

NAME:CH.Kruthankiran

H.NO:2303A51404

BATCH:26

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			
Mr. G. Kranthi			
Course Code	23CS002PC304	Course Title	AI Assisted Coding
Year/Sem	III/I	Regulation	R23
Date and Day of Assignment	Week 5 - Thursday	Time(s)	23CSBTB01 To 23CSBTB52
Duration	2 Hours	Applicable to Batches	All Batches
AssignmentNumber: 10.4 (Present assignment number)/24(Total number of assignments)			
Q.No.	Question		Expected Time to complete
1	Lab 9 – Code Review and Quality: Using AI to Improve Code Quality and		Week 5

	<p>Readability</p> <p>Lab Objectives</p> <ul style="list-style-type: none"> • Use AI for automated code review and quality enhancement. • Identify and fix syntax, logical, performance, and security issues in Python code. • Improve readability and maintainability through structured refactoring and comments. • Apply prompt engineering for targeted improvements. • Evaluate AI-generated suggestions against PEP 8 standards and software engineering best practices 	
	<p>Task 1: AI-Assisted Syntax and Code Quality Review</p> <p>Scenario</p> <p>You join a development team and are asked to review a junior developer's Python script that fails to run correctly due to basic coding mistakes. Before deployment, the code must be corrected and standardized.</p> <p>Task Description</p> <p>You are given a Python script containing:</p> <ul style="list-style-type: none"> • Syntax errors • Indentation issues • Incorrect variable names • Faulty function calls <p>Use an AI tool (GitHub Copilot / Cursor AI) to:</p> <ul style="list-style-type: none"> • Identify all syntactic and structural errors • Correct them systematically • Generate an explanation of each fix made <p>Expected Outcome</p> <ul style="list-style-type: none"> • Fully corrected and executable Python code • AI-generated explanation describing: <ul style="list-style-type: none"> ◦ Syntax fixes ◦ Naming corrections ◦ Structural improvements • Clean, readable version of the script 	

```

AI ASSISTED PYTHON CODE REVIEW AND IMPROVEMENTS
# LUMINESCE USER
def calculate_average(numbers):
    if not numbers:
        return 0
    return sum(numbers) / len(numbers)

def print_student_info(name, age, grade):
    print("Student Information:\nName: {} \nAge: {} \nGrade: {}").format(name, age, grade)

def validate_email(email):
    valid = '@' in email
    print("Email is " + ("valid" if valid else "invalid"))

def calculate_discount(price, discount_percent):
    return price * (1 - discount_percent)

def get_user_input():
    return input("Enter your name: "), input("Enter your age: "), input("Enter your grade: ")

def process_data():
    names, ages = ["Alice", "Bob", "Charlie"], [25, 30, 35]
    for name, age in zip(names, ages):
        print("Processing {} , age {}".format(name, age))

def main():
    user_name, user_age, user_email = get_user_input()
    print_student_info(user_name, user_age, user_email)
    print("Average: (calculate_average([10, 20, 30, 40, 50]))")
    print("Final Price: (calculate_discount(100, 0.2))")
    process_data()

if __name__ == "__main__":
    main()

```

```

def calculate_average(numbers):
    if not numbers:
        return 0
    return sum(numbers) / len(numbers)

def print_student_info(name, age, grade):
    print("Student Information:\nName: {} \nAge: {} \nGrade: {}").format(name, age, grade)

def validate_email(email):
    valid = '@' in email
    print("Email is " + ("valid" if valid else "invalid"))
    return valid

def calculate_discount(price, discount_percent):
    return price * (1 - discount_percent)

def get_user_input():
    return input("Enter your name: "), input("Enter your age: "), input("Enter your grade: ")

def process_data():
    names, ages = ["Alice", "Bob", "Charlie"]
    for name, age in zip(names, ages):
        print("Processing {} , age {}".format(name, age))

def main():
    user_name, user_age, user_email = get_user_input()
    print_student_info(user_name, user_age, user_email)
    print("Average: (calculate_average([10, 20, 30, 40, 50]))")
    print("Final Price: (calculate_discount(100, 0.2))")
    process_data()

if __name__ == "__main__":
    main()

```

PS C:\Users\kruth\OneDrive\Desktop\Java> & 'c:\Users\kruth\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\kruth\OneDrive\Desktop\Java\AI 10.4.py'
Student Information:
Name: Kiran
Age: 20
Grade: Kirandigmail.com
Age: 30.0
Final Price: 80.0
Processing Alice, age 25
Processing Bob, age 30
Processing Charlie, age 35

```

AI 10.4()> ...
1 def find_duplicates_efficient(lst):
2     seen = set()
3     duplicates = set()
4
5     for item in lst:
6         if item in seen:
7             duplicates.add(item)
8         else:
9             seen.add(item)
10
11     return list(duplicates)
12
13
14 if __name__ == "__main__":
15     print(find_duplicates_efficient([1, 2, 3, 2, 3, 4, 1]))
16

