

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	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: <b>6.3</b> (Present assignment number)/ <b>24</b> (Total number of assignments)			

Q.No.	Question	Expected Time to complete
1	<b>Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals</b> <b>Lab Objectives</b> <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>	Week3 - Wednesday

### Lab Outcomes (LOs)

After completing this lab, students will be able to:

- Use AI tools to generate and complete Python class definitions and methods.
- Understand and assess AI-suggested loop constructs for iterative tasks.
- Generate and evaluate conditional statements using AI-driven prompts.
- Critically analyze AI-assisted code for correctness, clarity, and efficiency.

### Task Description #1: Classes (Student Class)

#### Scenario

You are developing a simple student information management module.

#### Task

- Use an AI tool (GitHub Copilot / Cursor AI / Gemini) to complete a Student class.
- The class should include attributes such as name, roll number, and branch.
- Add a method `display_details()` to print student information.
- Execute the code and verify the output.
- Analyze the code generated by the AI tool for correctness and clarity.

#### Expected Output #1

- A Python class with a constructor (`__init__`) and a `display_details()` method.
- Sample object creation and output displayed on the console.
- Brief analysis of AI-generated code.

```
C:\> Users > pujar > OneDrive\လାଭକ > ai ass-6.3.py > ...
1  class Student:
2      def __init__(self, name, roll_number, branch):
3          self.name = name
4          self.roll_number = roll_number
5          self.branch = branch
6
7      def display_details(self):
8          print(f"Name: {self.name}")
9          print(f"Roll Number: {self.roll_number}")
10         print(f"Branch: {self.branch}")
11
12 if __name__ == "__main__":
13     s = Student("Alice Smith", "CS2026", "Computer Science")
14     s.display_details()
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\pujar\OneDrive\လାଭକ & 'C:\Users\pujar\anaconda3\python.exe' 'c:\Users\pujar\.vscode\extensions\ms-python.python-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52158' '--' 'c:\Users\pujar\OneDrive\လାଭକ\ai ass-6.3.py'
Name: Alice Smith
Roll Number: CS2026
Branch: Computer Science
PS C:\Users\pujar\OneDrive\လାଭକ> (C:\Users\pujar\anaconda3\shell\condabin\conda-hook.ps1) ; (conda activate base)
(base) PS C:\Users\pujar\OneDrive\လାଭକ>
```

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

#### Scenario

You are writing a utility function to display multiples of a given number.

#### Task

- Prompt the AI tool to generate a function that prints the first 10 multiples of a given number using a loop.
- Analyze the generated loop logic.
- Ask the AI to generate the same functionality using another controlled looping structure (e.g., `while` instead of `for`).

### Expected Output #2

- Correct loop-based Python implementation.
- Output showing the first 10 multiples of a number.
- Comparison and analysis of different looping approaches.

```
C:\Users\pujar> OneDrive > เอกสาร > ai > ...
1 def print_multiples_for(n):
2     for i in range(1, 11):
3         print(n * i)
4
5 def print_multiples_while(n):
6     i = 1
7     while i <= 10:
8         print(n * i)
9         i += 1
10
11 if __name__ == "__main__":
12     number = 7
13     print("For loop multiples:")
14     print_multiples_for(number)
15     print("While loop multiples:")
16     print_multiples_while(number)
17
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(base) PS C:\Users\pujar\OneDrive\เอกสาร> c:; cd 'c:\Users\pujar\OneDrive
:\Users\pujar\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bu
ujar\OneDrive\เอกสาร\ai'
7
14
21
28
35
42
49
56
63
70
(base) PS C:\Users\pujar\OneDrive\เอกสาร>
```

### Task Description #3: Conditional Statements (Age Classification)

#### Scenario

You are building a basic classification system based on age.

#### Task

- Ask the AI tool to generate nested if-elif-else conditional statements to classify age groups (e.g., child, teenager, adult, senior).
- Analyze the generated conditions and logic.
- Ask the AI to generate the same classification using alternative conditional structures (e.g.,

simplified conditions or dictionary-based logic).

