

# **Formatting for Printing, Named Tuples, Random Numbers**

**CS 8: Introduction to Computer Science, Spring 2019  
Lecture #11**

Ziad Matni, Ph.D.  
Dept. of Computer Science, UCSB

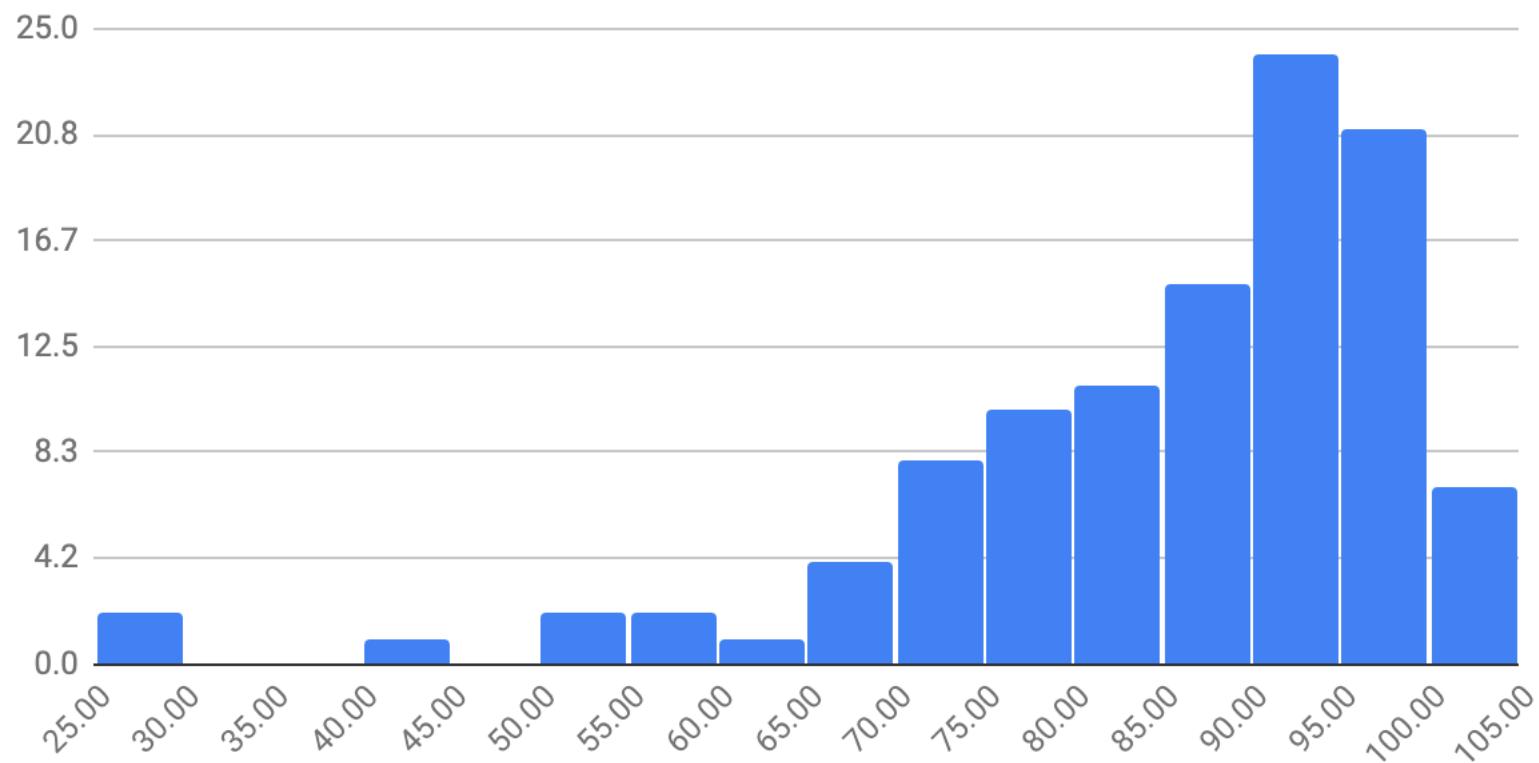
# Midterm #1 Results

---

- I had 106 respondents – that's just over 80% of you
- Congrats – everyone gets +2 points extra credit on ME1!
- Thank you! You have given me valuable feedback  
and I will incorporate it as appropriate for next half

