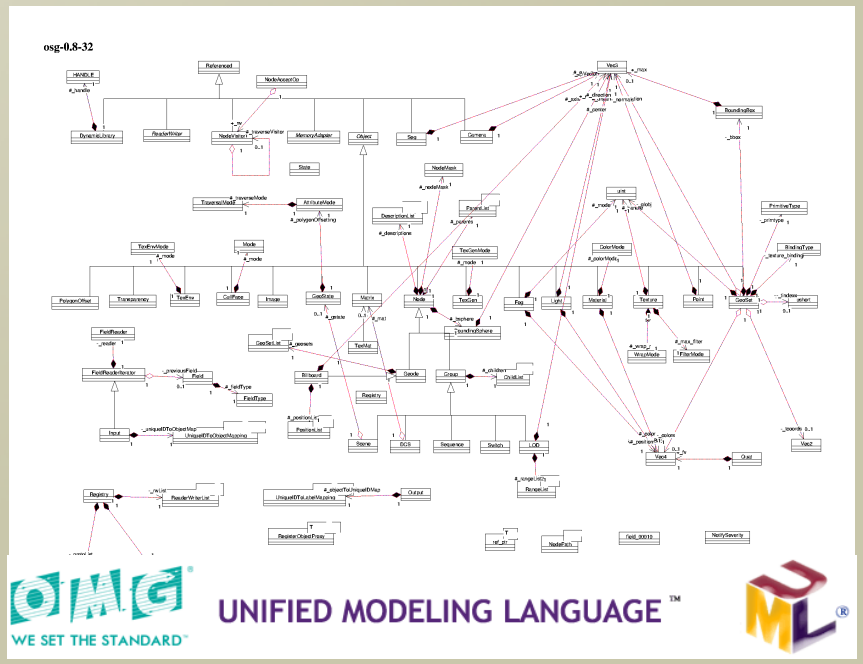
IE 362 Lecture 2: Implementing Class with Diagram

