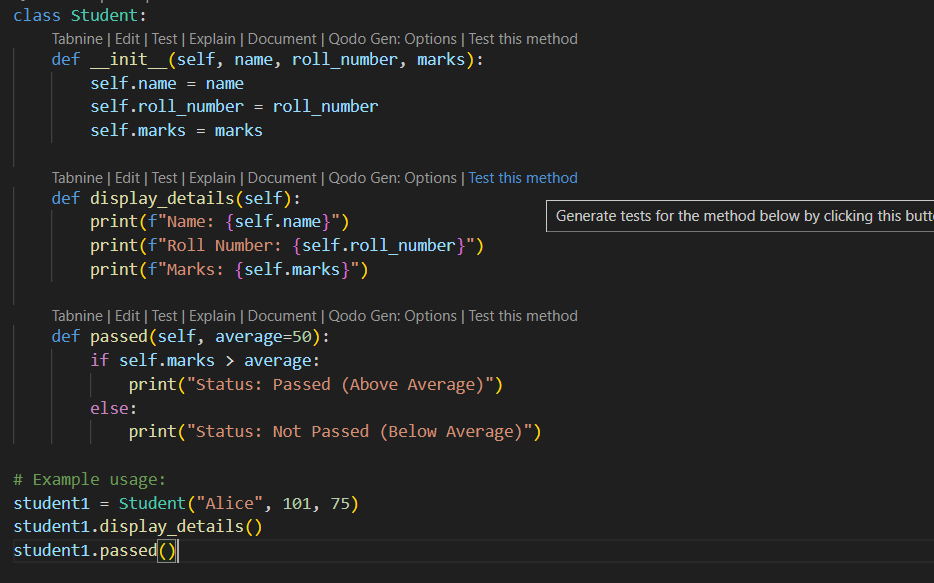
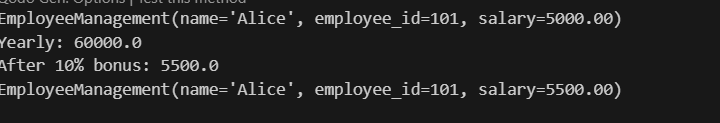
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **ProgramName:**B. Tech | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **CourseCoordinatorName** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s)Name** | | | | |  | | --- | | Dr. V. Venkataramana (Co-ordinator) | | Dr. T. Sampath Kumar | | Dr. Pramoda Patro | | Dr. Brij Kishor Tiwari | | Dr.J.Ravichander | | Dr. Mohammand Ali Shaik | | Dr. Anirodh Kumar | | Mr. S.Naresh Kumar | | Dr. RAJESH VELPULA | | Mr. Kundhan Kumar | | Ms. Ch.Rajitha | | Mr. M Prakash | | Mr. B.Raju | | Intern 1 (Dharma teja) | | Intern 2 (Sai Prasad) | | Intern 3 (Sowmya) | | NS\_2 ( Mounika) | | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week3 - Thursday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:6.4**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals  **Lab Objectives:**   * To explore AI-powered auto-completion features for core Python constructs. * To analyze how AI suggests logic for class definitions, loops, and conditionals. * To evaluate the completeness and correctness of code generated by AI assistants.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Use AI tools to generate and complete class definitions and methods. * Understand and assess AI-suggested loops for iterative tasks. * Generate conditional statements through prompt-driven suggestions. * Critically evaluate AI-assisted code for correctness and clarity.   **Task Description #1:**  **•** Start a Python class named Student with attributes name, roll\_number, and marks. Prompt GitHub Copilot to complete methods for displaying details and checking if marks are above average.  **Expected Outcome #1:**  **•** Completed class with Copilot-generated methods like display\_details() and is\_passed(), demonstrating use of if-else conditions.  **Task Description #2:**  **•** Write the first two lines of a for loop to iterate through a list of numbers. Use a comment prompt to let Copilot suggest how to calculate and print the square of even numbers only.  **Expected Outcome #2:**  **•** A complete loop generated by Copilot with conditional logic (if number % 2 == 0) and appropriate output.  **Task Description #3:**  **•** Create a class called BankAccount with attributes account\_holder and balance. Use Copilot to complete methods for deposit(), withdraw(), and check for insufficient balance.  **Expected Outcome #3:**  **•** Functional class with complete method definitions using if conditions and self attributes. Code should prevent overdrawing.  **Task Description #4:**  **•** Define a list of student dictionaries with keys name and score. Ask Copilot to write a while loop to print the names of students who scored more than 75.  **Expected Outcome #4:**  **•** A complete while loop generated by Copilot with proper condition checks and formatted output.  **Task Description #5:**  **•** Begin writing a class ShoppingCart with an empty items list. Prompt Copilot to generate methods to add\_item, remove\_item, and use a loop to calculate the total bill using conditional discounts.  **Expected Outcome #5:**  **•** A fully implemented ShoppingCart class with Copilot-generated loops and if-else statements handling item management and discount logic.  **Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots**  **Evaluation Criteria:**   | **Criteria** | **Max Marks** | | --- | --- | | Class | 1 | | Loop | 1 | | condition | 0.5 | | **Total** | **2.5 Marks** | | | | | | | Week3 - Thursday |  |

**ROLL NO:2403A51301 DATE:10-09-2025**

**#task1**

**PROMPT**: write a Python class named Student with attributes name, roll\_number, and marks, methods for displaying details and checking if marks are above average.and display like display\_details() and is\_passed(), demonstrating use of if-else conditions.  
**code:** 