## CS 8, Sp 19 Midterm Exam #1 Distribution

Av. = 84.7    Median = 89



# Reviewing Your Midterm #1 Exam

---

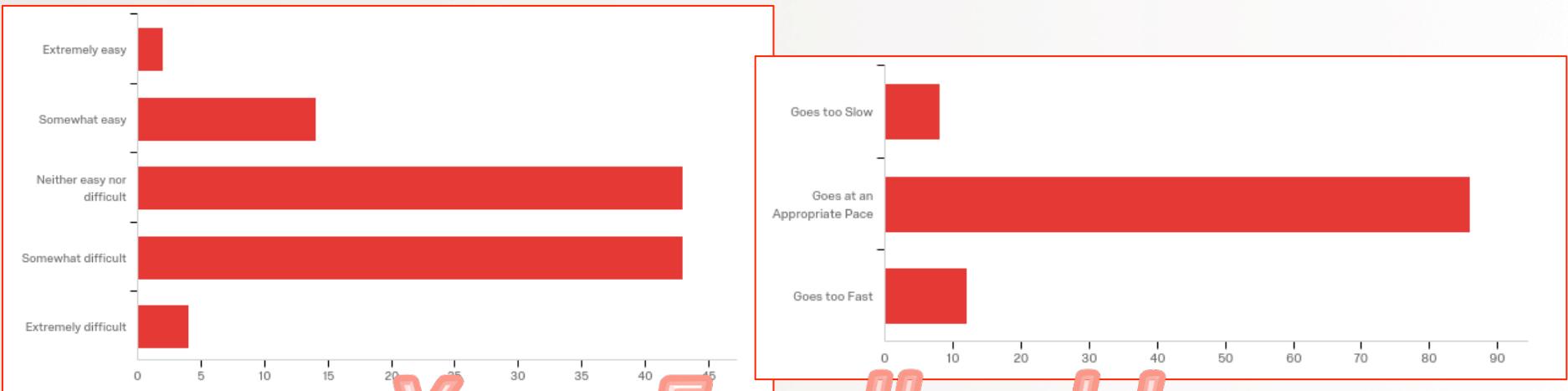
- Optional, but recommended for you to understand your mistakes
- If you're in the **8 AM** lab – go to **Chong Liu's** office hours
- If you're in the **9 AM** lab – go to **Brian Young's** office hours
- If you're in the **10 AM** lab – go to **Shane Masuda's** office hours
- If you're in the **11 AM** lab – go to **Prof. Matni's** office hours

## When Reviewing Your Exams (**IMPORTANT!**)

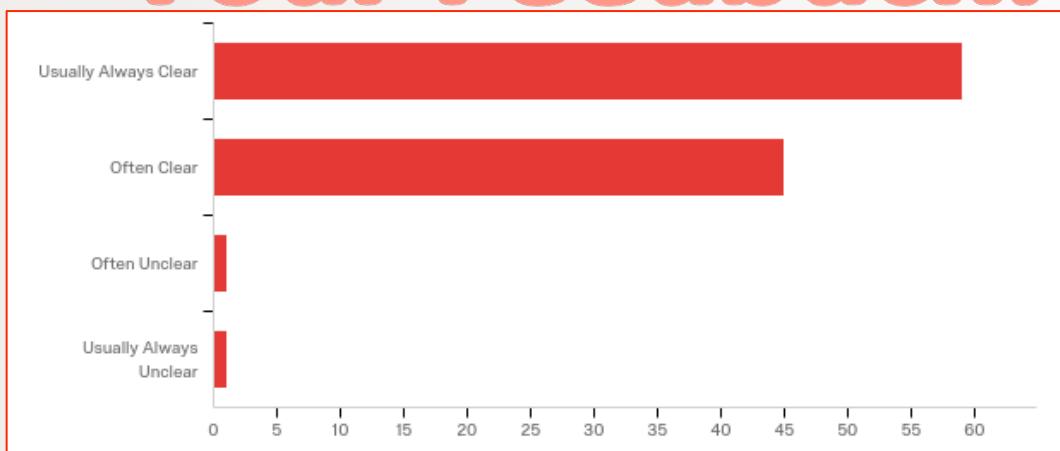
---

- Do **not** take pictures, do **not** copy the questions
- You can **only** view the exam during office hours
- You **cannot** take the exam with you
- TA cannot change your grade
  - If you have a legitimate case for grade change, the prof. will decide
  - **Legitimate case** = When we graded, we added the total points wrong
  - **Not legitimate case** =  
“Why did you take off  $N$  points on this question????”