Il-Chul Moon
Department of Industrial and Systems Engineering
KAIST

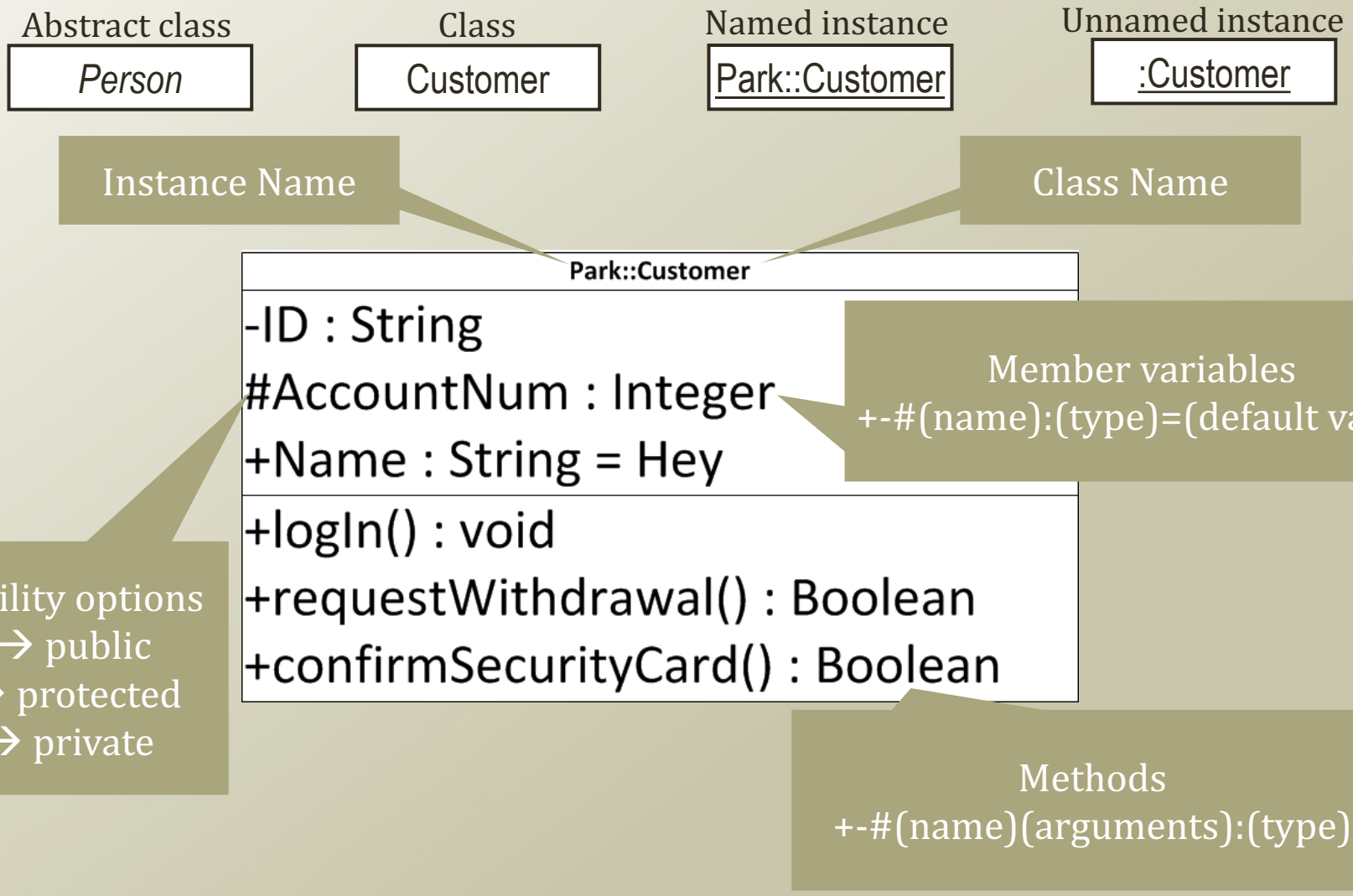
SHORT RECAP

Software Design as House Floorplan

- After your graduation, some of you will be constructors of software
  - Mainly design
  - Some coding
- Need to learn how to communicate your colleagues
  - Learn standard
  - Learn how to represent your design to your boss
- In software engineering,
  - UML is the standard



UML notation : Class and Instance

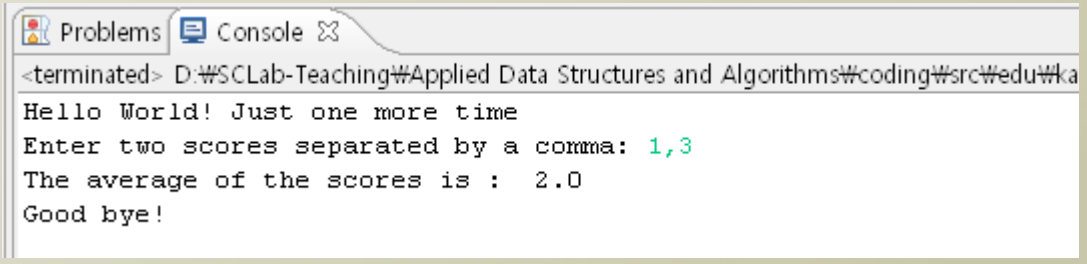


Python Program Structure - Another Hello World

```
class HelloWorld:
    def __init__(self):
        print "Hello World! Just one more time"
    def __del__(self):
        print "Good bye!"
    def performAverage(self,val1,val2):
        average = ( val1 + val2 ) / 2.0
        print "The average of the scores is : ", average

def main():
    world = HelloWorld()
    score1, score2 = input("Enter two scores separated by a comma: ")
    world.performAverage(score1,score2)

main()
```



- Your second Python program in this class
- Object-oriented program
  - HelloWorld is an object
  - __init__, __del__, and performAverage are methods
- Largely in two parts
  - Definition part
  - Execution part

