|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | | 1. Dr. Mohammed Ali Shaik  2. Dr. T Sampath Kumar  3. Mr. S Naresh Kumar  4. Dr. V. Rajesh  5. Dr. Brij Kishore  6. Dr Pramoda Patro  7. Dr. Venkataramana  8. Dr. Ravi Chander  9. Dr. Jagjeeth Singh | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week2-Tuesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | | 24CSBTB01 To 24CSBTB39 | | | |
| **AssignmentNumber:3.2**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 3: Prompt Engineering – Improving Prompts and Context Management  **Lab Objectives:**   * To understand how prompt structure and wording influence AI-generated code. * To explore how context (like comments and function names) helps AI generate relevant output. * To evaluate the quality and accuracy of code based on prompt clarity. * To develop effective prompting strategies for AI-assisted programming.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Generate Python code using Google Gemini in Google Colab. * Analyze the effectiveness of code explanations and suggestions by Gemini. * Set up and use Cursor AI for AI-powered coding assistance. * Evaluate and refactor code using Cursor AI features. * Compare AI tool behavior and code quality across different platforms.   **Task Description#1**   * Ask AI to write a function to calculate compound interest, starting with only the function name. Then add a docstring, then input-output example   **Expected Output#1**   * Comparison of AI-generated code styles   **Task Description#2**   * Do math stuff, then refine it to: # Write a function to calculate average, median, and mode of a list of numbers.   **Expected Output#2**   * AI-generated function evolves from unclear to accurate multi-statistical operation.   **Task Description#3**   * Provide multiple examples of input-output to the AI for convert\_to\_binary(num) function. Observe how AI uses few-shot prompting to generalize.   **Expected Output#3**   * Enhanced AI output with clearer prompts   **Task Description#4**   * Create an user interface for an hotel to generate bill based on customer requirements   **Expected Output#4**   * Consistent functions with shared logic   **Task Description#5**   * Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions   **Expected Output#5**   * Code quality difference analysis for various prompts   **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** | | --- | --- | | Task#1 | 0.5 | | Task#2 | 0.5 | | Task #3 | 0.5 | | Task #4 | 0.5 | | Task #5 | 0.5 | | **Total** | **2.5 Marks** | | | | | | | 03.08.2025 EOD |  |

**#Task 1:**

**#prompt:**

write a program in python, a function to calculate compound interest.

**#code & output:**

A screenshot of a computer

Description automatically generated

**#Explanation:**

The code defines a function called compound\_interest that takes four inputs:

* principal: The initial amount of money you start with.
* rate: The annual interest rate (how much extra money you earn on your investment each year).
* time: The number of years you leave the money invested.
* Compounding\_frequency: How many times a year the interest is calculated and added to your principal.

The function uses a formula to calculate the total amount of money you will have after the specified time, including the compound interest. It then returns this final amount.

The example usage shows how to call the function with specific values (1000 principal, 5% rate, 10 years, compounded quarterly) and prints the calculated final amount.

**#Task 2:**

**#Prompt:**

write a program in python to find average,median,mode of a list of numbers using functions based on user input.

**#Code:**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**#Output:**

A screenshot of a computer

Description automatically generated

**#Explanation:**

This code is designed to calculate the average, median, and mode of a list of numbers that you provide. Here's how it works:

1. **Getting the Numbers (get\_numbers\_from\_user function):**
   * This part asks you to enter numbers one by one.
   * When you are finished entering numbers, you type the word 'done' and press Enter.
2. **Calculating the Average (calculate\_average function):**
   * This part takes the list of numbers you entered.
   * It adds up all the numbers in the list.
   * Then, it divides the total sum by how many numbers are in the list to find the average.
3. **Calculating the Median (calculate\_median function):**
   * This part also takes the list of numbers.
   * It sorts the numbers from smallest to largest.
   * It then finds the middle number(s) in the sorted list.
   * If there's an odd number of items, the middle one is the median.
   * If there's an even number of items, the median is the average of the two numbers in the middle.
4. **Calculating the Mode (calculate\_mode function):**
   * This part takes the list of numbers and finds the number(s) that appear most often.
   * The number(s) with the highest count are the mode(s).
   * If all numbers appear only once, it indicates there's no single number that appears more than others.
5. **Putting it all together (Main execution block):**
   * This is the part that runs the program.
   * It first calls the get\_numbers\_from\_user function to get your list of numbers.
   * If you entered any numbers, it then calls the calculate\_average, calculate\_median, and calculate\_mode functions using your list.
   * Finally, it prints out the list of numbers you entered and the calculated average, median, and mode.

