

## **Assignment -9.1**

**Hallticket no:2303A510A6**

**Batch no:02**

**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.

**Code:**

```

C: > Users > shyam > ai assistes code > ass16ai.py
 1 def find_max(numbers):
 2     """
 3         Returns the maximum value from a list of numbers.
 4
 5     Parameters:
 6         numbers (list): A list of numerical values.
 7
 8     Returns:
 9         int or float: The largest value in the list.
10
11    Raises:
12        ValueError: If the list is empty."""
13 def find_max(numbers):
14     # Check if the list is empty
15     if not numbers:
16         # Raise an error if empty
17         raise ValueError("The input list cannot be empty.")
18
19     # Return the maximum value using Python's built-in max()
20     return max(numbers)
21
22 def find_max(numbers):
23     """Returns the maximum value from a list of numbers.
24
25     Args:
26         numbers (list): A list of numerical values.
27
28     Returns:
29         int or float: The largest value in the list.
30
31     Raises:
32         ValueError: If the list is empty.
33     """
34     if not numbers:
35         raise ValueError("The list cannot be empty.")
36     return max(numbers)
37

```

## 1) Docstring

### **Advantages:**

- Explains what the function does.
- Can be viewed using `help()` in Python.
- Easy to write and understand.

### **Disadvantages:**

- Not always structured.
- Can be inconsistent if different people write it differently.

## 2) Inline Comments

### **Advantages:**

- Explain how the code works step by step.
- Helpful for understanding complex logic.
- Disadvantages:

- Cannot be accessed using documentation tools.
- Can make code messy if overused.

### **3) Google-Style Documentation**

#### **Advantages:**

- Clear and well-structured format.
- Separates Args, Returns, and Raises.
- Good for large projects.

#### **Disadvantages:**

- Slightly longer to write.
- May be unnecessary for very small functions.

or a mathematical utilities library, **Google-style documentation is the most effective** because it clearly explains inputs, outputs, and errors in a structured way. Mathematical functions require precision, so a clear and consistent format is important.

**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.**

**Code:**

**For new developers onboarding a project, Google-style documentation is the most helpful.**

```
C:\> Users > shyam > ai assistes code > ass16ai.py
 1 def login(user, password, credentials)
 2 """
 3     this function checks if the user and password are correct by comparing them to the credentials dictionary.
 4     It returns True if the login is successful and False otherwise."""
 5 if user in credentials and credentials[user] == password:
 6     return True
 7 else:
 8     return False
 9
10
11     def login (user, password, credentials):
12         # check if the user and password are correct by comparing them to the credentials dictionary.
13         # It returns True if the login is successful and False otherwise.
14         if user in credentials and credentials[user] == password:
15             return True
16         else:
17             return False
18
19 def login(user, password, credentials):
20 """
21     Check if the user and password are correct by comparing them to the credentials dictionary.
22
23     Parameters
24     -----
25     user : str
26         The username to check.
27     password : str
28         The password to check.
29     credentials : dict
30         A dictionary where keys are usernames and values are passwords.
31
32     Returns
33     -----
34     bool
35         True if the login is successful, False otherwise.
36
37     Examples
38     -----
39     >>> credentials = {'user1': 'pass1', 'user2': 'pass2'}
40     >>> login('user1', 'pass1', credentials)
41     True
42     >>> login('user2', 'wrongpass', credentials)
43     False
44     >>> login('nonexistent', 'pass', credentials)
45     False"""
46     if user in credentials and credentials[user] == password:
47         return True
48     else:
49         return False
50
```

For new developers onboarding a project, Google-style documentation is the most helpful.

### **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.

Code:

```
> Users > shyam > ai assistes code > ass16ai.html
1  """
2  calculator.py
3  A simple calculator module that provides basic arithmetic operations:
4  addition, subtraction, multiplication, and division."""
5
6  def add(a, b):
7      """Return the sum of two numbers."""
8      return a + b
9
10
11 def subtract(a, b):
12     """Return the difference of two numbers."""
13     return a - b
14
15
16 def multiply(a, b):
17     """Return the product of two numbers."""
18     return a * b
19
20
21 def divide(a, b):
22     """
23     Return the quotient of two numbers.
24
25     Raises:
26         ZeroDivisionError: If b is zero.
27     """
28     if b == 0:
29         raise ZeroDivisionError("Cannot divide by zero.")
30     return a / b
```

