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
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)
# A sample 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
# 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( "—————" )
```
for name in [ "Lea Jones", "Julie Fleur", "Anu Vias" ]:
    print( name, len( name ) )

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

RESTART: /Users/thiebaut/Desktop/Drop
Lea
Mary
Alice
Lujun
Anu
Shweta
>>>
Exercise 2

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>

RESTART: /Users/thiebaut/Desktop/Dropbox
Exercise 3

Lea
Mary
Alice
Lujun
Anu
Shweta
Exercise 4

<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>
We stopped here last time…
Outline

• Introduction to Lab 1
• Assignment
• Introduction to Variables
• Exercise
Practice Python!
Beginning of the Semester
AFTER ONE SEMESTER
Computer Science Major

D. Thiebaut, Computer Science, Smith College
Final Project From the Past

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

age = 23

age = 23
Variables

age = 23

assignment
Variables

name = "Smith"

assignment
Variables

rate = 21.34

rate = 21.34

assignment
Variables & Expressions

age = 23
newValue = 10

age
newValue
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

\[ \text{age} = 23 \]
\[ \text{newValue} = 10 \]
\[ \text{age} = \text{newValue} \]
\[ \text{age} = \text{age} + 2 \]
Variables & Expressions

\[ 10 + 2 \rightarrow 12 \]

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

10 + 2 $\rightarrow$ 12

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

\[
\begin{align*}
  a &= 10 \\
  b &= 20 \\
  c &= 30 \\
  a &= b & \text{# a = 20} \\
  b &= a & \text{# a = ? b = ?}
\end{align*}
\]
Exercise

a = 10
b = 20
c = 30

a = b                    # a = 20
b = a                    # a = 20    b = 20

\[
c = c \times 2 \quad \# c = ?
\]
Exercise

```
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 = ?
```
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

CamelCase
Camel Case

Naming Variables

a
age
delta
name1
name2
R2D2
aVeryLongName

1tooMany
Naming Variables

this_is_good_too
but
we Prefer
thisIsGoodToo

lambda
for
def
Naming Variables

this_is_good_too
but
wePrefer
thisIsGoodToo

---
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

Refine the specifications 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?
What does the memory really look like?

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

Motherboard

Random Access Memory (RAM)

Single In-line Memory Module (SIMM)
What does the memory really look like?

- RAM: 4, 8, 12, 16 GigaBytes
- \textbf{Giga} = billion: $10^9$ bytes
- In RAM: room for approximately \textbf{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 tall are 2 billion quarters?

2 miles, or 3.2 km!

2 billion integers
Variables and Assignment

age = 3

age
Variables and Assignment

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

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

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

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

```python
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**

```python
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

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"

Overloaded operators
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
```

D. Thiebaut, Computer Science, Smith College
Exercises

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 = ( name + " " + col for col in col )

>>> 
======= RESTART: /Users/thiebaut/Desktop/Week1Friday.py 
======= 
SmithCollegeCollege
SmithCollegeSmith College
>>> 
======= RESTART: /Users/thiebaut/Desktop/Week1Friday.py 
======= 
Smith College College
Smith College Smith College
>>> 
======= RESTART: /Users/thiebaut/Desktop/Week1Friday.py 
======= 
Smith College Smith College
>>> 
```
Simultaneous Assignments
Simultaneous Assignments

```python
>>> a, b, c = 10, 20, 30
>>> a
10
>>> b
20
>>> c
30
>>> |
```
Swapping Variables

```python
>>> a
20
>>> b
10
>>> a, b = b, a
```
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 [ ]:
open( can )
drink( can )
throwAway( can )
for <var> in <sequence>:
<body>

for can in [pepsi, CocaCola, Crush, DietDrPepper, Sprite, MountainMist]:
  open(can)
  drink(can)
  throwAway(can)

Many actions repeated, each group for each can
for <var> in <sequence>:
<body>

for name in [ "Alex", "Max", "Rui" ]:
  open( can )
  drink( can )
  throwAway( can )
for <var> in <sequence>:
<body>

for x in range(10):
    print(x)
http://docs.python.org/3/
The Python Standard Library

While The Python Language Reference describes the exact syntax and semantics of the Python language, this manual describes the standard library that is distributed with Python. It also describes some of the modules that are commonly included in Python distributions.

Python’s standard library is very extensive, offering a wide range of facilities as indicated by the table below. The library contains built-in modules (written in C) that provide access to system functionality otherwise be inaccessible to Python programmers, as well as modules written in Python that provide solutions to many problems that occur in everyday programming. Some of these modules are explicitly designed to assist with the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often additional components. For Unix–like operating systems Python is normally provided as a collection of packages to use the packaging tools provided with the operating system to obtain some or all of the optional components.

In addition to the standard library, there is a growing collection of several thousand components (from modules to packages and entire application development frameworks), available from the Python Package Index (PyPI).

- 2. Built-in Functions
- 3. Built-in Constants
  - 3.1. Constants added by the site module
- 4. Built-in Types
  - 4.1. Truth Value Testing
  - 4.2. Boolean Operations — and, or, not
This code is exactly equivalent to the first example. Be sure to give the additional functions the same name as the original property (x in this case.)

The returned property object also has the attributes \texttt{fget}, \texttt{fset}, and \texttt{fdel} corresponding to the constructor arguments.

\texttt{range(stop)}
\texttt{range(start, stop[, step])}

Rather than being a function, \texttt{range} is actually an immutable sequence type, as documented in \texttt{Ranges} and \texttt{Sequence Types — list, tuple, range}.

\texttt{repr(object)}

Return a string containing a printable representation of an object. For many types, this function makes an attempt to return a string that would yield an object with the same value when passed to \texttt{eval()}, otherwise the representation is a string enclosed in angle brackets that contains the name of the type of the object together with additional information often including the name and address of the object. A class can control what this...
Examples to Try Out:

```python
for x in range( ... ): # replace ... with
    print( x ) # range expression
    # below:

# range( 10 )
# range( 2, 10 )
# range( -5, 5 )
# range( 0, 10, 2 )
# range( 0, 10, 3 )
# range( 9, 0, -1 )
```
Exercise

Generate an equivalency table of temperatures in Fahrenheit and Celsius. 100 F should be on the first line, and -30F on the last line. Show only Fahrenheit temperatures that are multiples of 10.

Celsius = (Farhenheit - 32) * 5 / 9