1. Write a Python function named classify\_number that takes an integer as input and returns:

"Positive" if the number is greater than zero.

"Negative" if the number is less than zero.

"Zero" if the number is zero.

Use a while loop to repeatedly prompt the user for a number until they enter a valid integer.

1. Define a Python function calculate\_average that accepts a variable number of arguments (using \*args) and returns the average of those numbers. Include appropriate documentation for the function using a docstring.
2. Write a program that handles user input for a number. Use a try block to catch any ValueError exceptions that may occur if the user inputs an invalid number. Print an error message and prompt the user to enter a valid number again.
3. Create a Python script that writes a list of names to a file called names.txt. Each name should be on a new line. Then, read the file and print each name to the console. Use the with statement to handle file operations and ensure the file is properly closed.
4. Write a Python program that uses a lambda function and the map function to convert a list of temperatures in Celsius to Fahrenheit. The formula for conversion is F = C \* 9/5 + 32. Include a sample list of Celsius temperatures and print the converted list.
5. Create a function called divide\_numbers that takes two parameters, numerator and denominator. Use a try block to handle:

Division by zero (ZeroDivisionError).

Invalid input types (TypeError).

The function should return the result of the division if successful, and print an appropriate error message for each exception.

1. Define a custom exception class called NegativeNumberError that inherits from the built-in Exception class. Write a function check\_positive that raises this exception if a given number is negative.

Demonstrate the use of this function in a try block.

1. Write a Python program that uses the random module to generate a list of 10 random integers between 1 and 100. Then, calculate and print the average of the generated numbers. Use appropriate functions to achieve this.
2. Write a program that:
3. Uses regular expressions to extract all email addresses from a given text.
4. Uses regular expressions to validate a date in the format "YYYY-MM-DD".
5. Uses regular expressions to replace all occurrences of a word with another word in a string.
6. Use regular expressions to split a string by all non-alphanumeric characters.

10. Write a simple client-server program:

* The server should listen for incoming connections and send a message "Hello from server!".
* The client should connect to the server and receive the message.
* Include basic error handling for network operations.

11. What is an API? Explain how to make a GET request to an API using the requests library in Python.

* Explain how to connect to a SQLite database using Python. Describe the steps involved and the purpose of each step.