Display the module documentation in the terminal using

Python's documentation tools.

```
PS C:\Users\shyam\ai assistes code> python -m pydoc .\ass16ai.html
problem in .\ass16ai.html - SyntaxError: invalid syntax (ass16ai.html, line 1)
PS C:\Users\shyam\ai assistes code> python -m pydoc .\ass16ai.html
Help on module ass16ai:

NAME
    ass16ai

DESCRIPTION
    calculator.py
        A simple calculator module that provides basic arithmetic operations:
        addition, subtraction, multiplication, and division.

FUNCTIONS
    add(a, b)
        Return the sum of two numbers.

    divide(a, b)
        Return the quotient of two numbers.

        Raises:
            ZeroDivisionError: If b is zero.

    multiply(a, b)
        Return the product of two numbers.

    subtract(a, b)
        Return the difference of two numbers.

FILE
    c:\users\shyam\ai assistes code\ass16ai.html
```

Generate and export the module documentation in HTML

format using the pydoc utility, a

```
PS C:\Users\shyam\ai assistes code> python -m pydoc -w .\ass16ai.html
wrote ass16ai.html
```

[index](#)

ass16ai <c:\users\shyam\ai assistes code\ass16ai.html>

calculator.py

A simple calculator module that provides basic arithmetic operations:  
addition, subtraction, multiplication, and division.

## Functions

**add(a, b)**

Return the sum of two numbers.

**divide(a, b)**

Return the quotient of two numbers.

Raises:

ZeroDivisionError: If b is zero.

**multiply(a, b)**

Return the product of two numbers.

**subtract(a, b)**

Return the difference of two numbers.

## Webserver:

```
S C:\Users\shyam\ai assistes code> python -m pydoc -p1234
erver ready at http://localhost:1234/
erver commands: [b]rowser, [q]uit
erver> b
erver> []
```

## Index of Modules

### Built-in Modules

_abc	_imp	_signal	baseshii
_ast	_interchannels	_sre	builtins
_bisect	_interqueues	_stat	cmath
_blake2	_interpreters	_statistics	errno
_codecs	_io	_string	faulthandler
_codecs_cn	_locale	_struct	gc
_codecs_hk	_lsprof	_symtable	iterools
_codecs_iso2022	_md5	_thread	marshal
_codecs_jp	_multibytecodec	_tokenize	math
_codecs_kr	_opcode	_tracemalloc	mmap
_codecs_rw	_operator	_typing	msvcrt
_collections	_pickle	_warnings	nt
_contextvars	_random	_weakref	os
_csv	_sha1	_winapi	time
_datetime	_sha2	_array	wineg
_functools	_sha3	_atexit	xxsubtype
_heapq			zlib

### C:\Users\shyam\ai assistes code

ass11ai	ass2ai	ass7ai	ass73ai
ass14ai	ass3ai	ass8ai	file_processor
ass16ai	ass3ai	ass9ai	lablai
ass17ai	ass4ai	asshp3	sec
ass1ai	ass5ai	ass12ai	testcasedemo
ass21ai	ass6ai	ass14ai	

### Module Index

future	copyreg	modulefinder	sre_constants
hello	cav	multiprocessing	sre_parse
phello (package)	ctypes (package)	(package)	stl
aix_support	curses (package)	netrc	stat
android_support	dataclasses	nturl2path	statistics
angle_support	datetime	numbers	string
collections_abc	dbm (package)	opcode	stringprep
colorize	decimal	operator	struct
compat_pickle	difflib	optparse	subprocess
compression	dis	os	syntable
ios_support	doctest	pathlib	sysconfig (package)
markupbase	email (package)	pdb	tabnanny
_opcode_metadata	encodings (package)	pickle	tarfile
osx_support	ensurepip (package)	pickletools	tempfile
py_abc	enum	pkgutil	test (package)
pydatetime	filecmp	platform	textwrap
pydecimal	fileinput	plistlib	this
pyio	fimfilter	poolib	threading
pylongs	fractions	posixpath	timeit
pyrepl (package)	functools	pprint	tkinter (package)
sitebuiltins	genericpath	profile	token
strptime	getopt	pstats	tokenize
threading_local	getpass	pty	tomllib (package)
_weakrefset	gettext	py_compile	trace
abc	glob	pyclbr	traceback
antigravity	graphlib	pydoc	traceback
argparse	gzip	pydoc_data (package)	tracemalloc
ast	hashlib	queue	ttv
asyncio (package)	heapq	random	turne
base64	html	re (package)	urllibdemo (package)
bdb	html5lib	reprlib	types
bisect	http (package)	rlcompleter	typing
bz2	idlelib (package)	rungy	unittest (package)
cProfile	imaplib	sched	urllib (package)
calendar	imghdr	secrets	uuid
cmd	importlib (package)	selectors	venv (package)
code	inspect	shelve	warnings
codecs	io	shlex	wave
codeop	ipaddress		weakref