CS8 OPEN LABS (i.e. Office Hours) - PHELPS 3525						
Day of Week	Start Time	End Time	TA On Duty	Mentors on Duty	Mentors on Duty	Mentors On Duty
MONDAY	5:00 PM	5:30 PM		Jacqueline Mai		
	5:30 PM	6:00 PM		Jacqueline Mai		
	6:00 PM	6:30 PM		Jose Cuellar		
	6:30 PM	7:00 PM	Brian Young	Jose Cuellar		
	7:00 PM	7:30 PM	Brian Young	Jose Cuellar		
	7:30 PM	8:00 PM		Jose Cuellar		
	8:00 PM	8:30 PM		Zhao Sqi		
	8:30 PM	9:00 PM		Zhao Sqi		
TUESDAY	7:00 PM	7:30 PM	Brian Young	Zhao Sqi	Daniel Shu	Jacqueline Mai
	7:30 PM	8:00 PM	Brian Young	Zhao Sqi	Daniel Shu	Jacqueline Mai
	8:00 PM	8:30 PM		Zhao Sqi		Jacqueline Mai
	8:30 PM	9:00 PM		Zhao Sqi		Jacqueline Mai
WEDNESDAY	7:00 PM	7:30 PM	Shane Masuda	Jackson Shao	Jose Cuellar	
	7:30 PM	8:00 PM	Shane Masuda	Jackson Shao	Jose Cuellar	
	8:00 PM	8:30 PM	Shane Masuda			
	8:30 PM	9:00 PM	Shane Masuda			
THURSDAY	7:00 PM	7:30 PM	Chong Liu	Jackson Shao	Daniel Shu	
	7:30 PM	8:00 PM	Chong Liu	Jackson Shao	Daniel Shu	
	8:00 PM	8:30 PM	Chong Liu	Jackson Shao	Daniel Shu	Jacqueline Mai
	8:30 PM	9:00 PM	Chong Liu	Jackson Shao	Daniel Shu	Jacqueline Mai



# Your Feedback!



- All good
- This class is very interesting
- Really knows how to teach!!
- Good sense of humor keeps class interesting
- I think the class works

## Selected Quotes

- Good use of example problems
- I think the lectures are very structured and organized
- Examples are very helpful!
- Running live demos in class really helps.
- Working the coding on the board helps me a lot

### PRACTICE PROBLEMS

- I think the examples are always easier than the real hw or exam
- The only thing I feel isn't working for me is an inadequate number of practice problems for us to try.
- I think giving us some suggested optional practice might be helpful for those who want more practice

### LABS

- Lab requires a lot of time even (if) I should figure it out... (in) section.
- The labs are very wordy
- I recommend giving more specific instructions on the lab assignments
- Solutions for labs?

5/9/19

### MIDTERM

- I felt like the midterm had a few too many questions
- The midterm was easy

## *Selected Quotes*

### OTHER

- I'm too nervous I'll get called on
- I like that participation isn't encouraged during lectures
- I think you could be a little harsh sometimes
- I would love to see all the code we try in class post on the class website

# Administrative

---

- Homework #5 – due on Tuesday
- Lab04 – due on Sunday by midnight (11:59 pm) on **Gradescope!**
- **Project Lab description is now up!**
  - Project counts as 2 lab grades
  - Due at the end of the quarter (June 2<sup>nd</sup>)
- Midterm Exam #1 Grades are now up!
- Midterm Exam #2 is on **May 23<sup>rd</sup>**
- There will **NOT** be a lecture on **Thursday, May 16<sup>th</sup>**

# Lecture Outline

---

- Using the **format()** function
- The **namedtuple** data type
- Random numbers
- File Input / Output

# str.format()

## to Format Multiple Variables Into a String

- You can print an output while you **define** your general format!

