CSC 111
Introduction to Computer Science
Spring 2018 — Week 1

Dominique Thiébaut
dthiebaut@smith.edu
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

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()
# 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()
Integrated Development Environment = IDLE
Integrated Development Environment = IDLE
Integrated Development Environment = IDLE

(MAC)
Integrated Development Environment = IDLE

(Windows)
# 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()
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 = "-"
greetings = greeting * 4
print( greetings * 4 )
print( dash * 10 )
greetings = "hello"
longGreetings = greeting * 4
print( greetings )
print( longGreetings )
```
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
<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>
</tbody>
</table>
Lea
Mary
Alice
Lujun
Anu
Shweta

Exercise 3

=*=*=*=*= RESTART: /Users/thiebaut/Desktop/Dropbox/1:
Lea
+---+-+-
| Box: | Id: |
+---+-+-

Mary
+---+-+-
| Box: | Id: |
+---+-+-

Alice
+---+-+-
| Box: | Id: |
+---+-+-
Exercise 4

Lea
Mary
Alice
Lujun
Anu
Shweta
We stopped here last time...
Outline

• Introduction to Lab 1
• Introduction to Variables
Lab 1
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
Memory
Variables

age
Variables

```
age = 23
```
Variables

age = 23

assignment

d. thiebaut, computer science, smith college
name = “Smith”
Variables

rate = 21.34

rate assignment
Variables & Expressions

\[ \text{age} = 23 \]
\[ \text{newValue} = 10 \]
Variables & Expressions

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

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

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

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

10 + 2

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

10 + 2 → 12

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

10 + 2 \rightarrow 12

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

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

```
a = 10
b = 20
c = 30
a = b                   # a = 20
b = a                   # a = ?      b = ?
```

Exercise

a = 10
b = 20
c = 30
a = b                  # a = 20
b = a                  # a = 20    b = 20
c = c * 2              # c = ?
\[ \begin{align*}
    a &= 10 \\
    b &= 20 \\
    c &= 30 \\
    a &= b \quad \# a = 20 \\
    b &= a \quad \# a = 20, \ b = 20 \\
    c &= c \times 2 \quad \# c = 60 \\
    d &= d - 10 \quad \# d = ?
\end{align*} \]
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)

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

- a
- age
- delta
- name1
- name2
- R2D2
- aVeryLongName
- 1tooMany
Naming Variables

CamelCase

a
age
delta
name1
name2
R2D2
aVeryLongName

1tooMany

D. Thiebaut, Computer Science, Smith College
This is good too
but we prefer thisIsGoodToo

lambda
for
def
Naming Variables

this_is_good_too
but
we Prefer
thisIsGoodToo

CamelCase

lambda
def
Exercise 1

***

Mae

*****

Alice

******

Felicia
Exercise 2
(Extremely Hard and Unfair)

* 
Mae
*******
Alice
****
Felicia
**