8. Demonstrate the concept of String-Based Exceptions, Class-Based Exceptions and Nesting Exception handlers.
Name- Gaurav Sudhir Sapkale
Roll no. - 18

```
try:
    print('try block')
    x=int(input('Enter a number: '))
    y=int(input('Enter another number: '))
    z=x/y
except ZeroDivisionError:
    print("except ZeroDivisionError block")
    print("Division by 0 not accepted")
else:
    print("else block")
    print("Division = ", z)
finally:
    print("finally block")
    x=0
    y=0
print ("Out of try, except, else and finally blocks." )
```

Output:

```
1.try block
Enter a number: 2
Enter another number: 0
except ZeroDivisionError block
Division by 0 not accepted
finally block
Out of try, except, else and finally blocks.
2. try block
Enter a number: 2
Enter another number: 3
else block
Division =  0.6666666666666666
finally block
Out of try, except, else and finally blocks.
```

```
2. # define Python user-defined exceptions
class Error(Exception):
    """Base class for other exceptions"""
    pass
class ValueTooSmallError(Error):
    """Raised when the input value is too small"""
    pass
class ValueTooLargeError(Error):
    """Raised when the input value is too large"""
    pass
# you need to guess this number
number = 10
# user guesses a number until he/she gets it right
while True:
    try:
        i_num = int(input("Enter a number: "))
        if i_num < number:
            raise ValueTooSmallError
        elif i_num > number:
            raise ValueTooLargeError
        break
    except ValueTooSmallError:
        print("This value is too small, try again!")
        print()
    except ValueTooLargeError:
        print("This value is too large, try again!")
        print()
print("Congratulations! You guessed it correctly.")
```

Output:

```
Enter a number: 2
This value is too small, try again!
Enter a number: 11
This value is too large, try again!
Enter a number: 10
```