Task-3 – Converting to Object-Oriented Programming (OOP)

Hello everyone,

We have successfully completed the solutions and applications of **Tasks 1 to 9** using functions. Now it's time to take the next step!

What You Need to Do:

Convert **each of the tasks (from 1 to 9)** into a **class-based design** using the OOP principles we've learned.

Implementation Guidelines:

1. For each task, create a **class** that includes:
   - A constructor method `__init__` to receive and store parameters.
   - One or more methods inside the class to perform the task's main functionality.
2. Add **documentation** (docstrings) using a clear format for each class and method, like the following:

Example of a class docstring:

```
python

class Task1:
    """
    Task1 class handles the operation of checking whether a number is even or odd.

    Attributes:
        number (int): The number to be checked.
    """
```

Example of a method docstring:

```
python

def check_even_odd(self):
    """
    Determines whether the number is even or odd.

    Returns:
        str: 'Even' if the number is even, otherwise 'Odd'.
    """
```

3. At the end of each cell:

    - Create an **object** from the class.
    - **Call the main function** using the object.
    - **Print the result** to verify that the class works correctly.

Important Notes:

- Use clear and meaningful names for classes and methods.
- Make sure to **test each class** after implementation.
- Use inline comments when needed to explain the logic.

Feel free to ask any questions — and enjoy applying Object-Oriented Programming in your tasks!