

Mathematics and Computation

COMPSCI 230 --- Discrete Math

January 12, 2017

Outline

- 1 What and Why
 - Math
 - Code
 - Writing
- 2 Practicalities
 - Lectures and Recitations
 - Homework
 - Exams and Grades
 - Academic Integrity

On The Web

- <http://www.cs.duke.edu/courses/spring17/compsci230/> for course materials
- [Piazza](#) for communication
- Homework:
 - [Gradescope](#) for text (.pdf)
 - [Sakai](#) for code (.py) (do *not* use the Sakai dropbox!) and grades
 - Much more about homework submission later

Flavor of the Course

- Discrete math ...
 - Sets, functions, combinatorics, proofs, graphs, probability, computation
- ... about and through code ...
 - Code in Python 3
- ... with emphasis on clarity
 - Write in \LaTeX to make PDF files
 - Much more about \LaTeX later

What is this Math For?

- To build models
- To understand computation
- To think clearly
- Used in all aspects of CS



The Imitation Game

Why Code?

- To see ideas at work
- To force clarity: The factorial of a natural number n is

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n(n-1)! & \text{if } n > 0 \end{cases}$$

```
def factorial(n):  
    assert(type(n) == int and n >= 0)  
    return 1 if n == 0 else n * factorial(n-1)
```

- To convert formulas to numbers

50! = 3041409320171337804361260816606476...
...88443776415689605120000000000000

Why Python?

- Feels familiar, even if you don't know the language
- Keeps the focus on the math
 - We will help you
 - We will use a subset of the language
- However, form matters
 - Elegant code is understandable code (by you and others)
 - ``Elegant:`` As succinct as possible, but not more succinct

Prolix Code

```
def factorialFunction(integer):
    if type(integer) != int:
        print("The input is supposed to be an integer but is'", end = ' ')
        print(integer, "' instead", end = '\n')
        raise TypeError
    if integer < 0:
        print("The input must be nonnegative but is'", end = ' ')
        print(integer, "' instead", end = '\n')
        raise ValueError

    def factorial(value):
        if value == 0:
            result = 1
        elif value > 0:
            value = value - 1
            phi = factorialFunction(value)
            value = value + 1
            result = value * phi
        else:
            print('value is neither zero nor positive, but is equal to', end = ' ')
            print(value, ' instead', end = '\n')
        return result

    output = factorial(integer)
    return output
```


Overly Succinct and Hard-to-Read Code

```
def f(n): return n and n*f(n-1) or 1
```

- Python evaluates left-to-right: $(n \text{ and } n*f(n-1)) \text{ or } 1$
- **A and B**: If **A** is **False**, return **A** and ignore **B**. Otherwise return **B**
- **C or D**: If **C** is **True**, return **C** and ignore **D**. Otherwise return **D**
- For Python, an integer is **False** iff it is zero
- If $n == 0$, the **and** evaluates to 0 (which is **False**)
 - We are left with $0 \text{ or } 1$
 - 0 is **False**, so the **or** returns the value of 1, which is 1
- If $n > 0$, evaluate $n*f(n-1)$ to determine the value of the **and**
 - The result is a positive integer p , which is **True**, so the **and** has value p , and the **or** evaluates to p regardless of the value of 1 (which is not evaluated)
- So the line above is the same as


```
def f(n): return n*f(n-1) if n else 1
```
- [Chaos still ensues from the call $f(-1)$]

Goldilocks Code

```
def factorial(n):  
    assert(type(n) == int and n >= 0)  
    return 1 if n == 0 else n * factorial(n-1)
```

Why L^AT_EX?

