Assignment 6.4

Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals

Name: Mohd. Aadil Ashraf

Roll No.: 2403A51315

Batch: 13 CSE CORE

Subject: AI Assisted Coding

Date: 10-09-2025

# Objectives

• Explore AI-powered code completion features.

• Generate class definitions, loops, and conditionals using AI assistance.

• Evaluate AI-suggested code for correctness and clarity.

# Task 1: Student Class

Prompt given to AI:  
Start a Python class named Student with attributes name, roll\_number, and marks. Complete methods for displaying details and checking if marks are above average.

AI-Assisted Code:

class Student:  
 def \_\_init\_\_(self, name, roll\_number, marks):  
 self.name = name  
 self.roll\_number = roll\_number  
 self.marks = marks  
  
 def display\_details(self):  
 print("Name:", self.name)  
 print("Roll Number:", self.roll\_number)  
 print("Marks:", self.marks)  
  
 def is\_passed(self):  
 if self.marks >= 40:  
 return "Passed"  
 else:  
 return "Failed"  
  
# Example  
s1 = Student("Aadil", "2403A51315", 72)  
s1.display\_details()  
print("Result:", s1.is\_passed())

Output (Screenshot):

Explanation:  
- Class created with 3 attributes.  
- display\_details() prints details.  
- is\_passed() checks marks using if-else.

# Task 2: For Loop with Even Squares

Prompt given to AI:  
Write a for loop to iterate a list of numbers and print square of even numbers.

AI-Assisted Code:

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
  
for num in numbers:  
 if num % 2 == 0:  
 print(f"Square of {num} is {num\*num}")

Explanation:  
- Loop runs through list.  
- If condition checks even numbers only.  
- Prints their squares.

# Task 3: BankAccount Class

Prompt given to AI:  
Create a BankAccount class with deposit, withdraw, and balance check methods.

AI-Assisted Code:

class BankAccount:  
 def \_\_init\_\_(self, account\_holder, balance=0):  
 self.account\_holder = account\_holder  
 self.balance = balance  
  
 def deposit(self, amount):  
 self.balance += amount  
 print(f"Deposited {amount}, New Balance: {self.balance}")  
  
 def withdraw(self, amount):  
 if self.balance >= amount:  
 self.balance -= amount  
 print(f"Withdrawn {amount}, Remaining Balance: {self.balance}")  
 else:  
 print("Insufficient Balance!")  
  
# Example  
acc = BankAccount("Aadil", 1000)  
acc.deposit(500)  
acc.withdraw(300)  
acc.withdraw(1500)

Explanation:  
- Class created with deposit and withdraw.  
- Prevents overdrawing with condition.

# Task 4: While Loop with Student Scores

Prompt given to AI:  
Define a list of students and print names of those with score > 75 using while loop.

AI-Assisted Code:

students = [  
 {"name": "Aadil", "score": 80},  
 {"name": "Anya", "score": 65},  
 {"name": "Rahul", "score": 90}  
]  
  
i = 0  
while i < len(students):  
 if students[i]["score"] > 75:  
 print("Topper:", students[i]["name"])  
 i += 1

Explanation:  
- Uses while loop with index.  
- Checks score > 75.  
- Prints topper names.

# Task 5: ShoppingCart Class

Prompt given to AI:  
Create ShoppingCart class with add, remove, and total bill calculation with discounts.

AI-Assisted Code:

class ShoppingCart:  
 def \_\_init\_\_(self):  
 self.items = []  
  
 def add\_item(self, item, price):  
 self.items.append({"item": item, "price": price})  
 print(f"Added {item} - {price}")  
  
 def remove\_item(self, item):  
 self.items = [i for i in self.items if i["item"] != item]  
 print(f"Removed {item}")  
  
 def total\_bill(self):  
 total = 0  
 for i in self.items:  
 total += i["price"]  
 if total > 500:  
 total = total \* 0.9 # 10% discount  
 print("Total Bill:", total)  
  
# Example  
cart = ShoppingCart()  
cart.add\_item("Shoes", 600)  
cart.add\_item("Bag", 300)  
cart.total\_bill()  
cart.remove\_item("Bag")  
cart.total\_bill()

Explanation:  
- Items stored as dictionary in list.  
- For loop calculates bill.  
- Conditional discount applied if total > 500.

# Conclusion

This lab demonstrated AI-assisted code completion for classes, loops, and conditionals. AI suggestions were accurate and needed small modifications.









