CSE 121 Lesson 9:

While Loops

Simon Wu

Summer 2024

TAs:	Trey	Hannah	Mia	Vivian	Jolie	Colton	Ziao

sli.do #cse121

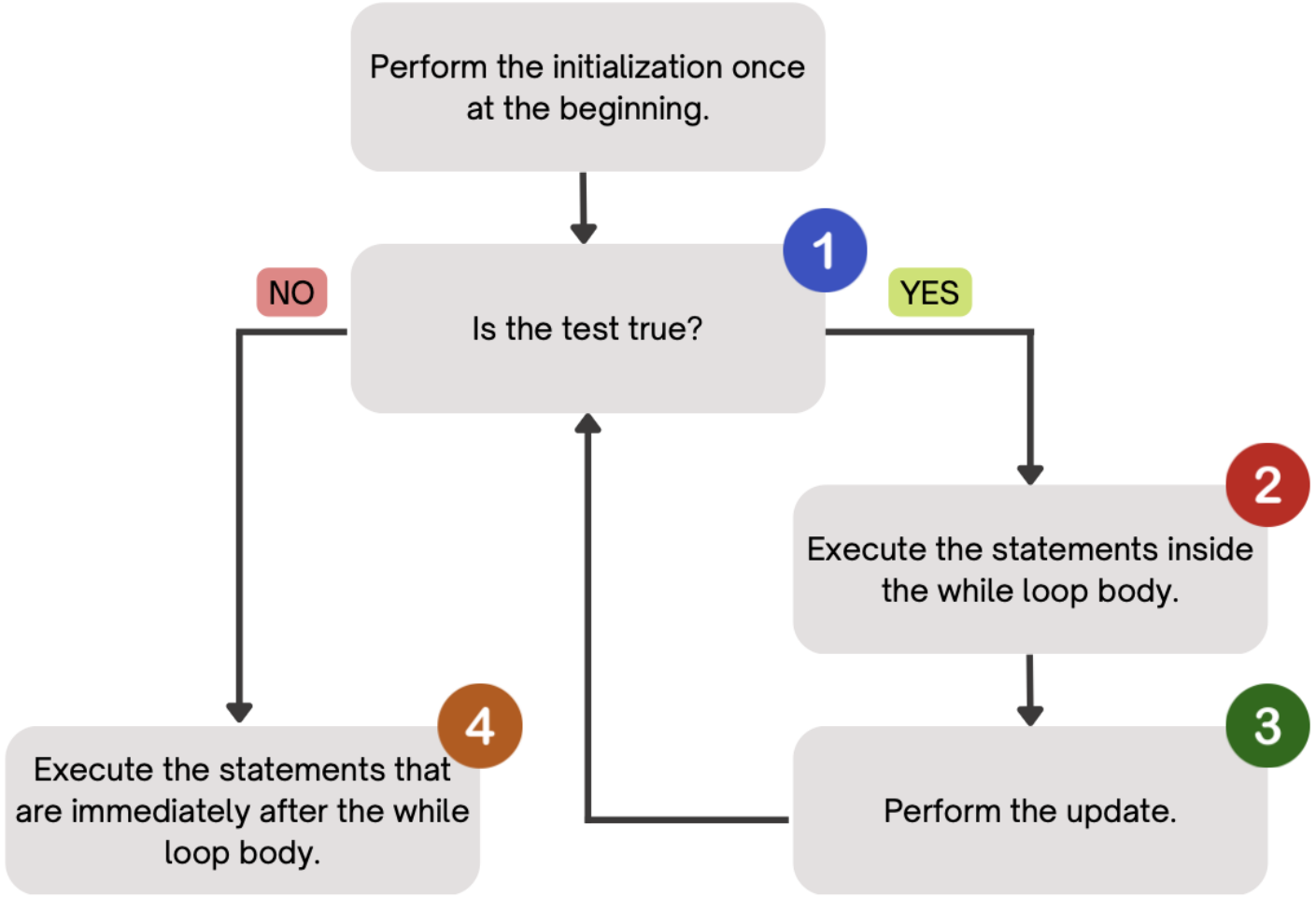
Announcements

- P1 Deadline **extended** to next Tuesday, 7/23 at 11:59 pm
- Resubmission Cycle 2 (R2) due tomorrow (7/20) at 11:59 pm
  - Eligible: **C0**, P0, C1
  - R3 opens tomorrow, due next **Thursday** (7/25); eligible: **P0**, C1, P1
- Programming Assignment 2 releases next Wednesday (7/24), due the following Tuesday after (7/30) at 11:59 pm
- Quiz 1 next week in section(7/25); topics include up to **conditionals**
  - more practice materials & resources coming soon!
  - Quiz 0 feedback to be released later tonight

(PCM) While Loops

```
while (test) {
    body (statements to be repeated)
}
```

Repeatedly executes its body **as long as** the logical test is true.



