Compsci 101
Introduction

This is a temporary schedule for what we will do, subject to change!
Information about grading and course logistics.

<table>
<thead>
<tr>
<th>January 3-7 Week</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRE-WORK</td>
<td>1/3</td>
<td>1/4</td>
<td>1/5</td>
<td>1/6</td>
<td>1/7</td>
</tr>
<tr>
<td>LECTURE/LAB</td>
<td>1/8</td>
<td></td>
<td></td>
<td>1/9</td>
<td>Lab 0</td>
</tr>
<tr>
<td>ASSIGNMENTS/QUIZ</td>
<td>1/10</td>
<td></td>
<td></td>
<td>1/11</td>
<td>Assign it out</td>
</tr>
</tbody>
</table>

Susan Rodger
January 6, 2022

Sea turtles hatching

• https://www.youtube.com/watch?v=sZYd0-egqbc

About Prof. Rodger

How do you keep your sanity?

Prof Rodger
Ms. Yesenia Velasco

Teaching Associate

- Right-hand woman - Handles logistics, substitute lectures, and much more!

- Handles accommodations
  - Email her your accommodation letter
  - yvelasco@cs.duke.edu

**Announcements**

- Check out the calendar on the course website
  - PRE-WORK – what you must do before the lecture
  - LECT/LAB – will put notes/videos here from the live lecture or for the lab
  - DUE – what is due each week.

- What has been updated?

- Assignment 0 is already out!
- Lab 0 on Friday
- Prelab for Lab1 (install Python)
- You will see a link to this video!

**A is for …**

- Algorithm
  - Step-by-step instructions realized in a program

- Abstraction
  - Hiding things is powerful
  - “What” vs “How”

- APT
  - Algorithmic Problem-solving Testing

- API
  - Application Programming Interface - using Libraries

**Go over CompSci 101 webpages**

CompSci 101: Introduction to Computer Science

Course Description

Introduction to practices and principles of computer science and programming and their impact on and potential to change the world. Algorithmic, problem-solving, and programming techniques in domains such as art, data visualization, mathematics, natural and social sciences. Programming using high-level languages and design techniques emphasizing abstraction, encapsulation, and problem decomposition. Design, implementation, testing, and analysis of algorithms and programs. No previous programming experience required. For this version of the course, you will learn the programming language Python 3.
Course overview, logistics
www.cs.duke.edu/courses/spring22/compsci101

- Programming assignments: APTs and Assignments
  - Acknowledge assistance, to learn to program …
  - Be aware of late policy
- Labs
  - Attend each Friday
- Lecture - Classwork
  - Attend the live lecture - participate
  - If you can't attend you must watch it and participate within 24 hours
- Exams: midterms and final
  - All old midterms available

What's in Compsci 101?
- Learning about computing, computer science, and programming
  - Vocabulary of Python and programming
  - Power of automation, repetition, scale
  - Understanding and changing the world
- Programming using Python
  - Tools: PyCharm, Libraries, …
  - Using mathematical and scientific techniques
  - Art and science of programming

Questions?
- Don't send us email!!!!!!!!!!!!
- Post your questions on Ed Discussion
  - We will answer them there!
  - You should try to answer them too
    - Want to be a UTA one day? Answer questions!
- Post Questions during lecture!

WOTO – Working Together
(breakout rooms in zoom)

Discuss with others, then everyone fills out their own form.

What is Computer Science?
Computer Science

Vacuum Tubes

- Control electric current using the vacuum
- Can be used to start/stop, or change the flow based on the current
  - Off/On → 0/1
  - 00000011

How it started

1906

What is Computer Science?

- Artificial Intelligence
  - Roomba
  - Spirit, Mars Rover
  - Self-driving car
  - Personal Robot
What is Computer Science?

- Medicine, Genomics

What is Computer Science?

- Animation

Prerequisites for Compsci 101

Who has taken CompSci 101?
Why is it called Python?

Who are you?

- Let’s look at survey to see who is taking CompSci 101 in Spring 2022
  - Do you recognize yourself?
  - Is there a stereotypical CompSci 101 student?
  - Is there a stereotypical computer scientist?

- Everyone can succeed! Ideally you won’t have lots of experience programming
What does this program do?

- "Hello World"
- Scratch Program
- Colors
  - Duke blue: motion
  - Mustard: control
  - Light blue: sensing
  - Orange: data
  - Purple: looks

Scratch Program

- If you want to experiment with this scratch program, here is the link:
  
  https://scratch.mit.edu/projects/94064630/

  You don’t have to understand this yet!!

WOTO: WOrking TOgether


Analyze this Scratch Program?

What language will we learn?

- http://www.python.org/
- Python is a multi-paradigm language
  - Procedural
  - Functional
  - Object-Oriented
- Simple, libraries, widely used
- Guido von Rossom
Python code
hello.py

```python
Created on 1/6/2022
@author: Susan H. Rodger

if __name__ == '__main__':
    print("Hello CompSci 101!")
```

OUTPUT:

```
C:\Users\Susan\AppData\Local\Programs\Python\Python 3.11\Scripts\hello.py
Hello CompSci 101!
Process finished with exit code 0
```

Python Code, second program

```python
def greeting(name):
    print("Hello " + name)

if __name__ == '__main__':
    greeting("CompSci 101!")
    greeting("Beenie, Keeah and Moe")
```

OUTPUT:

```
C:\Users\Susan\AppData\Local\Programs\Python\Python 3.11\Scripts\hello.py
Hello CompSci 101
Hello Beenie, Keeah and Moe
```

You don't have to understand this yet!!
Duke Connection: Fred Brooks '53

• What Would FB Say?
"The most important single decision I ever made was to change the IBM 360 series from a 6-bit byte to an 8-bit byte, thereby enabling the use of lowercase letters. That change propagated everywhere."

Why is programming fun?

Fred Brooks

• First is the sheer joy of making things
• Second is the pleasure of making things that are useful
• Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts
• Fourth is the joy of always learning
• Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff.