

## Ai Assisted Coding

Name : Sai Varsha

Roll NO : 2303A54065

### Lab 9.1: Documentation Generation -Automatic documentation and code comments

#### Problem 1:

Consider the following Python function:

```
def find_max(numbers):  
    return max(numbers)
```

#### Task:

- Write documentation for the function in all three formats:
  - (a) Docstring
  - (b) Inline comments
  - (c) Google-style documentation
- Critically compare the three approaches. Discuss the advantages, disadvantages, and suitable use cases of each style.
- Recommend which documentation style is most effective for a mathematical utilities library and justify your Answer.

#### Prompts Used :

Python3 –m pydoc task01

Python3 –m pydoc –w task01

Open task01.html

#### Expected output :

The screenshot shows a web browser window with the title "Python: module task01". The URL in the address bar is "/Users/anumandarshika/SRUI/AI\_Assisted\_Job/lab9/task01.html". The page content is titled "task01" and describes a module problem related to the find\_max function. It demonstrates different documentation styles. A section titled "Functions" contains three definitions:

```
Module: Problem 1 - find_max function
Demonstrates different documentation styles.

Functions
find_max_basic(numbers)
    Returns the maximum value in a list.

find_max_google(numbers)
    Finds the maximum value in a list of numbers.

Args:
    numbers (list): List of numeric values.

Returns:
    Int or float: Maximum number in list.

Raises:
    ValueError: If list is empty.

find_max_inline(numbers)
```

## Problem 2:

Consider the following Python function:

```
def login(user, password, credentials):
    return credentials.get(user) == password
```

### Task:

1. Write documentation in all three formats.
2. Critically compare the approaches.
3. Recommend which style would be most helpful for new developers onboarding a project, and justify your choice.

### Prompts Used :

```
Python3 –m pydoc task02
```

```
Python –m pydoc –w task02
```

```
Open task02.html
```

### Expected Output :

The screenshot shows a web browser window with three tabs open. The active tab is titled 'Python: module task02' and displays the contents of a Python module named 'task02'. The page has a blue header bar with the title 'task02' and a sub-header 'Problem 2 - Login Function Documentation Styles'. Below this, there's a section titled 'Functions' with three defined functions:

```
login_basic(user, password, credentials)
    Checks if login credentials are valid.

login_google(user, password, credentials)
    Authenticates a user using provided credentials.

Args:
    user (str): Username.
    password (str): Password entered.
    credentials (dict): Dictionary of username-password pairs.

Returns:
    bool: True if credentials match, else False.

login_inline(user, password, credentials)
    # @ INLINE COMMENTS
```

### Problem 3: Calculator (Automatic Documentation Generation)

**Task:** Design a Python module named calculator.py and demonstrate automatic documentation generation.

Instructions:

1. Create a Python module calculator.py that includes the following functions, each written with appropriate docstrings:
  - o add(a, b) – returns the sum of two numbers
  - o subtract(a, b) – returns the difference of two numbers
  - o multiply(a, b) – returns the product of two numbers
  - o divide(a, b) – returns the quotient of two numbers
2. Display the module documentation in the terminal using Python's documentation tools.
3. Generate and export the module documentation in HTML format using the pydoc utility, and open the generated HTML

file in a web browser to verify the output.