`for` loops vs. `while` loops ⚔️

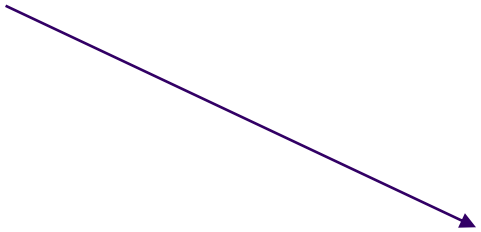
For loops and while loops are quite similar! This is the first (but certainly not the last) time where you need to decide which to use!

There's not always a "correct" answer, but some advice:

- thinking of definite versus indefinite conditions
- phrasing the problem out loud!
  - "I will do __ X times" or "for each __ I will __" : sounds like a for!
  - "I will do __ until __" or "while __ is true, do" : sounds like a while!

`for` loops are `while` loops!!! (1/2)

```
for (int i = 0; i < bigYikes; i++) {
  // ...
}
```

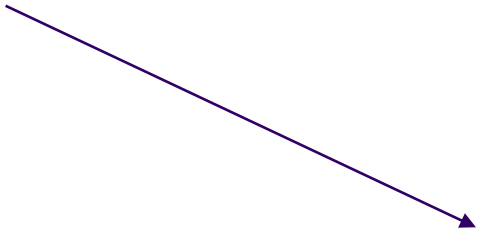


```
int i = 0;
while (i < bigYikes) {
  // ...

  i++;
}
```

for loops are while loops!!! (2/2)

```
for (int i = 0; i < bigYikes; i++) {
   // ...
}
```



```
int i = 0;
while (i < bigYikes) {
   // ...

   i++;
}
```

*as a technical note, these aren't exactly the same – there are some minor technical details that are different, most notably the scope of `i`

Poll in with your answer!

sli.do #cse121

What will be the output of this program?

```
int num = 0;

while (num < 5) {
  System.out.println("num is less than 5!");
}
```

A. num is less than 5!
   num is less than 5!
   num is less than 5!
   num is less than 5!
   num is less than 5!

B. (no output)

C. Compiler Error

D. Infinite Loop

Poll in with your answer!

sli.do #cse121

How would you describe what the variable x calculates?

```
int roll = -1; // "priming" the loop
int x = -1;
while (roll != lucky) {
  roll = randy.nextInt(sides) + 1;
  if (x < roll) {
    x = roll;
  }
}
System.out.println(roll + ": my lucky number!");
```

A. The largest value rolled
B. The smallest value rolled
C. The last value rolled
D. The first value rolled
E. The sum of all values rolled
F. Error
G. -1

Culminating Final Reflection

- Short (1-2 page) paper on a computing topic of your choice
  - Create and discuss your own reflection question
  - Must address at least **two** of the course's reflection learning objectives
  - Full rubric will be released ahead of time
- Full spec will be released early next week, due **on the last Friday of the quarter** (8/16) to Gradescope by 11:59 pm
  - **absolutely no late days allowed**
  - if submitted by 8/5, we will give you feedback by 8/12 and let you to resub
- Counted as a **single** "Exam Grade," eligible for the lowest drop

Reflection Recap

- **C0: Hello Bugs**
  - Intro to reflection: why is it important to think critically about technology?
- **P0: Cornbear Café**
  - The history of English-centric programming languages
  - Which communities are invited to participate in programming?
- **C1: Turtle Drawings**
  - Accessibility issues in coding
- **P1: Election Simulator**
  - The use and impact of technology on non-technical industries and systems
  - Should technology be used in every situation?

Reflection Learning Objectives

- Describe the ethical and social impacts of technology and explain how our choices as programmers can influence these impacts
- Challenge dominant assumptions, values, and goals reflected in computing and technology
- Analyze the strengths and limitations of using computing and technology to solve various problems
- Identify interdisciplinary applications of computing that can be deployed in service of different communities
- Understand disparities in access to computing, and explain the consequences of such disparities in technologies we build

We loved your C1 reflections!

I read (skimmed?) all 62 of your responses! Some themes:

- generally, learning about how blind people program
- “one minor addition and effort into making a program accessible can greatly impact the daily experience of those who need it”
- debugging is already hard – debugging without accessible error messages sounds even harder!

