CS 141, Lecture 3

Please login to the Math/Programming profile, and start up IDLE.

CS fact of the day: The first 1GB hard drive was announced in 1980. It weighed about 550 pounds, and had a price tag of $40,000.

Warmup: You are driving along an empty highway in South Carolina, keeping up with the flow of traffic, but going 74 miles-per-hour in a 60 miles-per-hour zone*. Assume that the fine is $15 for every mile per hour over the limit.

Discuss with the person sitting next to you which of the following programs correctly prints your fine.

* This in no way reflects any recent events that have happened to your professor.

You are caught going 74 miles/hour in a 60 miles/hour zone. The fine is $15 for every mile per hour over the limit. Which of the following programs correctly prints your fine? (there may be more than one)

1

```
speed = 74
limit = 60
print(fine)
fine = (speed – limit) * 15
```

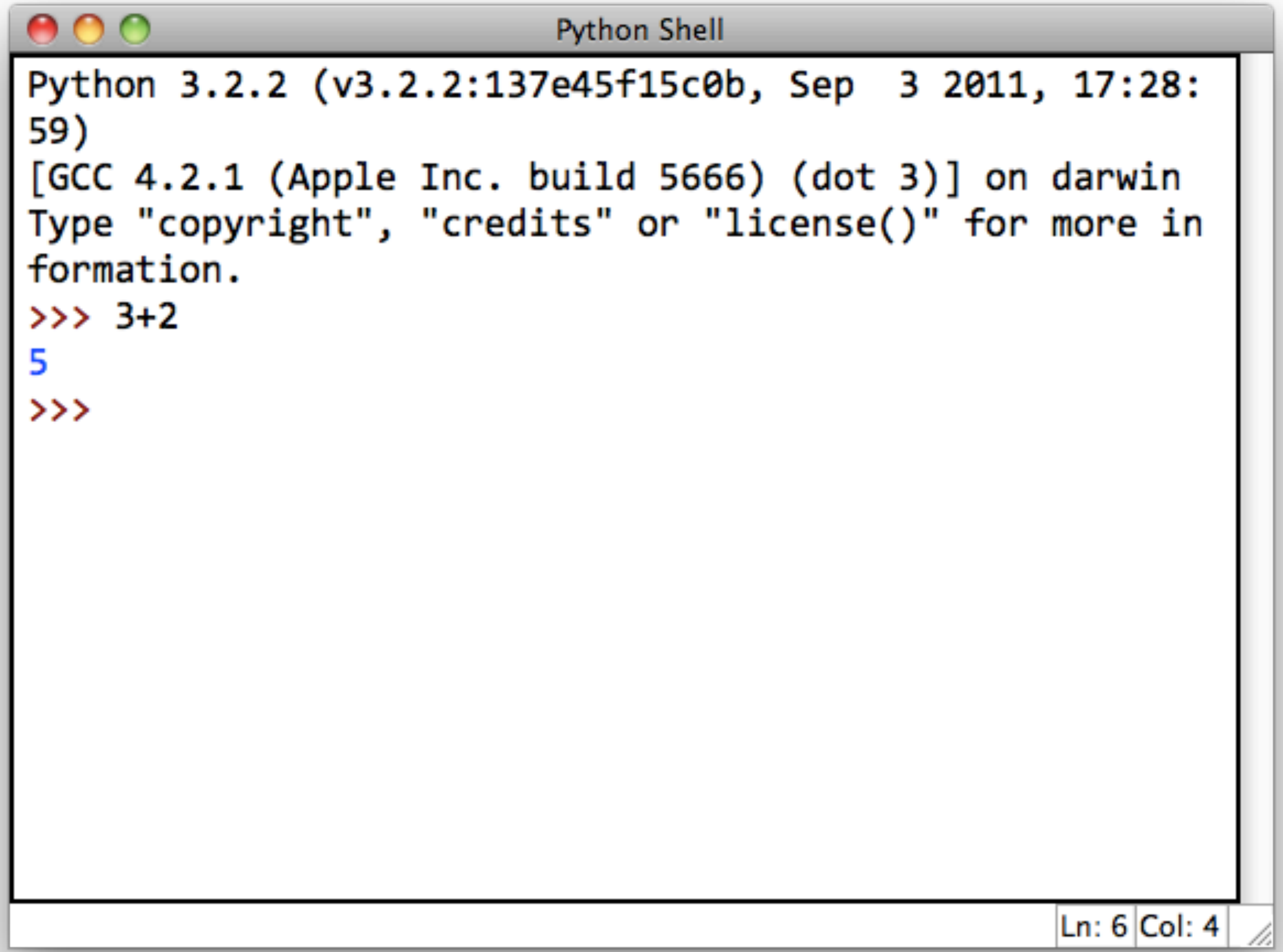
2

```
speed = 74
limit = 60
fine = over * 15
over = (speed – limit)
print(fine)
```