```

← AI ASSISTED PYTHON CODE REVIEW AND IMPROVE...

```

Time Complexity: O(n)"""
indices_map = {}
for idx, num in enumerate(numbers):
    if num not in indices_map:
        indices_map[num] = []
    indices_map[num].append(idx)
return [num: idx for num, idxs in in
```

3. Performance Comparison Demo

```

# performance_comparison_demo.py
# Run this file to see live performance benchmarks
# Shows: Nested Loops ( $O(n^2)$ ) vs Set-Based
# Expected result: Optimized approaches 1
```

4. Quick Reference Guide

```

# QUICK_REFERENCE_DUPLICATES.py
# Contains:
# - Copy-paste ready solutions
# - Comparison table
# - Decision flowchart
# - Key takeaways
```

All files are in: java

Files created:

- inefficient_duplicate_finder.py - 3 inefficient $O(n^2)$ approaches
- optimized_duplicate_finder.py - 4 optimized $O(n)$ approaches with benchmarks
- performance_comparison_demo.py - Live performance tests

> 2 files changed +489 -2

+ AI 10.4(py)

Describe what to build next

Task 3: Readability and Maintainability Refactoring

Scenario

A working script exists in a project, but it is difficult to understand due to poor naming, formatting, and structure. The team wants it rewritten for long-term maintainability.

Task Description

You are given a poorly structured Python function with:

- Cryptic function names
- Poor indentation
- Unclear variable naming
- No documentation

Use AI-assisted review to:

- Refactor the code for clarity
- Apply PEP 8 formatting standards
- Improve naming conventions
- Add meaningful documentation

Expected Outcome

- Clean, well-structured code
- Descriptive function and variable names
- Proper indentation and formatting
- Docstrings explaining the function purpose
- AI explanation of readability improvements

```

● AI 1040.py 2...
1 def find_duplicate_values(input_list):
2     seen = set()
3     duplicates = set()
4
5     for item in input_list:
6         if item in seen:
7             duplicates.add(item)
8         else:
9             seen.add(item)
10
11    return list(duplicates)
12
13
14 if __name__ == "__main__":
15     test_list = [1, 2, 3, 2, 4, 1]
16     result = find_duplicate_values(test_list)
17     print(result)
18

```

PROBLEMS 76 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```

PS C:\Users\kruth\OneDrive\Desktop\Java> & 'c:\Users\kruth\AppData\Local\Microsoft\WindowsApps\python3.11.exe' `c:\Users\kruth\VSCode\extensions\ms-python.python.debug-2025.18.0-win32-x64\build\libs\debug\py_launcher` '61261' '--' `c:\Users\kruth\OneDrive\Desktop\Java> & 'c:\Users\kruth\VSCode\extensions\ms-python.python.debug-2025.18.0-win32-x64\build\libs\debug\py_launcher` '62466' '--' `c:\Users\kruth\VSCode\extensions\ms-python.python.debug-2025.18.0-win32-x64\build\libs\debug\py_launcher` '53242' '--' `c:\Users\kruth\OneDrive\Desktop\Java\AI 10.4(1).py` [1, 2]
PS C:\Users\kruth\OneDrive\Desktop\Java> & cd `c:\Users\kruth\OneDrive\Desktop\Java` & 'c:\Users\kruth\AppData\Local\Microsoft\WindowsApps\python3.11.exe' `c:\Users\kruth\VSCode\extensions\ms-python.python.debug-2025.18.0-win32-x64\build\libs\debug\py_launcher` '53242' '--' `c:\Users\kruth\OneDrive\Desktop\Java\AI 10.4(1).py` [1, 2]
PS C:\Users\kruth\OneDrive\Desktop\Java>

```

← AI ASSISTED PYTHON CODE REVIEW AND IMPROVEMENT

You are provided with a function that identifies duplicate values in a list using inefficient nested loops.

Using AI-assisted Code Review:

- Analyze the logic for performance bottlenecks
- Refactor the code for better time complexity
- Preserve the correctness of the output

Expected Outcome

- Optimized duplicate-detection logic using sets or hash-based structures
- Improved time complexity ($O(n^2) \rightarrow O(n)$)
- AI explanation of performance improvement
- Clear, readable implementation

CODE - TASK 2

INEFFICIENT CODE (inefficient_duplicate_finder.py):

```

# INEFFICIENT VERSION - Duplicate Detection
# Time Complexity: O(n^2) - Very slow for large lists

def find_duplicates_inefficient(numbers):
    """Find duplicate values using nested loops"""