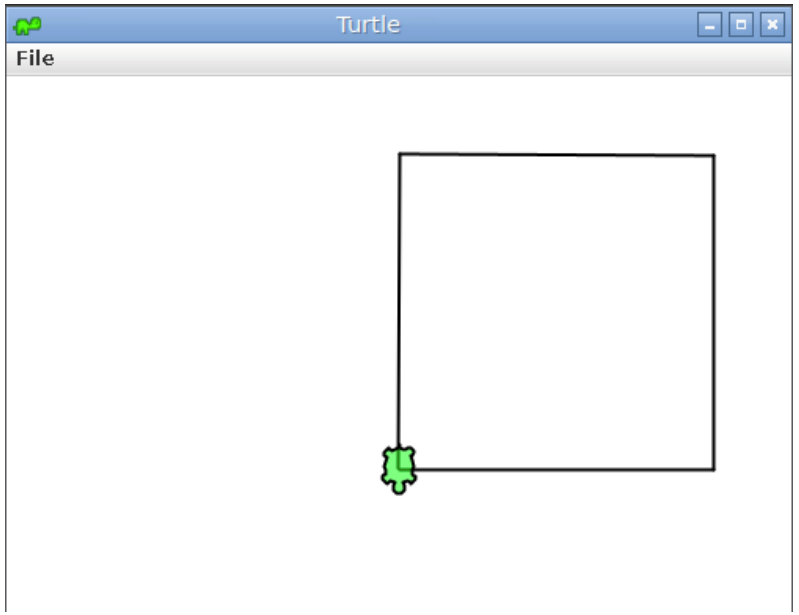
… with great points on "`accessiblePrinting`"

Printing out what the Turtle does is better than nothing.

But, y'all said:

- it is a ton of information – especially for complicated drawings
- the information provided might not be the "right" type
  (not precise, not high-level enough, not aware of shapes)
- **does not describe what the drawing actually is**

Describing images…



Which of these best describes this image?

1. A black square drawn by a Turtle
2. An image with a green cartoon turtle overlapping with a square
3. A screenshot of the Turtle library; the toolbar says “Turtle” and has a minimize, full-screen, and close buttons. The main canvas has a 200 by 200 square, drawn by a Turtle.
4. Instruction: MOVE FORWARD 200.0 Current Pos: (200.000, 0.000) …

Describing images requires context

“A black square drawn by a Turtle”

- brief overview for someone who knows Turtle, doesn’t care about the art

“An image with a green cartoon turtle overlapping with a square”

- short description focused on the core image – no Java-Turtle context assumed

“A screenshot of the Turtle library; the toolbar says “Turtle” and has a minimize, full-screen, and close buttons. The main canvas has a 200 by 200 square, drawn by a Turtle.”

- longer caption, perhaps useful in a user design textbook or case study

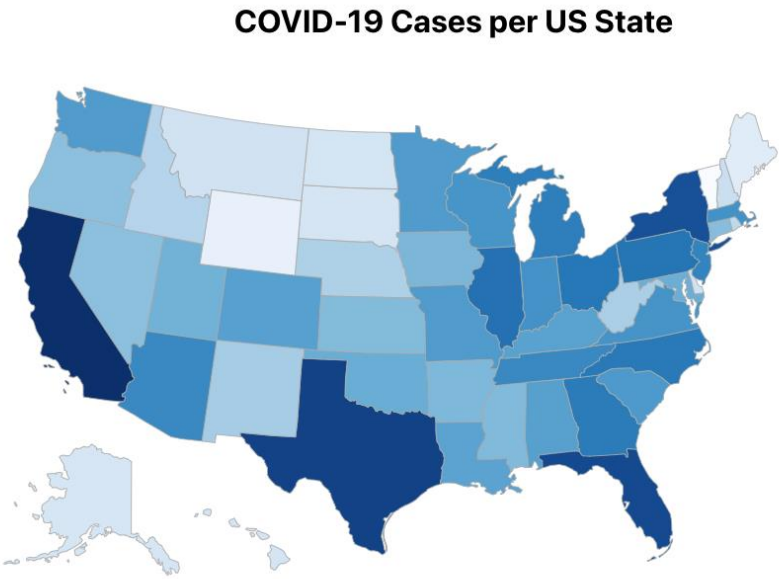
And, many other reasonable alternative texts & captions!

Active research – at UW!

Describing visualizations can be even harder!

How would you describe the visualization shown on the right?

- could read out the data as a table – 50 rows!!
- could summarize key points – loses data!
- from UW CSE + iSchool: let users decide – and ask questions about data (min, max, average)



Paper: VoxLens: Making Online Data Visualizations Accessible with an Interactive JavaScript Plug-In., Ather Sharif, Olivia H. Wang, Alida T. Muongchan, Katharina Reinecke, and Jacob O. Wobbrock. CHI '22.



Food for Thought

Interested in accessibility? UW is an amazing place to be!

- stellar professors in CSE – e.g. Jen Mankoff, who teaches CSE 493E: Accessibility
- amazing folks across campus – e.g. Jacob Wobbrock (iSchool) from the paper!
- people here do research, build tools, advocate for policy, and create community!

Many students mentioned that AI could be helpful. It's … complicated.

- "'Without these tools, I'd be lost': how generative AI aids in accessibility"
- "No, 'AI' Will Not Fix Accessibility"

Challenge Question (1/2)

We *could* attempt to improve the Turtle accessibility feature in various ways, such as using Artificial Intelligence.

However, this will come with ***tradeoffs:***

- might make the Turtle library even slower to run
- might require more resources on our computer
- might not be used by *"most"* users of the library

Challenge Question (2/2)

How do we *evaluate* code?

(i.e. how do we determine if a piece of code is “good” or not?)