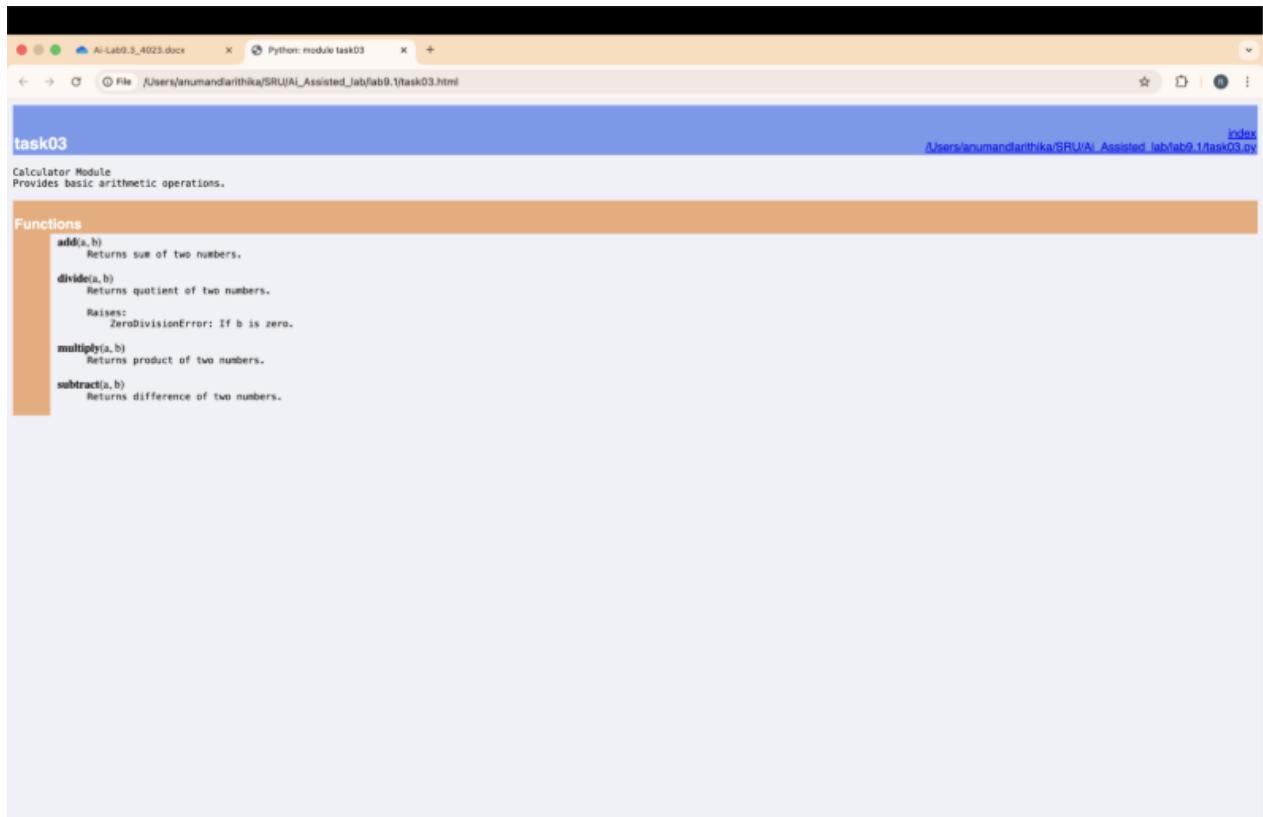
### Prompts Used :

Python3 –m pydoc task03

Python –m pydoc –w task03

Open task03.html

### Expected Output :



The screenshot shows a web browser window with the title "Python: module task03". The address bar indicates the file is located at "/Users/anumandarithika/SRUJ/AI\_Assisted\_Lab/lab9.1/task03.html". The page content is titled "task03" and describes the "Calculator Module" as "Provides basic arithmetic operations". Below this, there is a "Functions" section containing the following code:

```
Calculator Module
Provides basic arithmetic operations.

Functions
add(a, b)
    Returns sum of two numbers.

divide(a, b)
    Returns quotient of two numbers.

    Raises:
        ZeroDivisionError: If b is zero.

multiply(a, b)
    Returns product of two numbers.

subtract(a, b)
    Returns difference of two numbers.
```

## Problem 4: Conversion Utilities Module

### Task:

1. Write a module named conversion.py with functions:

- o decimal\_to\_binary(n)
- o binary\_to\_decimal(b)
- o decimal\_to\_hexadecimal(n)