**#Task 3:**

**#Prompt:**  
write a python program to convert any number into binary using convert\_to\_binary(num) function based on user input.

**#Code & Output:**

A screenshot of a computer program

Description automatically generated

**#Explanation:**

This program does the following:

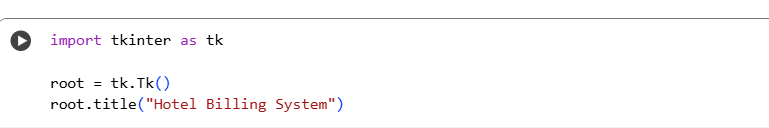
1. **Defines a function convert\_to\_binary(num):**
   * This function is designed to take a whole number (an integer) as input.
   * It figures out the binary (base-2) representation of that number. Binary uses only 0s and 1s.
   * If the input number is 0, it simply returns "0".
   * For numbers greater than 0, it repeatedly divides the number by 2 and records the remainder (which will be either 0 or 1). These remainders, read in reverse order, form the binary representation.
   * It returns the binary representation as a string of 0s and 1s.
2. **Gets user input:**
   * The program then prompts you to "Enter a decimal number to convert to binary: ".
   * It waits for you to type a number and press Enter.
   * It tries to convert your input into an integer.
3. **Converts and Prints:**
   * If you entered a valid integer, it calls the convert\_to\_binary function with your number to get its binary form.
   * Finally, it prints a message showing your original number and its binary representation.
4. **Handles Errors:**
   * If you enter something that is not a whole number (like text or a decimal with a point), the program catches this error and prints "Invalid input. Please enter an integer."

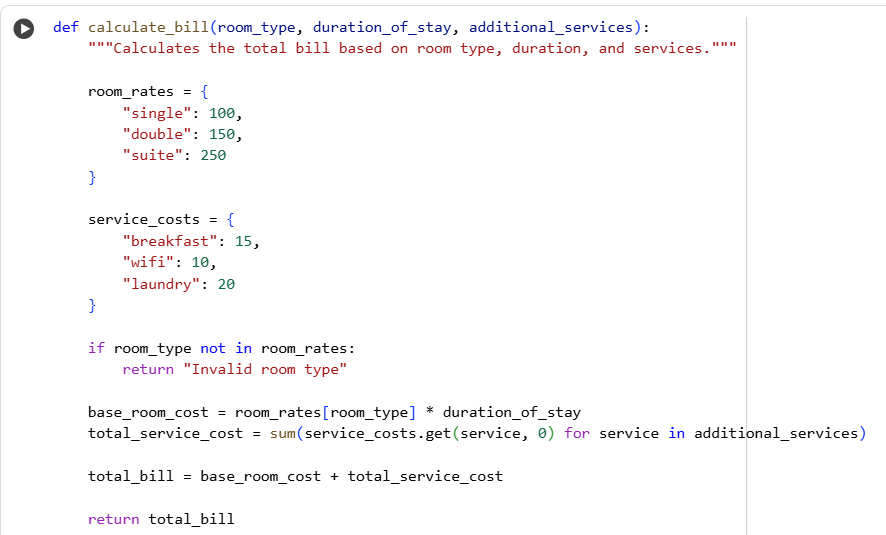
**#Task 4:**

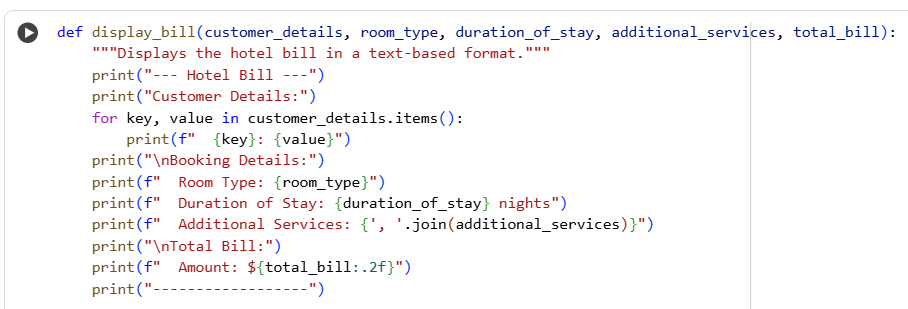
**Prompt:**

write a program in python that creates an user interface for an hotel to generate bill based on customer requirements.