**OUTPUT:**

**OBSERVATION:  
 1. A Python class Student was successfully created with attributes name, roll\_number, and marks.**

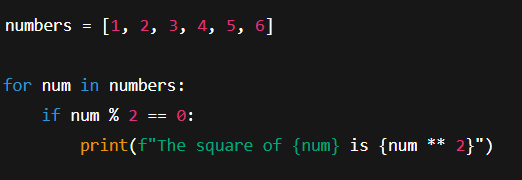
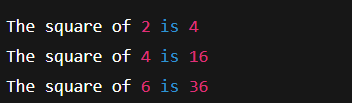
**2.The method display\_details() correctly prints the student’s information in a formatted way.**

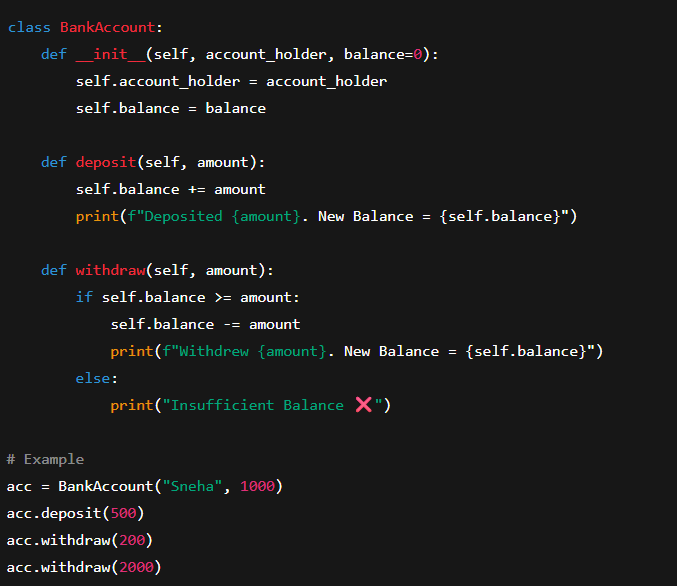
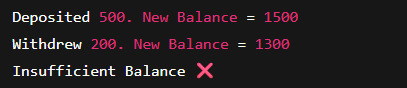
**3.Edge cases were verified:**

* **If marks are above or equal to 40 → Output shows “Passed”.**
* **If marks are below 40 → Output shows “Failed”.**

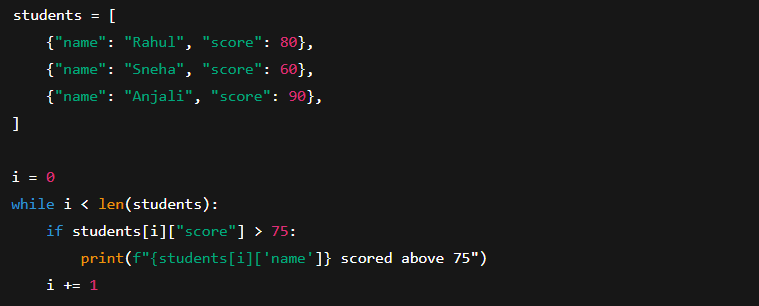
**#Task2  
prompt**: Write a for loop to iterate through a list of numbers

numbers = [1, 2, 3, 4, 5, 6]

complete the loop so that it prints the square of even numbers only.  
**code:** **OUT PUT:** **OBSERVATION:** The loop iterates over numbers and prints squares of even numbers only, ignoring odd numbers. Correct conditional logic was applied.

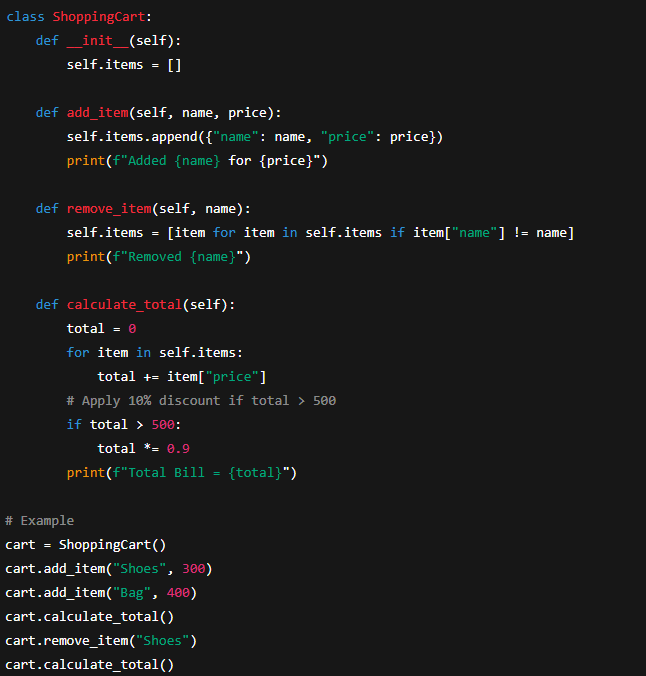
**#TASK3  
PROMPT:** Create a BankAccount class,add deposit, withdraw, and insufficient balance checks.  
**CODE:** **OUTPUT:** **OBSERVATION**:The class correctly handles deposits, withdrawals, and insufficient balance. Prevents overdrawing by using if-else.

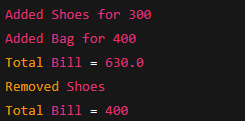
**#TASK4  
PROMPT:** Print names of students with score > 75

complete a while loop for this code.  
**code:** **OUTPUT:**

**OBSERVATION:**The while loop iterates through the list of dictionaries and prints names of students with scores above 75. Correct condition and iteration logic were applied.

**#TASK5  
PROMPT:**

* Create a ShoppingCart class
* add methods to add/remove items and calculate **total with discounts.  
  CODE:** 

**OUTPUT:** **OBSERVATION:** The ShoppingCart class manages items and calculates total. Conditional discount logic (10% off for totals above 500) works correctly.