3

```
limit = 60
speed = 74
fine = (speed – limit) * 15
print(fine)
```

Python Shell

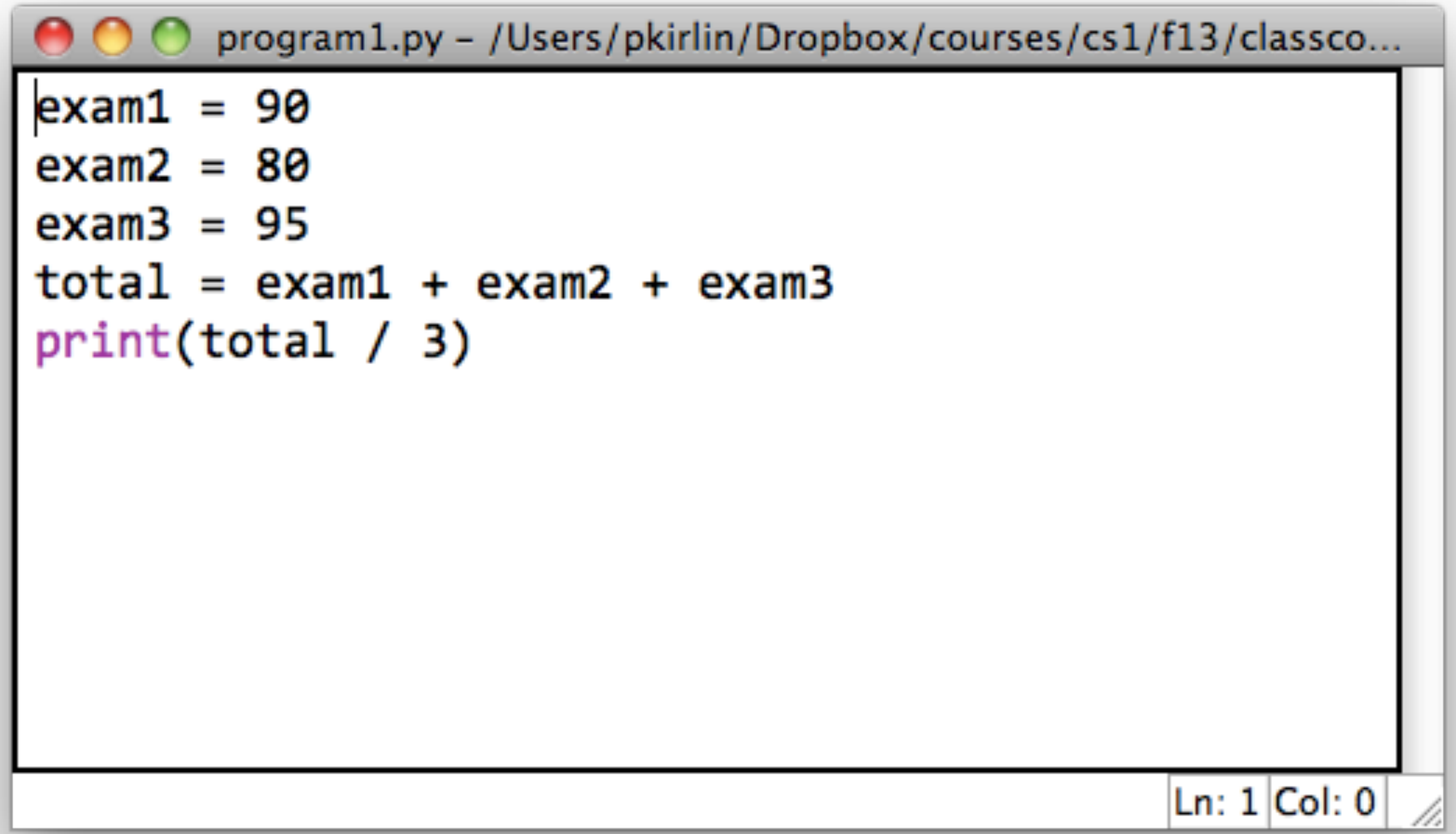


```
Python 3.2.2 (v3.2.2:137e45f15c0b, Sep  3 2011, 17:28:
59)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more in
formation.
>>> 3+2
5
>>>
```

Python Shell

- Runs single-line "mini-programs"
- Runs each line after you type it and press enter.
- Results of computations are automatically ***printed*** (displayed) back to you.

