CSC 111
Introduction to Computer Science
Spring 2018 — Week 1

Dominique Thiébaut
dthiebaut@smith.edu
Quick Review
Goals for This Week

- Learn the Rules for **Pair Programming**
- Learn how to use **Idle**
- Write simple programs that use **variables**, **for loops**, and **output** information
- **Install** Python and Idle on laptop (optional)
- Learn how to **submit** Python programs to **Moodle** (lab+homework)
• Read **Chapter 1** in John Zelle's *Python Programming*
What is a Programming language?
Important Concepts...

• Syntax and keywords

  and del from not while as elif global or with assert else if pass yield break except import print class exec in raise continue finally is return def for lambda try

• Algorithm
Rules for Pair Programming
An Example Program
# A simple program taken from Zelle, Chapter 1
# D. Thiebaut

```python
def main():
    print( "This program illustrates a chaotic function" )
    x = eval( input( "Enter a number between 0 and 1: " ) )
    for i in [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]:
        x = 3.9 * x * ( 1 - x )
    print( x )

main()
```

D. Thiebaut, Computer Science, Smith College
```
# A simple program taken from Zelle, Chapter 1
# D. Thiebaut

def main():
    print( "This program illustrates a chaotic function" )
    x = eval( input( "Enter a number between 0 and 1: " ) )
    for i in [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]:
        x = 3.9 * x * ( 1 - x )
    print( x )

main()
```

**INDENTATION IS IMPORTANT**

**COMMENT**

**DIFFERENT COLORS: SYNTAX HIGHLIGHTING**

**SPECIAL TOOL: EDITOR IDE**
Integrated Development Environment = IDLE
Integrated Development Environment = IDLE
Integrated Development Environment = IDLE
Integrated Development Environment = IDLE

(Windows)
A simple program taken from Zelle, Chapter 1

```python
def main():
    print( "This program illustrates a chaotic function" )
    x = eval( input( "Enter a number between 0 and 1: " ) )
    for i in [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]:
        x = 3.9 * x * ( 1 - x )
        print( x )

main()
```
Beginning of the Semester...
Concepts to Cover in Demo

• Console vs. Edit window

• Variables
  • numbers: integers and floats
  • text: strings of characters

• print function
Demo Programs To Play With…

```python
age = 20
year = 2015
yearBorn = year - age

print( "you are", age )
print( "you were born in", yearBorn )
```

```python
name = "Alex"
college = "Smith College"
print( name, "goes to", college )
```

```python
for name in [ "Lea Jones", "Julie Fleur", "Anu Vias" ]:
    print( name )
    print( "———"
```
Demo Programs To Play With… (cont’d)

```python
for name in [ "Lea Jones", "Julie Fleur", "Anu Vias" ]:
    print( name, len( name ) )
```

```python
print( "hello" * 4 )
print( "-" * 10 )
greetings = "hello"
dash = "-"
print( greetings * 4 )
print( dash * 10 )
greetings = "hello"
longGreetings = greeting * 4
print( greetings )
print( longGreetings )
```
Demo Programs To Play With… (cont’d)

```
for name in [ "Lea Jones", "Julie Fleur", "Anu Vias" ]:
    bar = len( name ) * "-"
    print( name )
    print( bar )

print( "hello" * 4 )
print( "-" * 10 )

greetings = "hello"
dash = "-"
print( greetings * 4 )
print( dash * 10 )

greetings = "hello"
longGreetings = greeting * 4
print( greetings )
print( longGreetings )
```
Exercise 1

Lea
Mary
Alice
Lujun
Anu
Shweta

====== RESTART: /Users/thiebaut/Desktop/Drop

Lea
Mary
Alice
Lujun
Anu
Shweta

>>>
Exercise 2

<table>
<thead>
<tr>
<th>Name</th>
<th>Box</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lea</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lujun</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shweta</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exercise 3

Lea
Mary
Alice
Lujun
Anu
Shweta

```
== RESTART: /Users/thiebaut/Desktop/Dropbox/1:
Lea
 ++ Box: | Id:
Mary
 ++ Box: | Id:
Alice
 ++ Box: | Id:
```
<table>
<thead>
<tr>
<th>Box</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Anu