2. Use Copilot for auto-generating docstrings.

3. Generate documentation in the terminal.

4. Export the documentation in HTML format and open it in a Browser.

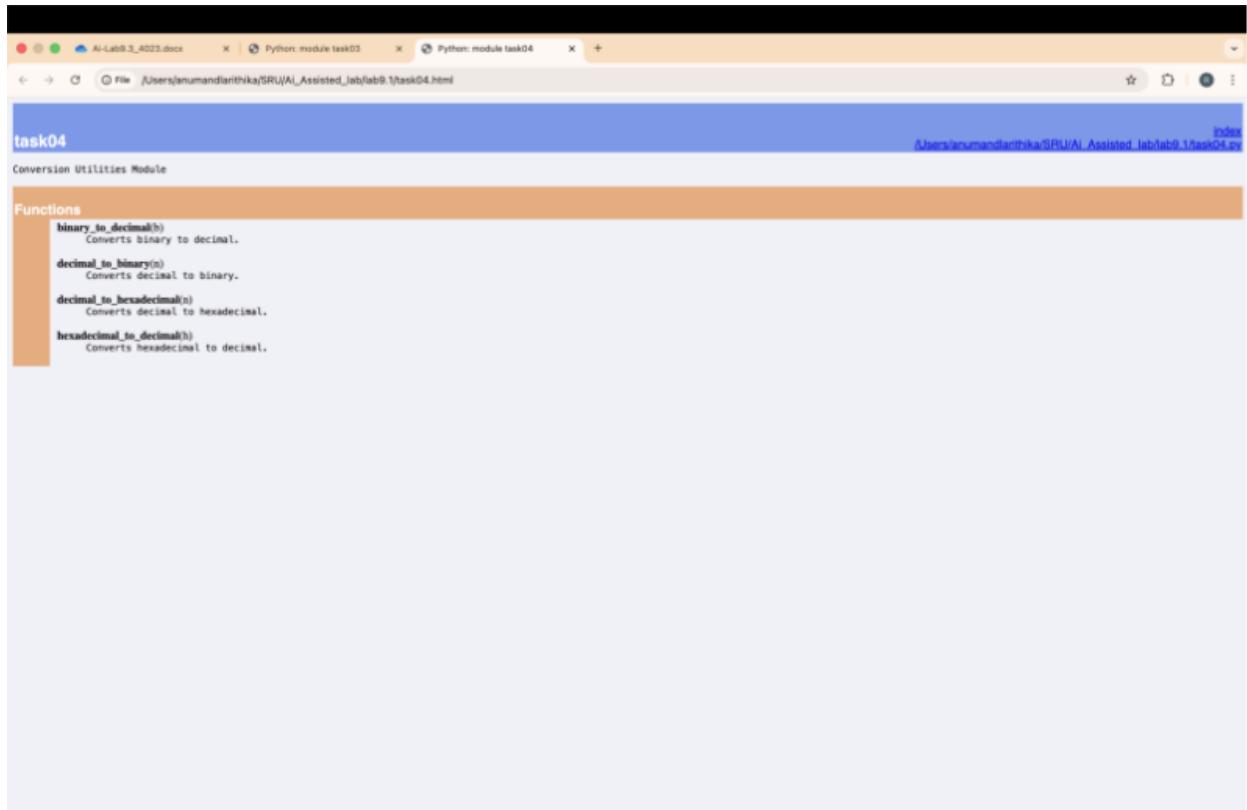
### Prompts Used :

Python3 –m pydoc task04

Python –m pydoc –w task04

Open task04.html

### Expected Output :



## Problem 5 – Course Management Module

### Task:

1. Create a module course.py with functions:

- o `add_course(course_id, name, credits)`
- o `remove_course(course_id)`
- o `get_course(course_id)`