Example:

```
hour = 12  
minute = 55  
second = 31
```

*Note: the {0} refers to hour (the 0<sup>th</sup> argument),  
the {1} to minute (the 1<sup>st</sup> argument), etc...*

**THIS ORDER MATTERS!!**

*Example, what would happen  
if I switched {0} and {1} in here?*

If you do this: `print('{0}:{1}:{2}'.format(hour, minute, second))`

You get this: **12:55:31**      *(it's a string output)*

## str.format()

### to Also Format the Use of Space In a String

- You can define how many spaces an object occupies when printed

Example:

```
>>> a = 19  
>>> b = 42  
>>> print('{0:3}xyz{1:5}'.format(a, b))  
' 19xyz    42'  
      3           5  
      spaces     spaces
```

Refers to the 0<sup>th</sup> item (that is, variable *a*)

Refers to the total number of spaces you want to format

Let's try it out!

## str.format()

### to Also Format the Use of Space In a String

- With strings instead of numbers

Example:

```
>>> a = "Be"  
>>> b = "Mine!"  
>>> print('{:7}{:>7}'.format(a, b))
```

'Be            Mine!'  
  7            7  
  spaces        spaces

Save 7 spaces for var. **a** and **left justify a**  
Put any extra spaces AFTER it

Save 7 spaces for var. **b** and **right justify b**  
Put any extra spaces BEFORE it

**What happens if you run out of space?**  
Does it:

- cut out the string to make it fit?
- still print out the string even if it's longer than the space format?

## str.format()

### to Format Floating Point Numbers In a String

- If you say, `print(100/3)`, you get: 33.33333333333336
- What if you wanted to instill some precision on your decimal values?

Example:

```
>>> n = 100/3  
>>> print('{:7.3f}'.format(n))  
' 33.333'  
    7  
    spaces
```

*Save 7 spaces for the floating point.  
Put 3 numbers after the decimal point*

Let's try it out!

# More Examples

---

- Go to your textbook and read through all the examples in Ch. 4.2
- There are other types of format
- CHECK THOSE OUT TOO!!!



# Tuples vs. `namedtuple()`

---

- The standard **tuple** uses numerical indexes to access its members
  - Like lists or strings do

- Example:

```
bob = ('Bob', 30, 'male')
```

```
print ("Bob's age is:", bob[1])
```

# Downside: I have to remember that the age is index 1

# namedtuple()

- We can now give the indices more relevant semantics *not just a number!*
- **namedtuple()** is a CLASS defined in the library **collections**

```
import collections
Person = collections.namedtuple('Person', 'name age gender')
bob = Person(name='Bob', age=30, gender='male')
print(type(bob))
print("The whole thing:", bob)

jane = Person(name='Jane', age=29, gender='female')
print("Name:", jane.name)
```

Let's try it out!



# Random Numbers

---

- “Pseudo-random” values can be generated using special functions in most programming languages
- In Python use functions of the **random module**
  - You have to *import random* first
- Simplest way to make a random number: **random.random()**
  - Returns a floating point value between 0.0 and 1.0

# Random Numbers

---

- Also: **randrange(n)**, **randint(low, high)** and many others
  - **randrange(n)** returns int random number between 0 and n-1
  - **randint(low, high)** returns int random number between low and high (inclusive)
- Try typing **help(random)** in IDLE to learn more...
  - And play around with it

# Question 1

---

Q: What is a Python statement that generates a number between 0 and 100  
**(including** floating point values like 55.5)

Assume I issue a statement at first, like this:

```
from random import *
```

- A. random() + 100
- B. random()\*100 
- C. random()/100
- D. random(100)

## Question 2

---

Q: What is a Python statement that generates a INTEGER between 50 and 100 (not inclusive). Assume you have the correct import statements...

- A. `random() * 50`
- B. `50 + int(random() * 50)`
- C. `randrange(50, 101)` 
- D. Both B and C do this
- E. All of A, B, C

# YOUR TO-DOS

---

- Homework #5 due **Tuesday, 5/14**
- Finish **Lab4** (**turn it in by Sunday**)
- Remember that next week Thursday (5/16), there's NO lecture
  
- Know that *for time in range(your\_life):  
    if yin = math.sin(time)  
        and yang = math.cos(time):  
            yin\*\*2 + yang\*\*2 == 1*

</LECTURE>