Longer Programs


program1.py – /Users/pkirlin/Dropbox/courses/cs1/f13/classco...
Ln: 1 Col: 0

```
exam1 = 90
exam2 = 80
exam3 = 95
total = exam1 + exam2 + exam3
print(total / 3)
```

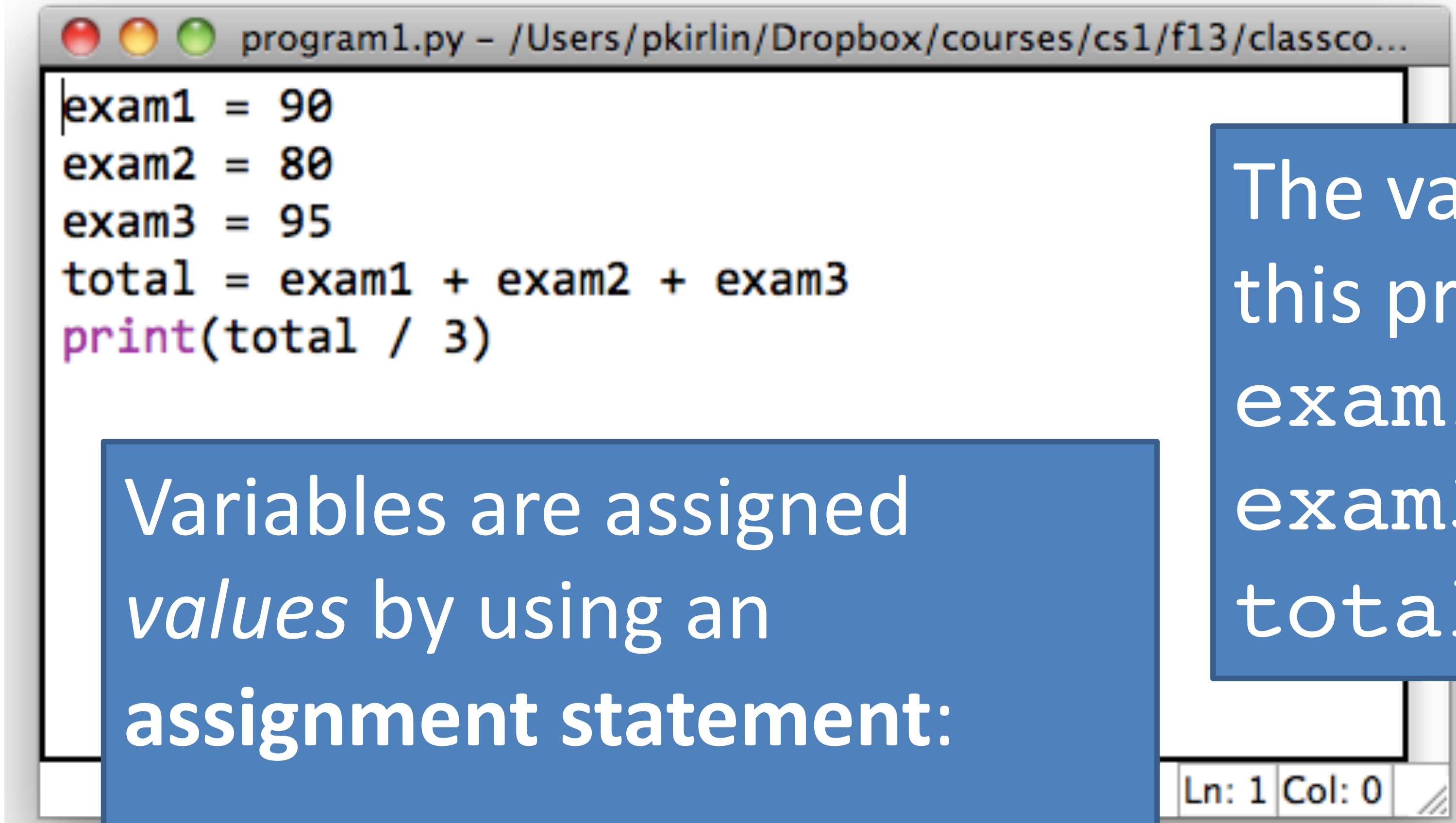
Longer Programs

- Code doesn't run until you ask Python to run it.
- Each line executes in order, top to bottom, line by line.
- Lets you run the code over and over without retyping.
- Nothing is automatically printed; to do so you must call the print function.

Math

- +   -   *   /
- ** → exponents
- // → integer division
- % → remainder
- Normal order of operations.