<table>
<thead>
<tr>
<th>Box</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Shweta

<table>
<thead>
<tr>
<th>Box</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

>>>
We stopped here last time...
Introduction to Lab 1

Assignment

Introduction to Variables

Exercise
Practice Python!
Beginning of the Semester
AFTER ONE SEMESTER
Computer Science Major
Final Project From the Past

https://www.youtube.com/watch?v=g_82xHimSNE
age
Variables

age = 23
Variables

age = 23

assignment
Variables

name

\texttt{name = "Smith"}

assignment
Variables

rate = 21.34

rate = 21.34
Variables & Expressions

age = 23
newValue = 10
Variables & Expressions

age = 23
enewValue = 10
age = newValue
Variables & Expressions

```plaintext
age = 23
newValue = 10
age = newValue
```
Variables & Expressions

age = 23
newValue = 10
age = newValue
age = age + 2
Variables & Expressions

10

age

newValue

age = 23
newValue = 10
age = newValue
age = age + 2

10

10
Variables & Expressions

age = 23
newValue = 10
age = newValue
age = age + 2

10+2
Variables & Expressions

10 + 2 → 12

age = 23
newValue = 10
age = newValue
age = age + 2
Variables & Expressions

10 + 2 → 12

age = 23
newValue = 10
age = newValue
age = age + 2
a = 10
b = 20
c = 30
a = b
# a = ?
Exercise

\[
a = 10 \\
b = 20 \\
c = 30 \\
a = b \\ # a = 20 \\
b = a \\ # a = ? \quad b = ?
\]
Exercise

\begin{align*}
a &= 10 \\
b &= 20 \\
c &= 30 \\
a &= b \quad \# a = 20 \\
b &= a \quad \# a = 20 \quad b = 20 \\
c &= c \times 2 \quad \# c = ?
\end{align*}
a = 10
b = 20
c = 30

a = b  # a = 20
b = a  # a = 20   b = 20
c = c * 2 # c = 60
d = d - 10 # d = ?
Exercise

```python
a = 10
b = 20
c = 30
a = b  # a = 20
b = a  # a = 20  b = 20
c = c * 2  # c = 60
d = d - 10  # NameError:
# name 'd' is not defined
```
Naming Variables

• Variable name cannot be a **keyword**
  
  and del from not while as elif global or with assert else if pass yield break except import print class exec in raise continue finally is return def for lambda try

• First letter must be **alphabetic** (upper- or lower-case, or underscore)

• Can be followed by 0, 1, or more **letters**, **digits**, or **underscore**
Naming Variables

a
age
delta
name1
name2
R2D2
aVeryLongName
1tooMany
Naming Variables

a
age
delta
name1
name2
R2D2
aVeryLongName
1tooMany
Naming Variables

this_is_good_too
but
we Prefer
thisIsGoodToo

lambda
for
def
this_is_good_too
but
we Prefer
thisIsGoodToo

-CamelCase-

-lambda-
-for-
-def-
Exercise 1

***
Mae
*****
Alice
*******
Felicia
Exercise 2
(Tricky and Unfair)

*  
Mae  
*******  
Alice  
****  
Felicia  
**
We stopped here last time…
• The Programming Process
• Memory: RAM
• Variables revisited
  • Literals: numbers, strings, lists
  • Types: type( )
  • Multiple assignments
  • Operators. Overloaded operators.
• Loops
  • range( ); list( )
• Programming exercises
The Programming Process
The Programming Process

- Analyze the Problem
The Programming Process

- Analyze the **Problem**
- Determine **Specifications**
The Programming Process

- Analyze the **Problem**
- Determine **Specifications**
- Create a **Design**
The Programming Process

- Analyze the **Problem**
- Determine **Specifications**
- Create a **Design**
- **Implement**
The Programming Process

- Analyze the **Problem**
- Determine **Specifications**
- Create a **Design**
- **Implement**
- **Test & Debug**
The Programming Process

- Analyze the Problem
- Determine Specifications
- Create a Design
- Implement
- Test & Debug

iterate many times
The Programming Process
The Programming Process
The Programming Process
The Programming Process
The Programming Process
Back to the Memory
What does the memory really look like?
What does the memory really look like?
What does the memory really look like?
Motherboard

What does the memory really look like?

