a Python program for a simple bank account system using class, loops, and conditional statements.

Code:

```
# Generate a Python program for a simple bank account system using class, loops, and conditional statements.
class BankAccount:
    def __init__(self, account_holder, balance=0):
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited: ${amount:.2f}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        if amount > 0:
            if amount <= self.balance:
                self.balance -= amount
                print(f"Withdrew: ${amount:.2f}")
            else:
                print("Insufficient funds.")
        else:
            print("Withdrawal amount must be positive.")

    def display_balance(self):
        print(f"Account Holder: {self.account_holder}, Balance: ${self.balance:.2f}")

# Example usage
account_holder = input("Enter account holder name: ")
account = BankAccount(account_holder)
while True:
    print("\nOptions:")
    print("1. Deposit")
    print("2. Withdraw")
    print("3. Display Balance")
    print("4. Exit")
    choice = input("Choose an option (1-4): ")
    if choice == '1':
        amount = float(input("Enter amount to deposit: "))
        account.deposit(amount)
    elif choice == '2':
        amount = float(input("Enter amount to withdraw: "))
        account.withdraw(amount)
    elif choice == '3':
        account.display_balance()
    elif choice == '4':
        print("Exiting the program.")
        break
    else:
        print("Invalid choice. Please try again.")
```

Output:

```
✓ TERMINAL Python + ▢ 🗑️ ...

/usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
● (base) harini@HARINIs-MacBook-Air AI assisted coding % /usr/local/bin/python3 "/Users/harini/Desktop/AI assisted coding/Assi_6.1.py"
Enter account holder name: Harini

Options:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Choose an option (1-4): 1
Enter amount to deposit: 2000000
Deposited: $2000000.00

Options:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Choose an option (1-4): 2
Enter amount to withdraw: 10000
Withdrew: $10000.00

Options:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Choose an option (1-4): 3
Account Holder: Harini, Balance: $1990000.00

Options:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Choose an option (1-4): 4
Exiting the program.
○ (base) harini@HARINIs-MacBook-Air AI assisted coding %
```

