

AI

NAME:-P.LIKITHA

HT NO:-2303A52393

BATCH-43

# (a) Function with Standard Python Docstring

```
def find_max_docstring(numbers):
    """
    Returns the maximum value from a list of numbers.

    Parameters:
        numbers (list): A list of numeric values.

    Returns:
        int or float: The largest number in the list.

    Raises:
        ValueError: If the list is empty.
    """
    return max(numbers)
```

# (b) Function with Inline Comments

```
def find_max_inline(numbers):
    # Take a list of numbers as input
    # Use Python's built-in max() function to find largest value
    # Return the maximum number from the list
    return max(numbers)
```

# (c) Function with Google-Style Documentation

```
def find_max_google(numbers):
    """Finds the maximum value in a list of numbers.

    Args:
        numbers (list): A list containing numeric values.

    Returns:
```

```
"""Finds the maximum value in a list of numbers.

Args:
    numbers (list): A list containing numeric values.

Returns:
    int | float: The largest number in the list.

Raises:
    ValueError: If the input list is empty.
"""
return max(numbers)

# Example usage
data = [10, 25, 7, 42]

print("Docstring version:", find_max_docstring(data))
print("Inline comment version:", find_max_inline(data))
print("Google style version:", find_max_google(data))
```

... Docstring version: 42  
Inline comment version: 42  
Google style version: 42

```
def find_max(numbers):
    """
    Finds the maximum value from a list of numbers.

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

    Returns:
```

Alipynb

File Edit View Insert Runtime Tools Help

Share P

RAM Disk

```
[ ] finds the maximum value from a list of numbers.

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

Returns:
    int/float: The maximum value in the list.
"""

# Using Python built-in max() function
return max(numbers)

# Inline comment version explanation:
# This function takes a list and returns the largest number

# Example usage
nums = [10, 45, 3, 99, 23]
print("Maximum number:", find_max(nums))

Maximum number: 99
```

Alipynb

File Edit View Insert Runtime Tools Help

Share P

RAM Disk

```
[ ] def login(user, password, credentials):
    """
    Validates user login credentials.

    Args:
        user (str): Username entered by user
        password (str): Password entered by user
        credentials (dict): Dictionary of valid usernames and passwords

    Returns:
        bool: True if credentials are valid, False otherwise
    """

    # Check if password matches stored password
    return credentials.get(user) == password

# Example usage
users = {
    "admin": "admin123",
    "student": "stud456"
}

print("Login result:", login("admin", "admin123", users))

Login result: True
```

```
Calculator Module
Provides basic arithmetic operations.

def add(a, b):
    """
    Returns the sum of two numbers.
    """
    return a + b

def subtract(a, b):
    """
    Returns the difference of two numbers.
    """
    return a - b

def multiply(a, b):
    """
    Returns the product of two numbers.
    """
    return a * b

def divide(a, b):
    """
    Returns the quotient of two numbers.
    """
```

```
def multiply(a, b):
    """
    Returns the product of two numbers.
    """
    return a * b

def divide(a, b):
    """
    Returns the quotient of two numbers.

    Raises:
        ValueError: If division by zero is attempted
    """
    if b == 0:
        raise ValueError("Cannot divide by zero")
    return a / b

# Example usage
print("Add:", add(5, 3))
print("Subtract:", subtract(5, 3))
print("Multiply:", multiply(5, 3))
print("Divide:", divide(10, 2))
```

Add: 8  
Subtract: 2  
Multiply: 15  
Divide: 5.0

A screenshot of a Jupyter Notebook interface titled "AI.ipynb". The code cell contains the following Python code:

```
"""
Conversion Utilities Module
Provides number system conversion functions.

"""

def decimal_to_binary(n):
    """
    Converts a decimal number to binary.

    Args:
        n (int): Decimal number

    Returns:
        str: Binary representation

    """
    return bin(n)[2:]

def binary_to_decimal(b):
    """
    Converts a binary number to decimal.

    Args:
        b (str): Binary number as string

    Returns:
        int: Decimal value

    """
    return int(b, 2)
```

The interface includes standard Jupyter Notebook navigation buttons (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with Share, RAM/Disk status, and a search bar.

A screenshot of a Jupyter Notebook interface titled "AI.ipynb". The code cell contains the following Python code:

```
    int: Decimal value
    return int(b, 2)

def decimal_to_hexadecimal(n):
    """
    Converts a decimal number to hexadecimal.

    Args:
        n (int): Decimal number

    Returns:
        str: Hexadecimal representation

    """
    return hex(n)[2:]

# Example usage
print("Binary:", decimal_to_binary(10))
print("Decimal:", binary_to_decimal("1010"))
print("Hexadecimal:", decimal_to_hexadecimal(255))
```

The interface shows the output of the print statements in the code cell:

```
Binary: 1010
Decimal: 10
Hexadecimal: ff
```

AI.ipynb

File Edit View Insert Runtime Tools Help

RAM Disk

```
[ ]
```

```
Course Management Module
Handles course-related operations.

courses = {}

def add_course(course_id, name, credits):
    """
    Adds a course to the course list.

    Args:
        course_id (str): Unique course ID
        name (str): Course name
        credits (int): Number of credits
    """
    courses[course_id] = {
        "name": name,
        "credits": credits
    }

def remove_course(course_id):
    """
    Removes a course using course ID.

    Args:
        course_id (str): Course ID
    """
    return courses.pop(course_id, None)

def get_course(course_id):
    """
    Retrieves course details.

    Args:
        course_id (str): Course ID
    Returns:
        dict or None: Course information
    """
    return courses.get(course_id)

# Example usage
add_course("CS101", "Python Programming", 4)
add_course("CS102", "Data Structures", 3)

print("Course CS101:", get_course("CS101"))
remove_course("CS102")
print("All Courses:", courses)
```

AI.ipynb

File Edit View Insert Runtime Tools Help

RAM Disk

```
[ ]
```

```
return courses.pop(course_id, None)

def get_course(course_id):
    """
    Retrieves course details.

    Args:
        course_id (str): Course ID
    Returns:
        dict or None: Course information
    """
    return courses.get(course_id)

# Example usage
add_course("CS101", "Python Programming", 4)
add_course("CS102", "Data Structures", 3)

print("Course CS101:", get_course("CS101"))
remove_course("CS102")
print("All Courses:", courses)
```

```
Course CS101: {'name': 'Python Programming', 'credits': 4}
All Courses: {'CS101': {'name': 'Python Programming', 'credits': 4}}
```