Additional Knowledge for Exercise

```
import numpy as np

a = np.zeros((3, 2))
print (a)
print (len(a),len(a[0]))

b = np.zeros(shape=(2, 3))
print (b)

print (np.shape(b))

c = np.array([[1,2],[3,4],[5,6]])

d = np.arange(np.shape(c)[0])
np.random.shuffle(d)

print (c)
print (d)
print (c[d])

print (np.sqrt(4))
print (np.log(2))
```

```
[[ 0.  0.]
 [ 0.  0.]
 [ 0.  0.]]
3 2
[[ 0.  0.  0.]
 [ 0.  0.  0.]]
(2, 3)
[[1 2]
 [3 4]
 [5 6]]
[0 2 1]
[[1 2]
 [5 6]
 [3 4]]
2.0
0.69314718056
```

OFFLINE CLASS PLAN

Product Review and Sentiment Analysis

- Objective
  - Make the complete analysis: calculate the analysis accuracy
  - Use the attribute of classes
  - Make the charts
- Problem 1. ***runAnalysis***
  - Complete the if-then statements to count the accuracy of analysis
- Problem 2. ***runWholeAnalysis***
  - Complete the for loop and the related codes to run the analysis for the entire testing set.
- Problem 3. ***runExperiments***
  - Complete the for loop to calculate the averages and the standard deviations
  - Calculate the average and the standard deviations

Problem 1. *runAnalysis*

- Find how to determine the correct case for a single trial

```
# return correct as 1 if the review is positive and the analysis is positive and if the review is negative and the analysis is negative
# return correct as 0 otherwise
# self.dataReviewTesting stores the correct review result by specifying 1 as a positive review
if self.dataReviewTesting[idxReview] == 1:
    if ????????????????????????????????:
        ????????????????????????????????
    else:
        ????????????????????????????????
else:
    if ????????????????????????????????:
        ????????????????????????????????
    else:
        ????????????????????????????????
return probLogPositive, probLogNegative, correct
```

Problem 2. *runWholeAnalysis*

- Find how to count the correct case for multiple trials
  - You need to count the correct cases
  - You need to store the count
- Vary the training dataset size
  - By looping with 30 case interval
  - Create the training size as “j”

```
# for loop with 0, 30, 60, 90, 120, 150
# make
# numCorrect(0) = (sum of correct cases for 0 case) / (size of testing which is 1 in the current iteration)
# numCorrect(1) = (sum of correct cases for 30 case) / (size of testing which is 30 in the current iteration)
# and so on...
for j in ????????????????????????????????:
    self.dataSentimentTraining = self.sentidata[self.shuffle[0:j+1], 0:self.wordLimit]
    self.dataReviewTraining = self.sentidata[self.shuffle[0:j+1], -1]
    numCorrect[cnt] = 0
    for i in range(np.shape(self.dataSentimentTesting)[0]):
        p, n, c = self.runAnalysis(i)
        if c == 1:
            numCorrect[cnt] += 1
    numCorrect[cnt] = numCorrect[cnt] / np.shape(self.dataSentimentTesting)[0]
    cnt += 1
return numCorrect
```

Problem 3. *runExperiments*

- Find how to calculate the average performance and the confidence interval
  - Calculating the standard deviation with a single pass of a loop

```
# iterate by the numReplicate
for i in range(numReplicate):
    self.shuffle = np.arange(np.shape(self.sentidata)[0])
    np.random.shuffle(self.shuffle)

    self.dataSentimentTesting = self.sentidata[self.shuffle[self.numTraining+1:198], 0:self.wordLimit]
    self.dataReviewTesting = self.sentidata[self.shuffle[self.numTraining + 1:198], -1]

    # receive the correct information from runWholeAnalysis()
    correct = self.runWholeAnalysis()
    # calculate the average by the training case sizes
    average = ????????????????????????????????
    # calculate the squared average by the training case sizes
    averageSq += ????????????????????????????????

# finish the calculation of average
average = ????????????????????????????????
# finish the calculation of average squared
averageSq = ????????????????????????????????
# finish the calculation of standard deviation
std = ????????????????????????????????
```

Execution Result

- When you complete the codes, you are expected to see the below results.
  - But, the chart might not look same