

# CSC 223 - Advanced Scientific Programming

## Errors and Exceptions

# Errors

- There are three main types of errors in Python programming:
  - Syntax errors: errors where the code is not valid Python
  - Runtime errors: errors where syntactically valid code fails to execute
  - Semantic errors: errors in logic – the code executes but the result is not expected.

# Runtime Errors

- Python has an exception handling framework to deal with runtime errors.
- Runtime errors typically cause an exception to occur
- Examples of exceptions:
  - `NameError` – results from referencing an undefined variable
  - `TypeError` – results from undefined operations
  - `IndexError` – results from accessing an element that does not exist.

# Catching Exceptions

- The `try ... except` clause is used to handle runtime exceptions:

```
try:  
    print("this gets executed first")  
except:  
    print("this gets executed on runtime error")
```

# Catching Exceptions Explicitly

- The `except` clause can specify which exception it handles

```
def safe_divide(a, b):  
    try:  
        return a / b  
    except ZeroDivisionError:  
        return 1E100
```

- This will not handle other types of exceptions (which is typically what you want)

```
>>> safe_divide(1, '2')  
TypeError
```

# Raising Exceptions

- The raise statement is used to make an exception occur

```
def fibonacci(N):  
    if N < 0:  
        raise ValueError("N must be non-negative")  
    L = []  
    a, b, = 0, 1  
    while len(L) < N:  
        a, b = b, a + b  
        L.append(a)  
    return L
```

## Accessing the Error Message

- The error message that an exception contains can be referred to explicitly:

```
try:
    x = 1 / 0
except ZeroDivisionError as err:
    print("Error class is: ", type(err))
    print("Error message is:", err)
```

try ... except ... else ... finally

- The else and finally keywords can be used for more exception handling control

```
try:
    print("try something")
except:
    print("this happens only if it fails")
else:
    print("this happens only if it succeeds")
finally:
    print("this happens no matter what")
```