CompSci 101
Introduction to Computer Science

March 26, 2015
Prof. Rodger
Announcements

• No reading or RQ for next time
• Assignment 6 due today
  – Assignment 7 out soon, due after exam 2
• APT 8 is due on Tuesday
• Two practice tests available
  – Do them before Tuesday

• Finish lecture notes from last time
  – dictFun.py, songtitles.py
• Discuss APT EmailsCourse
Michael Stonebraker, Pioneer in Database Systems Architecture, Receives 2014 ACM Turing Award

Michael Stonebraker is being recognized for fundamental contributions to the concepts and practices underlying modern database systems. Stonebraker is the inventor of many concepts that were crucial to making databases a reality and that are used in almost all modern database systems. His work on INGRES introduced the notion of query modification, used for integrity constraints and views. His later work on Postgres introduced the object-relational model, effectively merging databases with abstract data types while keeping the database separate from the programming language.
Think about how parameters work

• Look at variables before passed to a method
• What happens to them after they are passed?
• Can the parameter and argument have the same name?
def testone(num, s):
    num = 4
    s = "Hello"
    print 'In testone, num is ', num, ', s is ', s

'''example 1 '''
print 'example 1'
number = 5
school = 'duke'
print 'number is ', number, ', school is ', school
testone(number, school)
print 'After call to testone, number is ', number, ', school is ', school
Parameter and Argument have the same name

def testone(num, s):
    num = 4
    s = "Hello"
    print 'In testone, num is ', num, ', s is ', s

    """ example 2 ""
    print 'example 2'
    num = 2
    s = 'cat'
    print 'num is ', num, ', s is ', s
    testone(num,s)
    print 'After call to testone, num is ', num, ', s is ', s
```python
def testtwo(lista):
    lista.append(6)
    print 'In testtwo, lista is ', lista

'"'example 3'"'
print 'example 3'
nums = [4,3,8]
ums.append(5)
print 'nums is ', nums
testtwo(nums)
print 'After call to testtwo, nums is ', nums
```
Parameter and Argument have the same name

def testtwo(lista):
    lista.append(6)
    print 'In testtwo, lista is ', lista

''' example 4'''
print 'example 4'
lista = [4,3,8]
lista.append(5)
print 'lista is ', lista
testtwo(lista)
print 'After call to testtwo, lista is ', lista
def testthree(lista):
    print 'In testthree, lista is ', lista
lista = (7,1,3)
print 'In testthree, lista is ', lista

''example 5''
print 'example 5'
lista = (4, 5, 2)
print 'lista is ', lista
testthree(lista)
print 'after call to testthree, lista is ', lista
def testfour(lista):
    print 'In testfour, lista is ', lista
    lista[1] = (4,3)
    print 'In testfour, lista is ', lista

''example 6''
print 'example 6'
lista = [(4, 5, 2),(3,2),(3,4)]
print 'lista is ', lista
testfour(lista)
pint 'after call to testfour, lista is ', lista
```python
def testfour(lista):
    print 'In testfour, lista is ', lista
    lista[1] = (4,3)
    print 'In testfour, lista is ', lista

''''example 7''''
print 'example 7'
mylist = [(4, 5, 2),(3,2),(3,4)]
print 'mylist is ', mylist
testfour(mylist)
print 'after call to testfour, mylist is ', mylist```

```python
def testfive(seta):
    print 'In testfive, seta is ', seta
    seta.add(5)
    seta.add(3)
    print 'In testfive, seta is ', seta

''''example 8''''
print 'example 8'
myset = set([4, 5, 2])
print 'myset is ', myset
testfive(myset)
print 'after call to testfive, myset is ', myset
```
def testsix(seta):
    print 'In testsix, seta is ', seta
    seta.add((5,8))
    seta.add((3,2))
    print 'In testsix, seta is ', seta

'''example 9'''
print 'example 9'
myset = set([(4,3), (5,2,1), (3,2)])
print 'myset is ', myset
testsix(myset)
print 'after call to testsix, myset is ', myset