What Next?
Python, Java, CSE Courses

Ruth Anderson
UW CSE 160
Autumn 2021
We want your feedback!

• Please fill out evaluations for lecture AND for section
  – The link for lecture is here.
There is more to learn!

• You have come a long way from the first day of class!
  – But there is more to learn!
• Data analysis, data science, and data visualization
• Scaling up:
  – Larger and more complex programs
  – Algorithm selection
  – “Big data”: out-of-memory data, parallel programming, ...
• Ensuring correctness
  – Principled, systematic design, testing, and programming
  – Coding style
• Managing complexity
  – Programming tools: testing, version control, debugging, deployment
  – Graphical User Interfaces (GUIs), user interaction
  – Data structures and algorithms
  – Working in a team
More UW Computer Science Courses!!

You could take any of these now!

• [22wi & 22sp] **CSE 163** Intermediate Data Programming

• [every quarter + summer] **CSE 142, 143, 143x** Programming in Java (143x only in fall)

• [22wi & 22sp] **CSE 154** Web Programming

• [22sp] **CSE/STAT 416** Intro to Machine Learning (requires Stat 311/390)

• [every quarter] **INFO/STAT/CSE 180** Intro to Data Science (some Math pre-req)

Require **CSE 143**:

• [every quarter] **CSE 373** Data Structures & Algorithms (all year)

• [22sp] **CSE 412** Intro to Data Visualization (requires CSE 143 or CSE 163)

• [22wi & 22sp] **CSE 414** Databases

• [22wi & 22sp] **CSE 374** Intermediate Programming Concepts & Tools

Require **CSE 373**:

• **CSE 410** Computer Systems (Operating Systems & Architecture)

• **CSE 413** Programming Languages and their Implementation

• **CSE 415** Artificial Intelligence

• **CSE 417** Algorithms and Complexity
More Info on UW CSE Courses!!

• Which Course should I take:

• Intro CSE courses:
More Python Resources

• More Python practice:

• Runestone – free interactive textbooks:
  – How to Think Like a Computer Scientist
    (the “Try” text we used this quarter)
    https://runestone.academy/runestone/books/published/thinkcspy/index.html
  – Problem Solving with Algorithms and Data Structures using Python
    https://runestone.academy/runestone/static/pythonds/index.html
Why the Python language?

<table>
<thead>
<tr>
<th></th>
<th>Python</th>
<th>Excel</th>
<th>MATLAB</th>
<th>R</th>
<th>C/C++</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>Readable syntax</td>
<td>😊</td>
<td>😑</td>
<td>😑</td>
<td>😑</td>
<td>😑</td>
<td>😊</td>
</tr>
<tr>
<td>Easy to get started</td>
<td>😊</td>
<td>😊</td>
<td>😊</td>
<td>😑</td>
<td>😑</td>
<td>😑</td>
</tr>
<tr>
<td>Powerful libraries</td>
<td>😊</td>
<td>😊</td>
<td>😊</td>
<td>😊</td>
<td>😊</td>
<td>😊</td>
</tr>
</tbody>
</table>
Comparison of Python with Java

• Python is better for learning programming
• Python is better for small programs
• Java is better for large programs

Main difference: dynamic vs. static typing
• Dynamic typing (Python): put anything in any variable
• Static typing (Java):
  – Source code states the type of the variable
  – Cannot run code if any assignment might violate the type