### C:\Users\shyam\AppData\Local\Programs\Python\Python313

### C:\Users\shyam\AppData\Local\Programs\Python\Python313\Lib\site-packages

IPython (package)	httpcore (package)	mistune (package)	requests (package)
PIL (package)	https (package)	nbsphinx (package)	rfc3339_validator
argon2_cffi_bindings (package)	idna (package)	nbclient (package)	rfc3986_validator
ctypes (package)	iniconf (package)	nbconvert (package)	rnds (package)
distutils_hack (package)	ipykernel (package)	nbformat (package)	selphy (package)
pytest (package)	ipython_launcher	nest_asyncio	seaborn (package)
_yaml (package)	ipython_nbextensions_lexers	notebook (package)	send2trash (package)
adodbapi (package)	ipywidgets (package)	notebook_shim (package)	setuptools (package)
anyio (package)	isapi (package)	numba (package)	six
argon2 (package)	isoduration (package)	numpy (package)	sklearn (package)
arrow (package)	jedi (package)	overrides (package)	sniffle (package)
asttokens (package)	jinja2 (package)	packaging (package)	soupsieve (package)
async_lru (package)	joblib (package)	pandas (package)	stack_data (package)
attr (package)	json (package)	pandocfilters	terminado (package)
attrs (package)	jsonpcounter	parso (package)	threadpoolctl
babel (package)	jsonschema (package)	pip (package)	tinyss2 (package)
bleach (package)	jsonschema_specifications (package)	pkg_resources (package)	tornado (package)
bs4 (package)	jupyter	platformdirs (package)	traitlets (package)
certifi (package)	jupyter_client (package)	plugy (package)	typing_extensions
cffi (package)	jupyter_console (package)	prometheus_client (package)	tzdata (package)
charset_normalizer (package)	jupyter_core (package)	prompt_toolkit (package)	uri_template (package)
colorama (package)	jupyter_events (package)	psutil (package)	urllib3 (package)
comm (package)	jupyter_lsp (package)	pure_eval (package)	wcwidth (package)
contourpy (package)	jupyter_server (package)	py	webcolors (package)
cycler (package)	jupyter_server_terminals (package)	pycparser (package)	webencodings (package)
dateutil (package)	jupyterlab (package)	pygments (package)	websocket (package)
debugpy (package)	jupyterlab_pygments (package)	pylab	wiggedsubextension (package)
decorator	jupyterlab_server (package)	pyParsing (package)	
defusedxml (package)			

## 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

Code:

terminal

```
ncher' '49771' '--' 'C:\Users\shyam\ai assistes code\ass16ai.py'
PS C:\Users\shyam\ai assistes code> python -m pydoc .\ass16ai.py
Help on module ass16ai:

NAME
    ass16ai

DESCRIPTION
    conversion.py
    A module that provides number conversion utilities.

FUNCTIONS
    binary_to_decimal(b)
        Convert a binary number to its decimal representation.

        Parameters:
            b (str): The binary number to be converted.

        Returns:
            int: The decimal representation of the binary number.

-- More --
```

html:

```
PS C:\Users\shyam\ai assistes code>
PS C:\Users\shyam\ai assistes code> python -m pydoc -w .\ass16ai.py
wrote ass16ai.html
```

---

[index](#)

ass16ai <c:\users\shyam\ai assistes code\ass16ai.py>

conversion.py

A module that provides number conversion utilities.