- Use parentheses to change order of operations.

Variables


program1.py - /Users/pkirlin/Dropbox/courses/cs1/f13/classco...
Ln: 1 Col: 0

```
exam1 = 90
exam2 = 80
exam3 = 95
total = exam1 + exam2 + exam3
print(total / 3)
```

The variables in this program are `exam1`, `exam2`, `exam3`, and `total`.

Variables are assigned *values* by using an **assignment statement:**

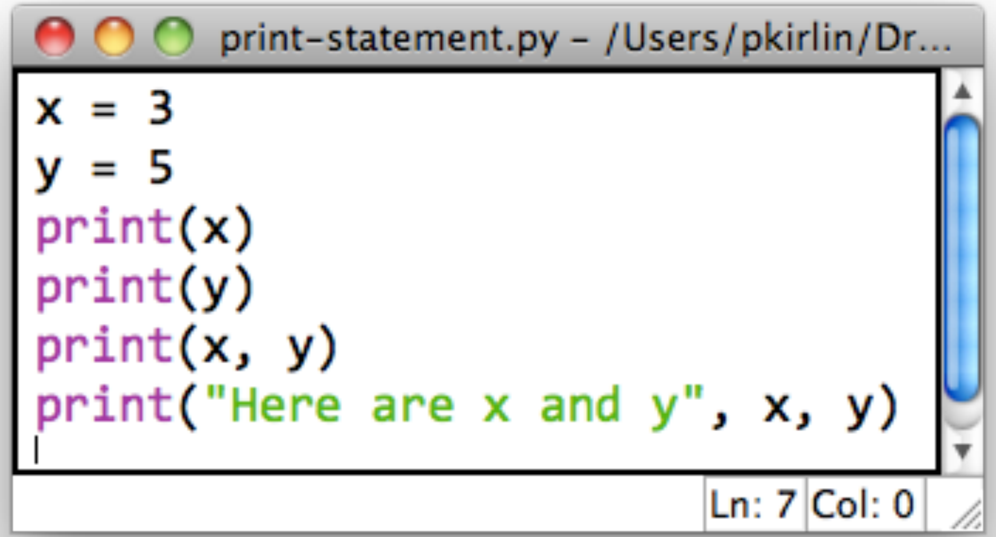
`variable = value`

Print function

- In a "real program" (not the Python Shell), nothing is displayed when you run the program unless you ask.
- Use the print function to do so.