#### Expected Output #3

- A Python function that classifies age into appropriate groups.
- Clear and correct conditional logic.
- Explanation of how the conditions work.

```
.. / Users / pujar / OneDrive / Lanans / ai...2.py > ...
1  def classify_age_nested(age):
2      if age < 0:
3          return "invalid"
4      if age <= 12:
5          return "child"
6      elif age <= 17:
7          return "teenager"
8      elif age <= 64:
9          return "adult"
10     else:
11         return "senior"
12
13 def classify_age_simplified(age):
14     if age < 0:
15         return "invalid"
16     if 0 <= age <= 12:
17         return "child"
18     if 13 <= age <= 17:
19         return "teenager"
20     if 18 <= age <= 64:
21         return "adult"
22     return "senior"
23
24 def classify_age_dict(age):
25     if age < 0:
26         return "invalid"
27     thresholds = [(12, "child"), (17, "teenager"), (64, "adult"), (float('inf'), "senior")]
28     for limit, label in thresholds:
29         if age <= limit:
30             return label
31
32 if __name__ == "__main__":
33     age = 25
34     print(classify_age_nested(age))
35     print(classify_age_simplified(age))
36     print(classify_age_dict(age))
37
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
(base) PS C:\Users\pujar\OneDrive\Лананс> c;; cd 'c:\Users\pujar\OneDrive\Лананс'; & 'C:\Users\pujar\anaconda3\pyt
debgpy-2025.18.0-win32-x64\bundled\libs\debgpy\launcher' '52165' --- 'c:\Users\pujar\OneDrive\Лананс\ai...2.py'
adult
adult
adult
(base) PS C:\Users\pujar\OneDrive\Лананс> c;; cd 'c:\Users\pujar\OneDrive\Лананс'; & 'C:\Users\pujar\anaconda3\pyt
ib\debgpy\launcher' '64788' --- 'c:\Users\pujar\OneDrive\Лананс\ai..2.py'
adult
adult
adult
(base) PS C:\Users\pujar\OneDrive\Лананс>
```

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

##### Scenario

You need to calculate the sum of the first n natural numbers.

##### Task

- Use AI assistance to generate a sum\_to\_n() function using a for loop.
- Analyze the generated code.
- Ask the AI to suggest an alternative implementation using a while loop or a mathematical formula.

#### Expected Output #4

- Python function to compute the sum of first n numbers.
  - Correct output for sample inputs.
  - Explanation and comparison of different approaches.

```
1 > Users > pujar > OneDrive > Lannan > ai...py > ...
1 def classify_age_nested(age):
2     if age < 0:
3         return "invalid"
4     if age <= 12:
5         return "child"
6     elif age <= 17:
7         return "teenager"
8     elif age <= 64:
9         return "adult"
10    else:
11        return "senior"
12
13
14 def classify_age_simplified(age):
15     if 18 <= age <= 64:
16         return "adult"
17     return "senior"
18
19
20 def classify_age_dict(age):
21     if age < 0:
22         return "invalid"
23     thresholds = [(12, "child"), (17, "teenager"), (64, "adult"), (float('inf'), "senior")]
24     for limit, label in thresholds:
25         if age <= limit:
26             return label
27
28
29 if __name__ == "__main__":
30     sample_ages = [3, 15, 30, 70, -1]
31
32     print("Nested if-elif-else:")
33     for a in sample_ages:
34         print(a, classify_age_nested(a))
35
36     print("Simplified chained conditions:")
37     for a in sample_ages:
38         print(a, classify_age_simplified(a))
39
40     print("Dictionary-threshold approach:")
41
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(base) PS C:\Users\pujar\OneDrive\Lannan> c;; cd 'c:\Users\pujar\OneDrive\Lannan'; & 'C:\Users\pujar\anaconda3\python.exe' 'debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52793' '--' 'c:\Users\pujar\OneDrive\Lannan\ai...3.py'
15 senior
30 adult
70 senior
-1 senior
Dictionary-threshold approach:
3 child
15 teenager
30 adult
70 senior
-1 invalid
(base) PS C:\Users\pujar\OneDrive\Lannan>
```

## **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.

```

:: > Users > pujar > OneDrive > ລັບສຳຮຽນ > ai...4.py > ...
1  def sum_to_n_for(n):
2      total = 0
3      for i in range(1, n + 1):
4          total += i
5      return total
6
7
8  def sum_to_n_while(n):
9      total = 0
10     i = 1
11     while i <= n:
12         total += i
13         i += 1
14     return total
15
16
17 def sum_to_n_formula(n):
18     if n < 0:
19         return None
20     return n * (n + 1) // 2
21
22
23 if __name__ == "__main__":
24     samples = [0, 1, 10, 100]
25     for n in samples:
26         print(n, sum_to_n_for(n), sum_to_n_while(n), sum_to_n_formula(n))
27

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```

(base) PS C:\Users\pujar\OneDrive\ລັບສຳຮຽນ> c:; cd 'c:\Users\pujar\OneDrive\ລັບສຳຮຽນ'; lib\debugpy\launcher' '52793' '--' 'c:\Users\pujar\OneDrive\ລັບສຳຮຽນ\ai...3.py'
3 child
15 teenager
30 adult
70 senior
-1 invalid
(base) PS C:\Users\pujar\OneDrive\ລັບສຳຮຽນ> c:; cd 'c:\Users\pujar\OneDrive\ລັບສຳຮຽນ'; -2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '63587' '--' 'c:\Users\pujar\OneDrive\ລັບສຳຮຽນ\ai...3.py'
0 0 0 0
1 1 1 1
10 55 55 55
100 5050 5050 5050
(base) PS C:\Users\pujar\OneDrive\ລັບສຳຮຽນ>

```

**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.