Lab Exercise: Exception Handling in Python

Objective

Create a Python program that:

1. Allows the user to input a value.
2. Traps for two specific exceptions (ValueError and ZeroDivisionError).
3. Allows the user to try again without the program aborting.
4. Handles any other exceptions and shows the usual exception message.

Instructions

1. **Create a new Python file:** Name the file exception_handling_lab.py.
2. **Define the main function:**
    - This function should contain a loop that allows the user to input a value and attempts to perform a specific operation.
    - The loop should continue until the user enters a valid input.
3. **Exception Handling:**
    - Use try-except blocks to trap for the following exceptions:
        - ValueError: Raised when the user inputs a non-numeric value.
        - ZeroDivisionError: Raised when attempting to divide by zero.
        - Any other exceptions should be handled and display the usual exception message.
4. **Ask the user to input a new value:** Prompt the user to input a new value and try the operation again until a valid input is provided.
5. **Print the result:** Once a valid input is provided and no exceptions occur, print the result of the operation.

Soultion

```
def main():
    while True:
        try:
            # Ask the user to input a number
            user_input = input("Enter a number: ")
            number = int(user_input)

            # Perform an operation (e.g., divide 100 by the input number)
            result = 100 / number

            # Print the result
            print(f"The result of 100 divided by {number} is: {result}")
            break

        except ValueError:
            print("Error: Invalid input. Please enter a valid integer.")

        except ZeroDivisionError:
            print("Error: Division by zero is not allowed. Please enter a non-zero integer.")

        except Exception as e:
            print(f"An unexpected error occurred: {e}")
            break

if __name__ == "__main__":
    main()
```

Testing

1. **Test with non-numeric input:** Ensure that the program traps the ValueError and prompts the user to enter a valid integer.
2. **Test with zero input:** Ensure that the program traps the ZeroDivisionError and prompts the user to enter a non-zero integer.
3. **Test with valid input:** Ensure that the program prints the result of the division operation.
4. **Test with other exceptions:** Ensure that the program handles any other exceptions and displays the usual exception message.