- *Writing* math is an important part of doing math. Math is all about clarity, on paper and in your head
- Form matters: Complicated or obscure math is useless
- Everybody does L^AT_EX in math or CS: papers, proposals, books, theses, ...
- Your textbook and these slides were written in L^AT_EX
- L^AT_EX is not just a writing tool, it's a thinking tool

Your Homework Will Look Like This

COMPSCI 230

Carlo Tomasi

Homework 0

Solution to Problem 1.1 (tag: factorial)

The factorial of a natural number n is denoted by $n!$ and is defined as follows:

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ 1 \cdot 2 \cdot \dots \cdot n & \text{if } n > 0 \end{cases}$$

This definition is equivalent to the following recursive definition:

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n(n-1)! & \text{if } n > 0 \end{cases}$$

The following Python definition is a straightforward translation of the recursive definition above:

```
def factorial(n):
    assert (type(n) == int and n >= 0)
    return 1 if n == 0 else n * factorial(n-1)
```

Lectures and Recitations

- Lectures on Tuesday, Thursday
 - Reading assignments **before** class
 - **Graded** iClicker quizzes in class
 - Also ungraded quizzes
- Recitation on Monday
 - More detail, examples, Q&A
 - Bring your laptop
- Attendance **required** at both
 - 5% of the grade comes from recitation attendance

Why Clickers?

- Non-graded questions
 - Engage you
 - Keep you listening
 - Solidify your understanding
 - Let you talk with your peers
- Graded questions
 - Motivate you to read *before* class ...
 - ... and *before* you do the homework

Register your Clicker Online

- You must register your clicker properly, or you will get no clicker credit
- Register your physical clicker at <https://www1.iclicker.com/register-clicker/>, or
- Register your REEF app (for smartphone, tablet, PC) at <https://app.reef-education.com/#/login>
- **IMPORTANT: Your student ID is your full Sakai email.**
Example: xyz53@duke.edu, including the @duke.edu
- If you get a used device, you still need to register it
- Fees involved either way

Clicker Exercise

What clicker do I have?

(Not Graded)

A: I have a physical clicker

B: I have a clicker app

C: I do not have a clicker (?)

Homework Rules

- \approx one homework assignment per week
- \LaTeX \rightarrow PDF, submit on [Gradescope](#)
- Often also `.py`, submit on [Sakai](#)
- More detailed submission instructions will be given
- Due by start of class on the due date
- (–) No late homework accepted
- (+) Single worst homework grade ignored
- Work alone or in duos or in trios on your homework
 - Groups can be different every time
 - **No** communication outside your group
 - No effort subdivision

Exams and Grades

- One midterm, one final (dates on class page)
- Grades:
 - 40% homework
 - 25% midterm, 25% final
 - 5% iClicker, 5% recitation attendance
- (–) Late homework gets zero
- (–) No clicker excuses
- (+) Single worst homework grade ignored
- (+) One point for every graded clicker answer, or 3 if correct
- (+) Worst 3 iClicker grades ignored (one lecture, one grade)

Academic Integrity

***Using anyone else's work
without giving credit is cheating***

- Cheating examples:
 - Looking up how to solve a problem online
 - Copying any part of someone else's homework
 - Divvying up homework
 - Discussion of homework other than by rules
 - Using code or other materials from anywhere (other people, web, books,...) without full, detailed credit
 - ``Using'' is not just literal use
 - Giving or receiving help during an exam
- Active and passive role are equally bad

Clicker Exercise

I work with Sam on a COMPSCI 230 homework assignment. I do the math, he does the coding.

Are we cheating?

(Not Graded)

A: Yes

B: Only if our amounts of work are different

C: No

Clicker Exercise

A homework problem asks to write code to compute the power set of a set. In class we covered only the concept of power set, not algorithms for it. I Google "power set" and read the Wikipedia page on the topic. There is a link to *power set algorithms* on the page.

Am I cheating?

(Not Graded)

A: Yes

B: If and only if I click on that link

C: No

Clicker Exercise

A homework problem asks to write code to compute the power set of a set. In class we covered only the concept of power set, not algorithms for it. I Google “power set algorithm” and read the Wikipedia page on the topic.

Am I cheating?

(Not Graded)

A: Yes

B: No, if I don't Google for Python code

C: No

For Next Time

- Remember the readings (Python, \LaTeX)
- Graded questions on the readings will be asked
- Clicker (or app) is mandatory, no excuses
- Homework 1 will come out by Jan 19 and will be due on Jan 26. Start early as you need to install and be familiar with Python 3 and \LaTeX
- Exercise: Make a \LaTeX file that produces a PDF page similar to slide 12. (Don't worry about the headers.) Also program `factorial` and try it out