POWERING
POTENTIAL

Exception Handling

What is Exception Handling?

- **Every operation involving user data entry may result in a crash**
  - Every I/O operation, files databases... may crash
- **Good programs are bullet proof**
  - Bad ones dump error message and die!
- **It may not be possible to make safe every route through an app**
  - But you must try
- **Catch exceptions in functions or let them go up the calls stack**
- **When an exceptions is handled it's cleared as if no error occurred**

Exception Handling example

```
import pyodbc

def showCompany():
    connectionString = r'DRIVER={ODBC Driver 13 for SQL Server};
        SERVER=.\SQLExpress;DATABASE=qastore;Trusted_Connection=yes'
    try:
        conn = pyodbc.connect(connectionString)
        cur = conn.cursor()
        result = cur.execute('SELECT * FROM companys').fetchall()
        conn.close()
        return result
    except:
        return None
#---------------------- main -----------------
rows = showCompany()
if rows != None:
    for row in rows:
        print(row)
else:
    print('Error reading data.')
```

Exception Handling in function

```
import pyodbc

def showCompany():
    connectionString = r'DRIVER={ODBC Driver 13 for SQL Server};
                SERVER=.\SQLExpress;DATABASE=qastore;Trusted_Connection=yes'
    try:
        conn = pyodbc.connect(connectionString)
        cur = conn.cursor()
        result = cur.execute('SELECT * FROM company').fetchall()
        conn.close()
        return result
    except Exception as ex:
        print("Error: ", ex)
        return None
#---------------------- main -----------------
rows = showCompany()
if rows != None:
    for row in rows:
        print(row)
```

Exception Handling in main

```
import pyodbc

def getQAStoreConnection():
    connectionString = r'DRIVER={ODBC Driver 13 for SQL Server};
                                SERVER=.\SQLExpress;DATABASE=qastore;Trusted_Connection=yes'
    conn = pyodbc.connect(connectionString)
    cur = conn.cursor()
    return conn

def getQAStoreRows(sql):
    try:
        conn = getQAStoreConnection()
        cur = conn.cursor()
        result = cur.execute(sql).fetchall()
        conn.close()
        return result
    except:
        return None
#---------------------- main -----------------
try:
    rows = getQAStoreRows('SELECT * FROM company')
    if rows != None:
        for row in rows:
            print(row)
except:
    print('Error reading data.')
```

Any Questions?

POWERING
POTENTIAL

Thank you