    Time Complexity: O(n^2) - Quadratic time
    Space Complexity: O(1) - No extra space required

    duplicates = []

    for i in range(len(numbers)):
        for j in range(i + 1, len(numbers)):
            if numbers[i] == numbers[j]:
                duplicates.append(numbers[i])

    return duplicates

```

Time Complexity: $O(n^2)$ - Quadratic time
Space Complexity: $O(1)$ - No extra space required

...
duplicates = []

Outer loop iterates through each element
for i in range(len(numbers)):
 # Inner loop compares with all other elements
 for j in range(i + 1, len(numbers)):
 if match found and not already added
 duplicates.append(numbers[i])

> 4 files changed +1001 -4

Describe what to do next

undo redo

Task 4: Secure Coding and Reliability Review

Scenario

A backend function retrieves user data from a database but has security vulnerabilities and poor error handling, making it unsafe for production deployment.

Task Description

You are given a Python script that:

- Uses unsafe SQL query construction
- Has no input validation
- Lacks exception handling

Use AI tools to:

- Identify security vulnerabilities
- Refactor the code using safe coding practices
- Add proper exception handling
- Improve robustness and reliability

Expected Outcome

- Secure SQL queries using parameterized statements
- Input validation logic
- Try-except blocks for runtime safety
- AI-generated explanation of security improvements
- Production-ready code structure

Task 5: AI-Based Automated Code Review Report

Scenario

Your team uses AI tools to perform automated preliminary code reviews before human review, to improve code quality and consistency across projects.

Task Description

| You are provided with a poorly written Python script.

Using AI-assisted review:

- Generate a **structured code review report** that evaluates:
 - Code readability
 - Naming conventions
 - Formatting and style consistency
 - Error handling
 - Documentation quality
 - Maintainability

The task is not just to fix the code, but to **analyze and report on quality issues**.

Expected Outcome

- AI-generated review report including:
 - Identified quality issues
 - Risk areas
 - Code smell detection
 - Improvement suggestions
 - Optional improved version of the code
 - Demonstration of AI as a **code reviewer**, not just a code generator

The screenshot shows a code editor window with a Python script named `AI 1049.py`. The code is for a user management system. The interface includes a sidebar for AI-assisted code review, a terminal output showing command-line interactions, and a bottom status bar.

```

# AI 1049.py
1 import sqlite3
2 def getdata(id):
3     try:
4         conn = sqlite3.connect("db.db")
5         cur = conn.cursor()
6         cur.execute("SELECT * FROM users WHERE id = ?", (id,))
7         res = cur.fetchone()
8         return res
9     except sqlite3.Error as e:
10        print("Database error: (e)")
11        return None
12
13    finally:
14        if 'conn' in locals():
15            conn.close()
16
17 if __name__ == "__main__":
18     print(getdata(1))
19
20
21
22
23

```

AI ASSISTED PYTHON CODE REVIEW AND IMPROVE...

- Code smell detection
- Improvement suggestions
 - Optional improved version of the code
 - Demonstration of AI as a code reviewer, not just a code generator
- give code

✓ Created 4 files
✓ Summarized conversation history
give code
Reviewed and updated corrected_code_review.py and checked lines 1-521
Based on the workspace, here's the improved code for Task 5 with all issues addressed:

```

# IMPROVED CODE - Based on AI Code Review
# Task 5: Refactored Version with AI Integration

---User management system for tracking users---

This module provides functionality for:
- Managing users (name, email, age)
- Fetching user data from remote APIs
- Processing file data
- Performing arithmetic calculations

Classes:
  UserManager: Manages collection of users

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

+ v ... | ☰ ×

softWindowsApps\python3.11.exe' 'c:\Users\kruthi\.vscode\extensions\ms-python.debugger-2025.18.0-win32-x64\bundled\libs\debug\java\launcher' '532d' -- '- ' c:\Users\kruthi\OneDrive\Desktop\java\AI 10.4(iii).py'
● PS C:\Users\kruthi\OneDrive\Desktop\java> cd 'c:\Users\kruthi\OneDrive\Desktop\java'; & 'c:\Users\kruthi\AppData\Local\Microsoft\softWindowsApps\python3.11.exe' 'c:\Users\kruthi\.vscode\extensions\ms-python.debugger-2025.18.0-win32-x64\bundled\libs\debug\java\launcher' '6448' -- '- ' c:\Users\kruthi\OneDrive\Desktop\java\AI 10.4(iii).py'
Database error: no such table: users
None
● PS C:\Users\kruthi\OneDrive\Desktop\java> cd 'c:\Users\kruthi\OneDrive\Desktop\java'; & 'c:\Users\kruthi\AppData\Local\Microsoft\softWindowsApps\python3.11.exe' 'c:\Users\kruthi\.vscode\extensions\ms-python.debugger-2025.18.0-win32-x64\bundled\libs\debug\java\launcher' '5489' -- '- ' c:\Users\kruthi\OneDrive\Desktop\java\AI 10.4(iv).py'
Database error: no such table: users
None

powerhell... △

powerhell

python Debug

Describe what to build next

Auto