2. Add docstrings with Copilot.

3. Generate documentation in the terminal.
4. Export the documentation in HTML format and open it in a Browser.

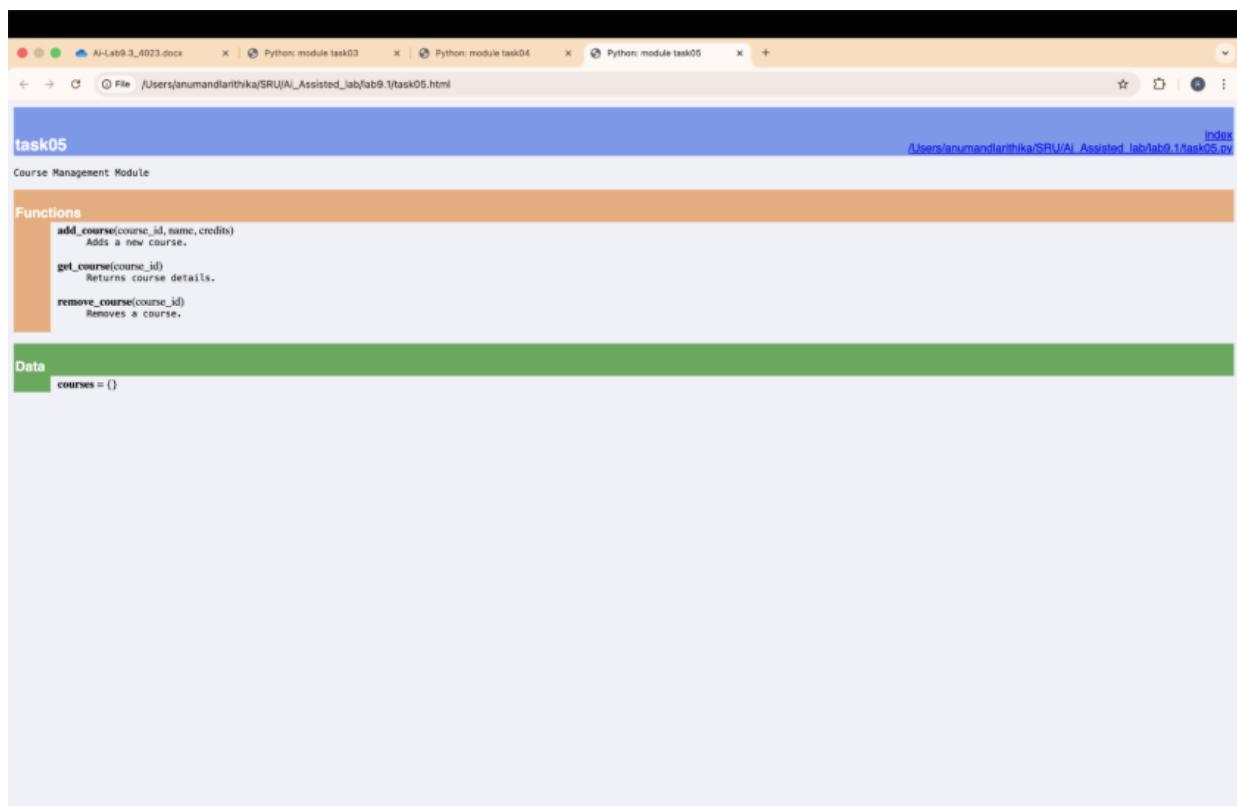
### Prompts Used :

Python3 –m pydoc task05

Python –m pydoc –w task05

Open task05.html

### Expected Output :



The screenshot shows a web browser window with four tabs open. The active tab is titled "task05" and displays the generated Python documentation. The page has a blue header bar with the title "task05" and a sub-header "Course Management Module". Below this is a yellow sidebar containing the word "Functions" and three code snippets:

```
add_course(course_id, name, credits)
    Adds a new course.

get_course(course_id)
    Returns course details.

remove_course(course_id)
    Removes a course.
```

Below the sidebar is a green header bar labeled "Data" and containing the variable definition "courses = {}". The main content area of the page is currently empty, showing only the header and sidebar.