Random Access Memory (RAM)
What does the memory really look like?

Random Access Memory (RAM)

Single In-line Memory Module (SIMM)
• RAM: 4, 8, 12, 16 GigaBytes

• **Giga** = billion: $10^9$ bytes

• In RAM: room for approximately **2 billion** integers

• 1 number takes **4 bytes**

• 1 character takes **1 byte** (sometimes **2 bytes**)
How big is 2 Billion?

2 billion integers

How tall are 2 billion quarters?
How big is 2 Billion?

2 billion integers

How tall are 2 billion quarters

2 miles, or 3.2 km!
Variables and Assignment

age = 3

age
Variables and Assignment

```python
age = 3
name = "Smith"
rate = 0.06
```
Variables and Assignment

```
age = 3
name = "Smith"
rate = 0.06
```

literals
Variables and Assignment

```python
age = 3
name = "Smith"
rate = 0.06

age = age * 2  # double the age
age = age + 1  # increment the age
```
Variables and Assignment

```
age = 3
name = "Smith"
rate = 0.06

age = age * 2  # double the age
age = age + 1  # increment the age

name = name + " College"  # name will contain "Smith College"
```
Variables and Assignment

In a programming language, operators may have different meanings depending on the context.
Variables and Assignment

Overloaded operators

```plaintext
age = 3
name = "Smith"
rate = 0.06

age = age * 2
# double the age
age = age + 1
# increment the age
name = name + " College"
# name will contain "Smith College"
```

D. Thiebaut, Computer Science, Smith College
Exercises
Exercises

Guess what Python will do

```python
age = 3
name = "Smith"
rate = 0.06

age = age * rate
```
Exercises

Guess what Python will do

```python
age = 3
name = "Smith"
rate = 0.06

age = age * rate  # age will contain 0.18
name = "his + hers"  # name will contain "his + hers"
rate = name * rate
```
Exercises

Guess what Python will do

```python
age = 3
name = "Smith"
rate = 0.06

age = age * rate  # age will contain 0.18
name = "his + hers"  # name will contain "his + hers"
rate = name * rate  # TypeError: can't multiply sequence by 'float'
```
Exercises

Guess what Python will do

```python
name = "Smith"
col = name + " College" * 2

print( col )

# output
```
Guess what Python will do:

```python
name = "Smith"
col = name + " College" * 2

print( col )

# output
```
Exercises

Guess what Python will do

```python
name = "Smith"
col = name + " College" * 2

print( col )

# output
# Smith College College
```
Using the Shell...

```python
name = "Smith College"
col = "Collage"
# output
# Smith College

print( col )

>>> print( col )
```

```
SmithCollegeCollege
SmithCollegeSmithCollege
```

```
Smith College College
Smith CollegeSmith College
```

```
Smith CollegeSmith College
```

```
```
```
Simultaneous Assignments
Simultaneous Assignments

```python
Python 3.5.4 (v3.5.4:3f56838976, Aug 7 2017, 12:56:33)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> a, b, c = 10, 20, 30
>>> a
10
>>> b
20
>>> c
30
>>> |
```
Swapping Variables

```python
>>> a
20
>>> b
10
>>> a, b = b, a
```

D. Thiebaut, Computer Science, Smith College
Lists and Variables

```python
>>> a
10
>>> b
20
>>> c
30
>>> a, b, c
(10, 20, 30)
>>> triplet = a, b, c
>>> x, y, z, = triplet
```
a, b, c = 10, 20, 30

# a = 10, b = 20, c = 30

triplet = a, b, c

# triplet = (10, 20, 30)

x, y, z = triplet

# x = 10
# y = 20
# z = 30
• The Programming Process

• Variables

• Definite Loops

• Input
for <var> in <sequence>:  
<body>
for <var> in <sequence>:
<body>

for can in [pepsi, Coca-Cola, Crush, Dr. Pepper, Fanta]:
open( can )
drink( can )
throwAway( can )
for <var> in <sequence>:
<body>

for can in [pepsi, CocaCola, Crush, DrPepper, Mist]:
open(can)
drink(can)
throwAway(can)

Many actions repeated, each group for each can