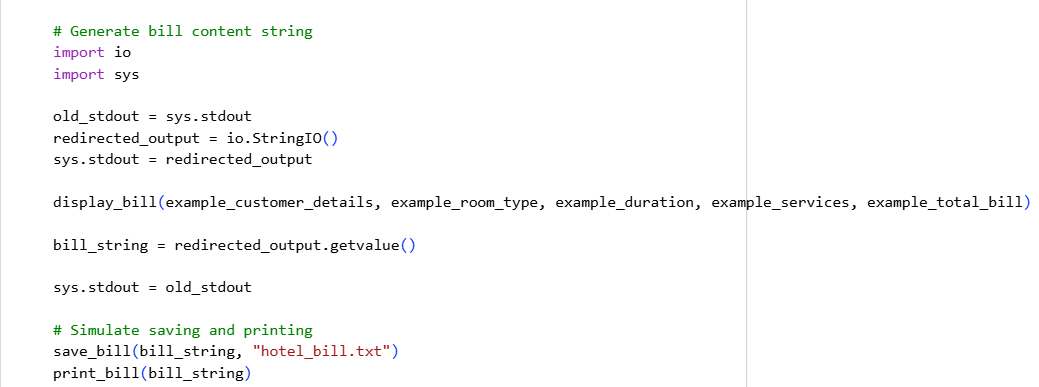
**Code:**

****

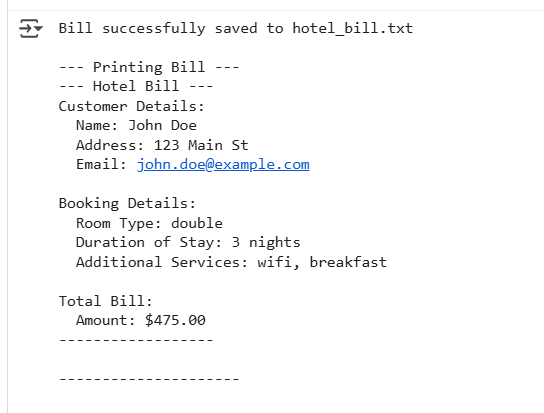
****







**Output:**



**Explanation:**

* The initial attempts to create a graphical user interface using tkinter failed due to the execution environment lacking a graphical display.
* A Python function calculate\_bill was successfully implemented to calculate the total bill based on room type, duration of stay, and additional services, incorporating predefined rates for different options.
* A text-based Python function display\_bill was created to present the bill details, including customer information, booking details, and the total amount, in a formatted manner to the console.
* Functions save\_bill and print\_bill were successfully implemented to save the generated bill content to a text file and print it to the console, respectively, demonstrating the ability to handle bill output.

**#Task 5:**

**#Prompt:**

write a program in python using function named convert\_temp ,that conversion happens between celsius, fahreinheit, kelvin based on user input.

**#Code:**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

**#Output:**

A screenshot of a computer

Description automatically generated

**#Explanation:**

This Python code defines several functions to convert temperatures between Celsius, Fahrenheit, and Kelvin.

* **celsius\_to\_fahrenheit(celsius):** Takes a temperature in Celsius and converts it to Fahrenheit.
* **fahrenheit\_to\_celsius(fahrenheit):** Takes a temperature in Fahrenheit and converts it to Celsius.
* **celsius\_to\_kelvin(celsius):** Takes a temperature in Celsius and converts it to Kelvin.
* **kelvin\_to\_celsius(kelvin):** Takes a temperature in Kelvin and converts it to Celsius.
* **fahrenheit\_to\_kelvin(fahrenheit):** Converts Fahrenheit to Kelvin by first converting to Celsius, then to Kelvin.
* **kelvin\_to\_fahrenheit(kelvin):** Converts Kelvin to Fahrenheit by first converting to Celsius, then to Fahrenheit.
* **convert\_temp():** This is the main function that interacts with the user. It asks for the temperature value, the original unit (C, F, or K), and the target unit (C, F, or K). Based on the user's input, it calls the appropriate conversion function and prints the result.

Finally, the line convert\_temp() at the end runs the main function to start the temperature conversion process.