

"Bill Gates", "Microsoft", "bill@msbob.com"

"Steve Jobs", "Apple", "steve@newton.com"

"Guido Von Rossum", "Google", "IWrotePython@google.com"

```
""" csvDemo.py
    reading content from comma-seperated values (CSV) files
    expects contacts.csv to have name, company, email
    one record per line
"""
```

```
file = open("contacts.csv", "r")
for line in file:
    currentLine = line.split(",")
    (name, company, email) = currentLine
    print "Name:  %s" % name
    print "Company: %s" % company
    print "Email:  %s" % email
    print
```

```
file.close()
```

```
""" pickleDemo.py
demonstrates object serialization
with pickle """

import cPickle

class Critter(object):
    def __init__(self, name = "", age = 0):
        object.__init__(self)
        self.name = name
        self.age = age

def main():
    c = Critter("Lizzie Borden", 23)
    file = open("serialKiller.dat", "w")
    cPickle.dump(c, file)
    file.close()

    file = open("serialKiller.dat", "r")

    d = cPickle.load(file)
    print d.name

    #save an entire list at once
    badCritters = [
        Critter("Darth Vader", 65),
        Critter("Knight who says 'Nii'", 43)
    ]

    file = open("badGuys.dat", "w")
    cPickle.dump(badCritters, file)
    file.close()

    #load them back in
    file = open("badGuys.dat", "r")
    otherGuys = cPickle.load(file)
    for guy in otherGuys:
        print guy.name

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

```
""" exceptions.py
    demonstrate exception-handling
"""

def loopedInput():
    """ keepg trying until you get a legitimate value """
    keepGoing = True
    while keepGoing:
        keepGoing = False
        try:
            x = int(raw_input("please enter a number: "))
        except (ValueError):
            print "That was not a number. Please try again"
            keepGoing = True

def main():
    try:
        x = raw_input("please enter a number: ")
        x = int(x)
        y = 10 / x

    except (ValueError):
        print "Can't convert to integer"
        x = 0
        y = 0
    except (ZeroDivisionError):
        print "Can't divide by zero"
        x = 0
        y = 0
    except:
        print "something went wrong"
        print sys.exc_info()

    print "x:", x
    print "y:", y

    loopedInput()

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

```
""" SQLiteDemo
   Demonstrate building and using
   a database in SQLite
   """
```

```
import sqlite3
```

```
#build connection to database
conn = sqlite3.connect("practice.db")
c = conn.cursor()
```

```
#create a table
c.execute("DROP TABLE IF EXISTS contacts")
sql = """
CREATE TABLE contacts (
  id INTEGER PRIMARY KEY,
  name VARCHAR(20),
  company VARCHAR(20),
  email VARCHAR(20)
) """
c.execute(sql)
```

```
#insert records into the table
c.execute("INSERT INTO contacts VALUES (null, ?, ?, ?)",
         ('Andy Harris', 'IUPUI', 'aharris@cs.iupui.edu'))
c.execute("INSERT INTO contacts VALUES (null, ?, ?, ?)",
         ('Bill Gates', 'Microsoft', 'bill@vista.com'))
c.execute("INSERT INTO contacts VALUES (null, ?, ?, ?)",
         ('Steve Jobs', 'Apple', 'steve@newton.com'))
```

```
#view the results
result = c.execute("SELECT * FROM contacts")
for record in result:
    for field in record:
        print field
    print ""
```

```
#make changes permanent
conn.commit()
c.close()
```