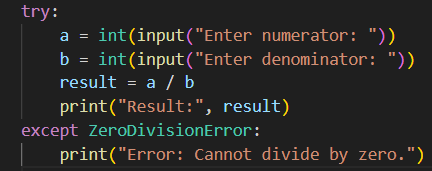
# Exception Handling – Question Paper

-Bavatharani S

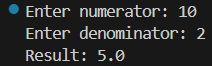
## Section A: Basic Try-Except (2 marks each)

1. Write a program to divide two numbers entered by the user. Handle ZeroDivisionError using try-except.

Code:

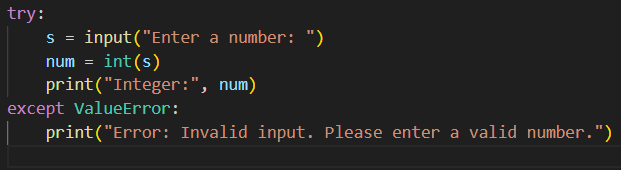


Output:



1. Write a program to convert a string to an integer. Handle ValueError if the input is not a valid number.

Code:

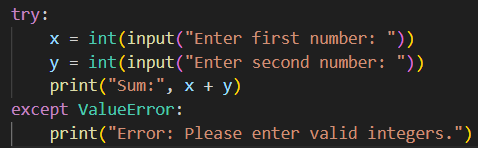


Output:



1. Accept two numbers from the user and perform addition. Use try-except to handle invalid input types.

Code:

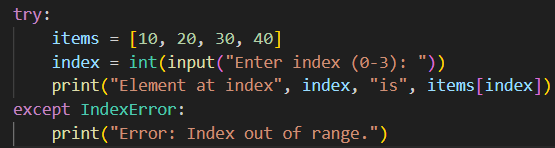


Output:



1. Write a program to read an element from a list using an index entered by the user. Handle IndexError.

Code:



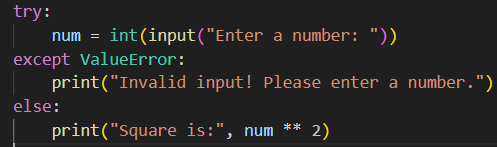
Output:



## Section B: Try-Except-Else (4 marks each)

1. Create a program that accepts a number from the user and prints its square. Use try-except-else to handle ValueError and ensure successful computation is shown only if there's no error.

Code:

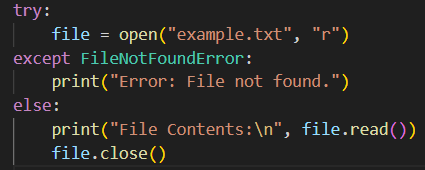


Output:



1. Write a program to open a file and read contents. Use try-except-else to handle FileNotFoundError.

Code:

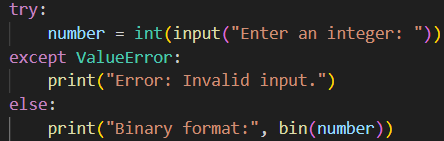


Output:



1. Write a Python program to convert a number to its binary format. Use try-except-else to handle any invalid input.

Code:



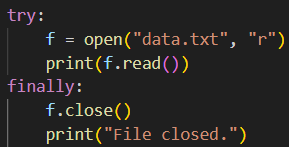
Output:



## Section C: Try-Finally (5 marks each)

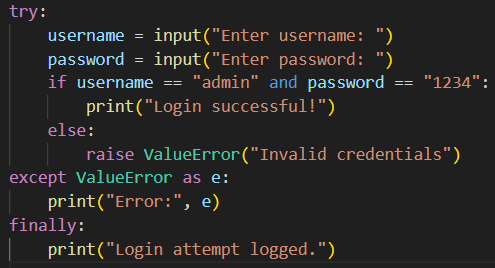
1. Write a program that opens a file and ensures it gets closed, whether or not an exception occurs. Use try-finally.

Code:

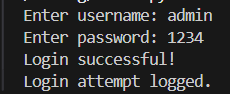


1. Simulate a login process where the user input is handled in a try block and a log message is printed in finally regardless of success or failure.

Code:

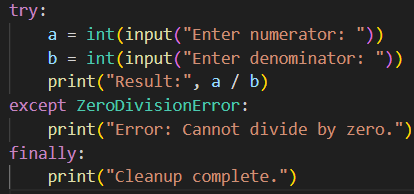


Output:

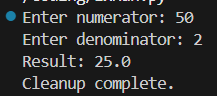


1. Write a program that divides two numbers, catching errors with try-except, and printing a clean-up message using finally.

Code:



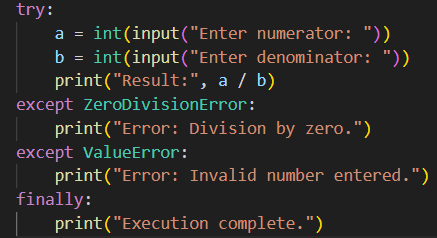
Output:



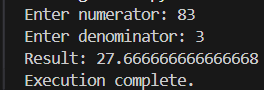
## Section D: Combined Exception Handling (6 marks each)

1. Create a program that handles multiple exceptions: ZeroDivisionError, ValueError, and always prints "Execution complete" using finally.

Code:

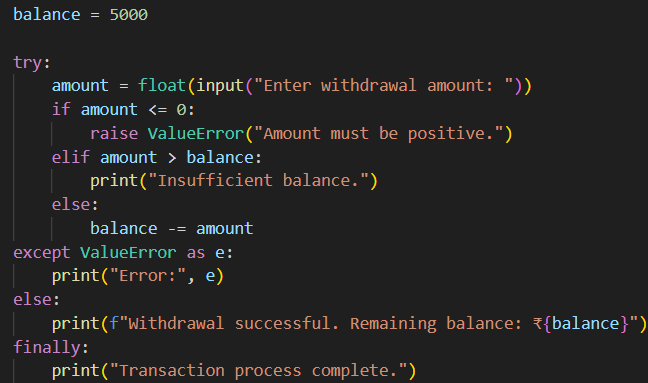


Output:



1. Write a program to simulate bank withdrawal. Use try-except-else-finally to handle incorrect amount input, and always print a message whether the transaction succeeded or failed.

Code:



Output:

