```python
# CS241 Checkpoint 9B
# Written by Chad Macbeth

# We can create our own Exception by creating a class and
# inheriting the Exception class.  Creating your own exceptions
# is useful if you want unique exceptions (helps with readability of
# code sometimes) or if you want to store more data in the exception.
#
# You will notice in this case, that no additional data is stored in the
# new exception.  Once could conclude that this is waste of a class.

class NegativeNumberError(Exception):

    def __init__(self, message):
        super().__init__(message)  # The constructor in Exception class
                                   # takes a string message



def get_inverse(n):
    n_float = float(n)  # If this fails, it will raise ValueError
                        # exception for the main to catch

    if n_float < 0:
        raise NegativeNumberError("Error: The value cannot be negative")  # Raise
        exception instead
                                                                          # of
                                                                          returning an
                                                                          invalid value
                                                                          # Function
                                                                          will exit here

    return 1 / n_float  # If this fails, it will raise ZeroDivisionError
                        # exception for the main to catch

def main():
    n = input("Enter a number: ")
    try:
        result = get_inverse(n)
        print("The result is: {}" .format(result))
    except ValueError:          # Multiple exceptions can be handled after a try block
        print("Error: The value must be a number")
    except NegativeNumberError as e:
        print(e)
    except ZeroDivisionError:
        print("Error: Cannot divide by zero")

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