## Functions

### **binary\_to\_decimal(b)**

Convert a binary number to its decimal representation.

Parameters:

b (str): The binary number to be converted.

Returns:

int: The decimal representation of the binary number.

Example:

```
>>> binary\_to\_decimal('1010')
10
```

### **decimal\_to\_binary(n)**

Convert a decimal number to its binary representation.

Parameters:

n (int): The decimal number to be converted.

Returns:

str: The binary representation of the decimal number.

Example:

```
>>> decimal\_to\_binary(10)
'1010'
```

### **decimal\_to\_hexadecimal(n)**

Convert a decimal number to its hexadecimal representation.

Parameters:

n (int): The decimal number to be converted.

Returns:

str: The hexadecimal representation of the decimal number.

Example:

```
>>> decimal\_to\_hexadecimal(255)
'ff'
```

---

## Webbrowser:

```
PS C:\Users\shyam\ai assistes code> python -m pydoc -p1234
Server ready at http://localhost:1234/
Server commands: [b]rowser, [q]uit
server> b
server> █
```

The screenshot shows the Python 3.13.2 documentation interface. At the top, it displays the Python version and build information: "python 3.13.2 [tags/v3.13.2:4f8bb39, MSC v.1942 64 bit (AMD64)] Windows-11". To the right, there are links for "Module Index", "Topics", and "Keywords", along with search fields for "Get" and "Search". Below the header, a blue bar contains the text "Index of Modules". A pink bar below it is titled "Built-in Modules" and lists numerous built-in module names. Another pink bar further down is titled "C:\Users\shyam\ai assistes code" and lists several files or modules. The main content area contains a large amount of text, likely the full list of built-in modules, which is mostly illegible due to the small font size.

## 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.

**Code:**

```
:> Users > shyam > ai assistes code > ass16ai.py
 1 """
 2 course.py
 3 A simple course management module that allows adding,
 4 removing, and retrieving course information.
 5 """
 6 # Dictionary to store courses
 7 courses = {}
 8
 9 def add_course(course_id, name, credits):
10     """
11         Add a new course to the course dictionary.
12
13     Args:
14         course_id (str): Unique course identifier.
15         name (str): Name of the course.
16         credits (int): Number of credits for the course.
17
18     Returns:
19         bool: True if course is added successfully.
20     """
21     courses[course_id] = {
22         "name": name,
23         "credits": credits
24     }
25     return True
26
27 def remove_course(course_id):
28     """
29         Remove a course from the course dictionary.
30
31     Args:
32         course_id (str): Unique course identifier.
33
34     Returns:
35         bool: True if course is removed, False if not found.
36     """
37     if course_id in courses:
38         del courses[course_id]
39         return True
40     return False
41
42 def get_course(course_id):
43     """
44         Retrieve course details.
45
46     Args:
47         course_id (str): Unique course identifier.
48
49     Returns:
50         dict or None: Course details if found, otherwise None.
51     """
52     return courses.get(course_id)
53
```

## Terminal

```
PS C:\Users\shyam\ai assistes code> python -m pydoc .\ass16ai.py
Help on module ass16ai:

NAME
    ass16ai

DESCRIPTION
    course.py
    A simple course management module that allows adding,
    removing, and retrieving course information.

FUNCTIONS
    add_course(course_id, name, credits)
        Add a new course to the course dictionary.

        Args:
            course_id (str): Unique course identifier.
            name (str): Name of the course.
            credits (int): Number of credits for the course.

        Returns:
            bool: True if course is added successfully.

    get_course(course_id)
        Retrieve course details.

        Args:
            course_id (str): Unique course identifier.

        Returns:
            dict or None: Course details if found, otherwise None.

    remove_course(course_id)
        Remove a course from the course dictionary.

        Args:
            course_id (str): Unique course identifier.

        Returns:
            bool: True if course is removed, False if not found.
```

Html:

```
PS C:\Users\shyam\ai assistes code>
PS C:\Users\shyam\ai assistes code> python -m pydoc -w .\ass16ai.py
wrote ass16ai.html
PS C:\Users\shyam\ai assistes code> []
```

← → ⌂ File C:/Users/shyam/ai%20assistes%20code/ass16ai.html

[index](#)  
ass16ai <c:\users\shyam\ai assistes code\ass16ai.py>

course.py  
A simple course management module that allows adding, removing, and retrieving course information.

