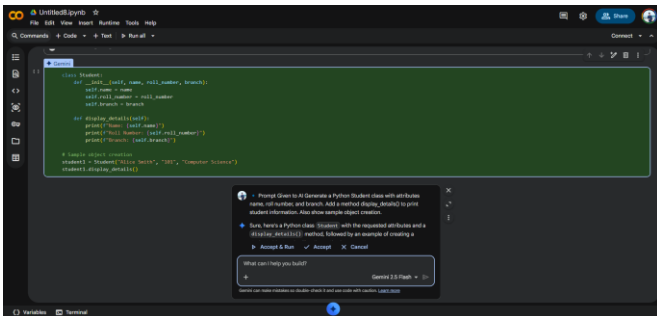
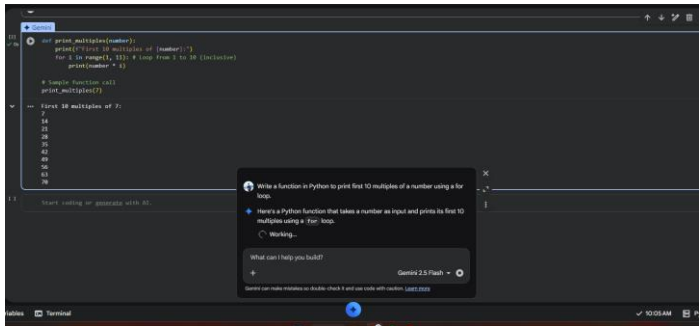
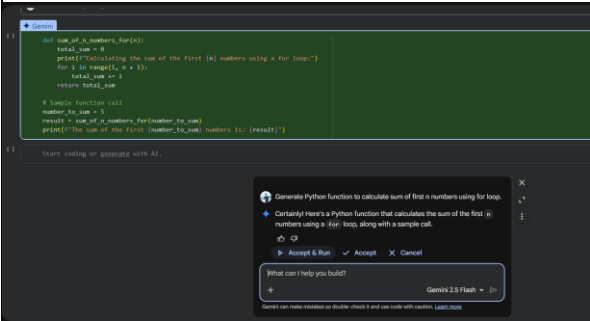


SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING																		
Program Name: B. Tech		Assignment Type: Lab	Academic Year:2025- 2026																	
Course Coordinator Name		Dr. Rishabh Mittal																		
Instructor(s) Name		<table border="1"> <tr><td>Mr. S Naresh Kumar</td></tr> <tr><td>Ms. B. Swathi</td></tr> <tr><td>Dr. Sasanko Shekhar Gantayat</td></tr> <tr><td>Mr. Md Sallauddin</td></tr> <tr><td>Dr. Mathivanan</td></tr> <tr><td>Mr. Y Srikanth</td></tr> <tr><td>Ms. N Shilpa</td></tr> <tr><td>Dr. Rishabh Mittal (Coordinator)</td></tr> <tr><td>Dr. R. Prashant Kumar</td></tr> <tr><td>Mr. Ankushavali MD</td></tr> <tr><td>Mr. B Viswanath</td></tr> <tr><td>Ms. Sujitha Reddy</td></tr> <tr><td>Ms. A. Anitha</td></tr> <tr><td>Ms. M.Madhuri</td></tr> <tr><td>Ms. Katherashala Swetha</td></tr> <tr><td>Ms. Velpula sumalatha</td></tr> <tr><td>Mr. Bingi Raju</td></tr> </table>		Mr. S Naresh Kumar	Ms. B. Swathi	Dr. Sasanko Shekhar Gantayat	Mr. Md Sallauddin	Dr. Mathivanan	Mr. Y Srikanth	Ms. N Shilpa	Dr. Rishabh Mittal (Coordinator)	Dr. R. Prashant Kumar	Mr. Ankushavali MD	Mr. B Viswanath	Ms. Sujitha Reddy	Ms. A. Anitha	Ms. M.Madhuri	Ms. Katherashala Swetha	Ms. Velpula sumalatha	Mr. Bingi Raju
Mr. S Naresh Kumar																				
Ms. B. Swathi																				
Dr. Sasanko Shekhar Gantayat																				
Mr. Md Sallauddin																				
Dr. Mathivanan																				
Mr. Y Srikanth																				
Ms. N Shilpa																				
Dr. Rishabh Mittal (Coordinator)																				
Dr. R. Prashant Kumar																				
Mr. Ankushavali MD																				
Mr. B Viswanath																				
Ms. Sujitha Reddy																				
Ms. A. Anitha																				
Ms. M.Madhuri																				
Ms. Katherashala Swetha																				
Ms. Velpula sumalatha																				
Mr. Bingi Raju																				
Course Code	23CS002PC304	Course Title	AI Assisted Coding																	
Year/Sem	III/II	Regulation	R23																	
Date and Day of Assignment	Week3 – Wednesday	Time(s)	23CSBTB01 To 23CSBTB52																	
Duration	2 Hours	Applicable to Batches	All batches																	
AssignmentNumber:6.3(Present assignment number)/24(Total number of assignments)																				
Q. No	Question		Expected Time to complete																	

