**Python Error Handling Assignment**

1. WAP to handle the **ZeroDivisionError** exception when dividing a number by zero.
2. WAP to handle the **FileNotFoundError** exception when trying to open a non-existent file.
3. WAP to handle multiple exceptions (**ZeroDivisionError** and **FileNotFoundError**) in the same try-except block.
4. WAP to use the **else** block to execute code when no exception is raised.
5. WAP to use the **finally** block to execute code regardless of whether an exception is raised or not.
6. WAP to create a custom exception class **CustomError** and raise it in your code.
7. WAP to raise a **ValueError** exception with a custom message.
8. WAP to use an **assert** statement to raise an **AssertionError** if a condition is not met.
9. WAP to handle exceptions while reading a file and display its content if no exception occurs.
10. WAP to use nested exception handling to catch different exceptions in different levels of code.

**Python Regular Expression Assignment**

1. WAP to use regular expressions to check if a string contains a specific pattern.
2. WAP to use regular expressions to check if a string contains multiple patterns.
3. WAP to use a regular expression to validate an email address.
4. WAP to use a regular expression to validate a phone number in a specific format.
5. WAP to use regular expressions to extract email addresses from a text.
6. WAP to use regular expressions to replace text in a string.
7. WAP to use regular expressions to split a string into a list of words.
8. WAP to use regular expressions to capture specific parts of a text.
9. WAP to use regular expressions to match whole words.
10. WAP to use se a custom character set to match specific characters.