## Functions

**add\_course(course\_id, name, credits)**  
Add a new course to the course dictionary.

Args:  
course\_id (str): Unique course identifier.  
name (str): Name of the course.  
credits (int): Number of credits for the course.

Returns:  
bool: True if course is added successfully.

**get\_course(course\_id)**  
Retrieve course details.

Args:  
course\_id (str): Unique course identifier.

Returns:  
dict or None: Course details if found, otherwise None.

**remove\_course(course\_id)**  
Remove a course from the course dictionary.

Args:  
course\_id (str): Unique course identifier.

Returns:  
bool: True if course is removed, False if not found.

## Data

courses = {}

## Web browser:

```
PS C:\Users\shyam\ai assistes code> python -m pydoc -p1234
Server ready at http://localhost:1234/
Server commands: [b]rowser, [q]uit
server> b
server> 
```

## Index of Modules

### Built-in Modules

<a href="#">abc</a>	<a href="#">_imp</a>	<a href="#">signal</a>	<a href="#">binascii</a>
<a href="#">ast</a>	<a href="#">_interpcchannels</a>	<a href="#">sre</a>	<a href="#">builtins</a>
<a href="#">bisect</a>	<a href="#">_interpoqueues</a>	<a href="#">stat</a>	<a href="#">cmath</a>
<a href="#">blake2</a>	<a href="#">_interpreters</a>	<a href="#">statistics</a>	<a href="#">errno</a>
<a href="#">codecs</a>	<a href="#">_io</a>	<a href="#">string</a>	<a href="#">faulthandler</a>
<a href="#">codecs_cn</a>	<a href="#">_json</a>	<a href="#">struct</a>	<a href="#">gc</a>
<a href="#">codecs_hk</a>	<a href="#">_locale</a>	<a href="#">_svntable</a>	<a href="#">itertools</a>
<a href="#">codecs_iso2022</a>	<a href="#">_lsprof</a>	<a href="#">_sysconfig</a>	<a href="#">marshal</a>
<a href="#">codecs_jp</a>	<a href="#">_multibytecodec</a>	<a href="#">_thread</a>	<a href="#">math</a>
<a href="#">codecs_kr</a>	<a href="#">_opcode</a>	<a href="#">_tokenize</a>	<a href="#">mmap</a>
<a href="#">codecs_tw</a>	<a href="#">_operator</a>	<a href="#">_warnings</a>	<a href="#">mrovert</a>
<a href="#">collections</a>	<a href="#">_pickle</a>	<a href="#">_weakref</a>	<a href="#">nl</a>
<a href="#">contextvars</a>	<a href="#">_random</a>	<a href="#">_winapi</a>	<a href="#">sys</a>
<a href="#">csv</a>	<a href="#">_sha1</a>	<a href="#">array</a>	<a href="#">time</a>
<a href="#">datetime</a>	<a href="#">_sha2</a>	<a href="#">atexit</a>	<a href="#">wincrt</a>
<a href="#">functools</a>	<a href="#">_sha3</a>		<a href="#">xxsubtype</a>
<a href="#">heapq</a>			<a href="#">zlib</a>

C:\Users\shyam\ai assistes code

<a href="#">ass1ai</a>	<a href="#">ass2ai</a>	<a href="#">ass7ai</a>	<a href="#">ass173ai</a>
<a href="#">ass14ai</a>	<a href="#">ass35ai</a>	<a href="#">ass7ai_py</a>	<a href="#">file_processor</a>
<a href="#">ass16ai</a>	<a href="#">ass3ai</a>	<a href="#">ass8ai</a>	<a href="#">lab1ai</a>
<a href="#">ass17ai</a>	<a href="#">ass4ai</a>	<a href="#">ass9ai</a>	<a href="#">sec</a>
<a href="#">ass1ai</a>	<a href="#">ass52</a>	<a href="#">asshp3</a>	<a href="#">testcasesdemo</a>
<a href="#">ass21ai</a>	<a href="#">ass5ai</a>	<a href="#">ass112ai</a>	
<a href="#">ass22ai</a>	<a href="#">ass6ai</a>	<a href="#">ass14ai</a>	