	<p><b>Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals</b></p> <p><b>Lab Objectives</b></p> <ul style="list-style-type: none"> <li>• To explore AI-powered auto-completion features for core Python constructs such as classes, loops, and conditional statements.</li> <li>• To analyze how AI tools suggest logic for object-oriented programming and control structures.</li> <li>• To evaluate the correctness, readability, and completeness of AI-generated Python code.</li> </ul> <p><b>Lab Outcomes (LOs)</b> After completing this lab, students will be able to:</p> <ul style="list-style-type: none"> <li>• Use AI tools to generate and complete Python class definitions and methods.</li> <li>• Understand and assess AI-suggested loop constructs for iterative tasks.</li> <li>• Generate and evaluate conditional statements using AI-driven prompts.</li> <li>• Critically analyze AI-assisted code for correctness, clarity, and efficiency.</li> </ul>	
1	<p><b>Task Description #1: Classes (Student Class)</b></p> <p><b>Scenario</b> You are developing a simple student information management module.</p> <p><b>Task</b></p> <ul style="list-style-type: none"> <li>• Use an AI tool (GitHub Copilot / Cursor AI / Gemini) to complete a Student class.</li> <li>• The class should include attributes such as name, roll number, and branch.</li> <li>• Add a method <code>display_details()</code> to print student information.</li> <li>• Execute the code and verify the output.</li> <li>• Analyze the code generated by the AI tool for correctness and clarity.</li> </ul> <p><b>Expected Output #1</b></p> <ul style="list-style-type: none"> <li>• A Python class with a constructor (<code>__init__</code>) and a <code>display_details()</code> method.</li> <li>• Sample object creation and output displayed on the console.</li> <li>• Brief analysis of AI-generated code.</li> </ul> <p>Prompt</p> <p>Generate a Python Student class with attributes name, roll number, and branch. Add a method <code>display_details()</code> to print student information. Also show sample object creation.</p>  <p>The screenshot shows a code editor with a Python class named <code>Student</code>. The class has three attributes: <code>name</code>, <code>roll_number</code>, and <code>branch</code>. It has a constructor <code>__init__</code> that takes these three arguments and a method <code>display_details</code> that prints the attributes. Below the class definition, there is a sample object creation: <code>student1 = Student("Vijay", "12345", "CSE")</code> and a call to <code>display_details(student1)</code>. The output in the terminal shows the details of the student: <code>Vijay 12345 CSE</code>.</p>	Week3 - Wednes day
	<p><b>Task Description #2: Loops (Multiples of a Number)</b></p> <p><b>Scenario</b> You are writing a utility function to display multiples of a given number.</p>	

	<p><b>Task</b></p> <ul style="list-style-type: none"><li>• Prompt the AI tool to generate a function that prints the first 10 multiples of a given number using a loop.</li><li>• Analyze the generated loop logic.</li><li>• Ask the AI to generate the same functionality using another controlled looping structure (e.g., while instead of for).</li></ul> <p><b>Expected Output #2</b></p> <ul style="list-style-type: none"><li>• Correct loop-based Python implementation.</li><li>• Output showing the first 10 multiples of a number.</li><li>• Comparison and analysis of different looping approaches.</li></ul>	
	<p><b>Prompt</b></p> <p>Write a function in Python to print first 10 multiples of a number using a for loop.</p> <div data-bbox="289 821 984 1144"></div> <ul style="list-style-type: none"><li>• Both implementations correctly print the first 10 multiples.</li><li>• for loop is simpler and more readable.</li><li>• while loop gives more control over the iteration variable.</li><li>• AI-generated logic is correct and efficient.</li></ul> <p><b>Task Description #3: Conditional Statements (Age Classification)</b></p> <p><b>Scenario</b></p> <p>You are building a basic classification system based on age.</p> <p><b>Task</b></p> <ul style="list-style-type: none"><li>• Ask the AI tool to generate nested if-elif-else conditional statements to classify age groups (e.g., child, teenager, adult, senior).</li><li>• Analyze the generated conditions and logic.</li><li>• Ask the AI to generate the same classification using alternative conditional structures (e.g., simplified conditions or dictionary-based logic).</li></ul> <p><b>Expected Output #3</b></p> <ul style="list-style-type: none"><li>• A Python function that classifies age into appropriate groups.</li><li>• Clear and correct conditional logic.</li><li>• Explanation of how the conditions work.</li></ul>	