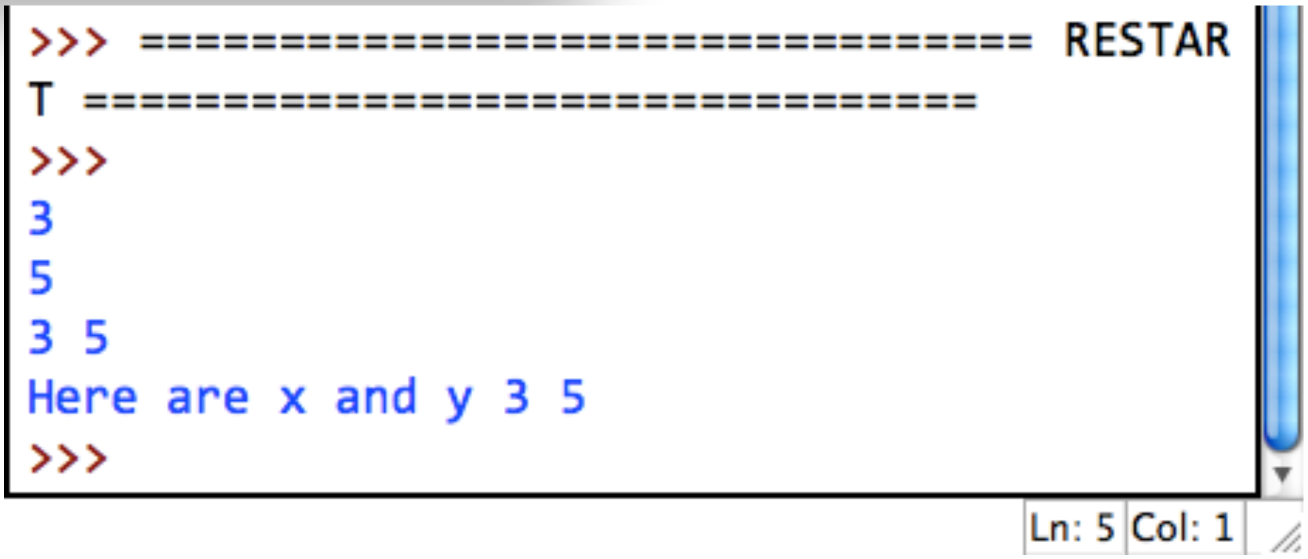
`print(_____, _____, _____, …)`

- Replace the blank spaces above with the name of a variable, or a math expression.
- You can print any number of things at once.
  - Separate each thing you want to print with a comma.
  - Each thing will be displayed with a space in between.
  - If you want to print words, surround the words with double quotes.


print-statement.py - /Users/pkirlin/Dr...

```
x = 3
y = 5
print(x)
print(y)
print(x, y)
print("Here are x and y", x, y)
```

Ln: 7 Col: 0



```
>>> ================================ RESTAR
T ================================
>>> 
3
5
3 5
Here are x and y 3 5
>>> 
```

Ln: 5 Col: 1

```
x = 3
print(x)
x = 6
print(x)
```

Computer Memory

Program Output

```
x = 3
print(x)
x = 6
print(x)
```

<u>Computer Memory</u>

Name	Value
x	3

<u>Program Output</u>

```
x = 3
print(x)
x = 6
print(x)
```

Computer Memory

Name	Value
x	3

Program Output

```
3
```

```
x = 3
print(x)
x = 6
print(x)
```

Computer Memory

Name	Value
x	6

Program Output

```
3
```

```
x = 3
print(x)
x = 6
print(x)
```

Computer Memory

Name	Value
x	6

Program Output

```
3
6
```

```
a = 4
b = 5
print(a, b)
a = 3
b = a
print(a, b)
a = b + 1
a = a + 1
print(a, b)
```

```
a = 1
b = 2
a = b
b = a
print(a, b)
```

- Variable names must be all one word (no spaces).
- Must consist of letters, numbers, or _.
  - Start with a letter.
- Choose a name that indicates the meaning of the variable.
  - For your grade on an exam: good ideas: `exam`, `exam_score`, `grade`,
  - Bad ideas: `e`, `g`, `the_score_i_got_on_the_exam`

- You're working at a fast food restaurant where a burger costs $3.99 and French fries cost $1.99.
- Write a program (in a separate file, saved as `burger.py`) that uses two variables to store these two prices.
- Your program should then print out the cost of buying two burgers and three orders of fries.
- If you finish early, make your program add in 9.25% sales tax.

Data types

- Integers (`int`)
  - Whole numbers; may be negative.
- Floating point numbers (`float`)
  - Any number with a decimal point; may be negative.
- Strings
  - Any sequence of letters, numbers, or punctuation.
  - String literals are always surrounded by quotation marks, single or double.

Input from the keyboard

- Use a variation of a variable assignment:
- For integers:

```
variable = int(input("Prompt"))
```

- For floats:

```
variable = float(input("Prompt"))
```

- For strings:

```
variable = input("Prompt")
```

- You're working at a fast food restaurant where a burger costs \$3.99 and French fries cost \$1.99.
- Write a program (in a separate file, saved as `burger.py`) that uses two variables to store these two prices.
- **CHANGE**: Make your program ask the user for how many burgers and orders of fries they want, and print the total cost.
- If you finish early, make your program ask the user for the costs of a burger and fries, and the sales tax rate.