	<ul style="list-style-type: none"><li>• The nested if-elif-else structure is clear and easy to understand.</li><li>• The alternative version uses a list of conditions, making it flexible and cleaner.</li><li>• AI-generated conditions are logically correct.</li></ul> <ul style="list-style-type: none"><li>• The nested if-elif-else structure is clear</li></ul>	
	<p><b>Task Description #4: For and While Loops (Sum of First n Numbers)</b></p> <p><b>Scenario</b> You need to calculate the sum of the first n natural numbers.</p> <p><b>Prompt</b> Write the same program using a while loop.</p> <div>The screenshot shows a code editor with a dark theme. The top part displays Python code for a function 'sum_of_n_numbers_for(n)' that calculates the sum of the first n numbers using a for loop. Below the code, there's a sample function call and its output. The bottom part of the screenshot shows a chat window with an AI assistant. The chat message asks to 'Generate Python function to calculate sum of first n numbers using for loop.' The AI response provides the Python function and a sample call. There are buttons for 'Accept &amp; Run', 'Accept', and 'Cancel'.</div> <p><b>Task</b></p> <ul style="list-style-type: none"><li>• Use AI assistance to generate a sum_to_n() function using a for loop.</li><li>• Analyze the generated code.</li><li>• Ask the AI to suggest an alternative implementation using a while loop or a mathematical formula.</li></ul> <p><b>Expected Output #4</b></p> <ul style="list-style-type: none"><li>• Python function to compute the sum of first n numbers.</li><li>• Correct output for sample inputs.</li><li>• Explanation and comparison of different approaches.</li></ul> <p><b>Prompt</b></p>	

```
13 def sum_of_n_numbers_for(n):
14     total_sum = 0
15     print(f"Calculating the sum of the first {n} numbers using a for loop.")
16     for i in range(1, n + 1):
17         total_sum += i
18     return total_sum
19
20 # Simple function call
21 number_to_sum = 5
22 result = sum_of_n_numbers_for(number_to_sum)
23 print(f"The sum of the first {number_to_sum} numbers is: {result}")
24
25 Start coding or generate with AI.
```

Generate Python function to calculate sum of first n numbers using for loop.

Certainly! Here's a Python function that calculates the sum of the first n numbers using a for loop, along with a sample call.

Accept & Run

What can I help you build?

Gemini 2.5 Flash

## Task Description #5: Classes (Bank Account Class)

### Scenario

You are designing a basic banking application.

### Task

- Use AI tools to generate a Bank Account class with methods such as deposit(), withdraw(), and check\_balance().
- Analyze the AI-generated class structure and logic.
- Add meaningful comments and explain the working of the code.

### Expected Output #5

- Complete Python Bank Account class.
- Demonstration of deposit and withdrawal operations with updated balance.
- Well-commented code with a clear explanation.

**Note: Report should be submitted as a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots.**

### Prompt

write a class for bank application with deposit and withdraw methods

```
1 class BankAccount:
2     """Base class for BankAccount, account holder name, initial balance"""
3     def __init__(self, name: str, initial_balance: float):
4         self.name = name
5         self.initial_balance = initial_balance
6         self.balance = initial_balance
7         print(f"Account holder name: {self.name}, initial balance: {self.balance}")
8
9     def deposit(self, amount: float):
10         """Deposit method"""
11         if amount > 0:
12             self.balance += amount
13             print(f"Deposited {amount} to {self.name}. New balance: {self.balance}")
14         else:
15             print("Deposit amount must be positive.")
16
17     def withdraw(self, amount: float):
18         """Withdrawal method"""
19         if amount > 0:
20             self.balance -= amount
21             print(f"Withdrew {amount} from {self.name}. New balance: {self.balance}")
22         else:
23             print("Withdrawal amount must be positive.")
24
25     def check_balance(self):
26         """Check balance method"""
27         print(f"Current balance for {self.name}: {self.balance}")
28
29     def __str__(self):
30         return f"BankAccount({self.name}, {self.balance})"
```

Write a class for bank application with deposit and withdraw methods.

Certainly! Here's a Python class, BankAccount, with deposit and withdrawal methods, along with a sample of how to create an account and perform some transactions.

Accept & Run

What can I help you build?

Gemini 2.5 Flash

	<ul style="list-style-type: none"><li>• AI tools generated correct and readable Python code for classes, loops, and conditionals.</li><li>• The logic produced was mostly optimal and easy to understand.</li><li>• Alternative approaches (for vs while, if-else vs simplified logic, formula vs loops) helped in understanding efficiency.</li><li>• AI assistance speeds up development but human verification is required to ensure correctness and edge-case handling.</li></ul>	
--	---	--