CSE 121 Lesson 5: Nested for loops, Random, Math

Matt Wang & Brett Wortzman

Autumn 2024

sli.do #cse121

TAs:

Abby	Afifah	Ailsa	Alice	Aliyan	Arohan
Chloë	Christopher	Dalton	Derek	Elizabeth	Ethan
Hanna	Hannah	Heather	Hibbah	Janvi	Jasmine
Judy	Julia	Kelsey	Lucas	Luke	Mahima
Maitreyi	Maria	Merav	Minh	Neha	Ronald
Ruslana	Sahej	Sam	Samrutha	Sushma	Vivian
Yijia	Zachary				

Today's playlist:
121 24au lecture tunes

Announcements, Reminders

- Creative Project 1 is out, due Tue Oct 15th
- Resubmission Cycle 0 released, due Thu Oct 17th
  - Eligible for submission: C0 & P0
- Course logistics reminders:
  - “Extra Resources” on course website
  - Post-section work must be done by 11:59 that day
  - Brett’s office hours out now!

Last time: for loops!

For loops are our first *control structure*

A syntactic structure that *controls* the execution of other statements.

```
for ( initialization ; test ; update ) {
    body (statements to be repeated)
}
```

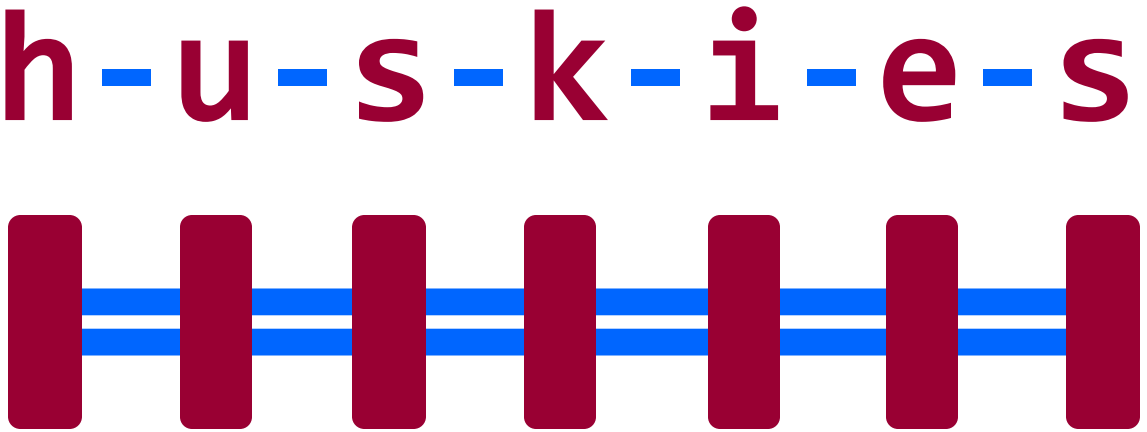
Fencepost Pattern 1

Some task where one piece is repeated *n* times, and another piece is repeated *n-1* times and they alternate

`h-u-s-k-i-e-s`

Fencepost Pattern 2

Some task where one piece is repeated *n* times, and another piece is repeated *n-1* times and they alternate



(PCM) Nested for loops (1/3)

```
for (int outerLoop = 1; outerLoop <= 5; outerLoop++) {
    System.out.println("outer loop iteration #" + outerLoop);
    for (int innerLoop = 1; innerLoop <= 3; innerLoop++) {
        System.out.println("    inner loop iteration #" + innerLoop);
    }
    System.out.println(outerLoop);
}
```

(PCM) Nested for loops (2/3)

```
for (int outerLoop = 1; outerLoop <= 5; outerLoop++) {
    System.out.println("outer loop iteration #" + outerLoop);
    for (int innerLoop = 1; innerLoop <= 3; innerLoop++) {
        System.out.println("    inner loop iteration #" + innerLoop);
    }
    System.out.println(outerLoop);
}
```

(PCM) Nested for loops (3/3)

```
for (int outerLoop = 1; outerLoop <= 5; outerLoop++) {
    System.out.println("outer loop iteration #" + outerLoop);
    for (int innerLoop = 1; innerLoop <= 3; innerLoop++) {
        System.out.println("    inner loop iteration #" + innerLoop);
    }
    System.out.println(outerLoop);
}
```

Poll in with your answer!

sli.do #cse121

What output is produced by the following code?

```
for (int i = 1; i <= 5; i++) {
  for (int j = 1; j <= i; j++) {
    System.out.print(i);
  }
  System.out.println();
}
```

A.
```
1
12
123
1234
12345
```

B.
```
i
ii
iii
iiii
iiiii
```

C.
```
1
22
333
4444
55555
```

D.
```
1
11
111
1111
11111
```

Poll in with your answer!

What code produces the following output?

sli.do #cse121

A.

```
for (int i = 1; i <= 5; i++) {
  for (int j = 1; j <= i; j++) {
    System.out.print(i);
  }
  System.out.println();
}
```

B.

```
for (int i = 1; i <= 5; i++) {
  for (int j = 1; j <= i; j++) {
    System.out.print(j);
  }
  System.out.println();
}
```

C.

```
for (int i = 1; i <= 5; i++) {
  for (int j = 1; i <= j; j++) {
    System.out.print(j);
  }
  System.out.println();
}
```

D.

```
for (int i = 1; i <= 5; i++) {
  for (int j = 1; j <= i; i++) {
    System.out.print(j);
  }
  System.out.println();
}
```

```
1
12
123
1234
12345
```

Pseudo-Randomness

Computers generate numbers that “look random” in a predictable way using mathematical formulas.

- can use “external” variables like time, mouse position, etc.

True randomness is hard – we rely on natural processes

- e.g., atmospheric noise, lava lamps

Why randomness?

Randomness is a core part of computer science! It powers:

- cryptography
- security
- machine learning!

But true randomness is really hard. If we just use math, someone could "reverse" the formula.

So … lava lamps.

LavaRand: CloudFlare's Wall of Lava Lamps

(PCM) Random

A Random object generates *pseudo*-random numbers.

- The `Random` class is found in the `java.util` package

```
import java.util.*;
```

- We can "seed" the generator to make it behave deterministically (helpful for testing!)

Method	Description
`nextInt()`	Returns a random integer
`nextInt(`*max*`)`	Returns a random integer in the range [0, *max*), or in other words, 0 to *max*-1 inclusive
`nextDouble()`	Returns a random real number in the range [0.0, 1.0)

Poll in with your answer!

sli.do #cse121

Assuming you've declared: `Random randy = new Random();`

Which of these best models picking a random card? (1-13 inclusive)

A. `randy.nextInt()`

B. `randy.nextInt(13)`

C. `randy.nextInt(13) + 1`

D. `randy.nextInt(14)`

(PCM) Math

Calling:

`Math.<method>(…)`

Method	Description
`Math.abs(`*value*`)`	Returns the absolute value of *value*
`Math.ceil(`*value*`)`	Returns *value* rounded up
`Math.floor(`*value*`)`	Returns *value* rounded down
`Math.max(`*value1*`, `*value2*`)`	Returns the larger of the two values
`Math.min(`*value1*`, `*value2*`)`	Returns the smaller of the two values
`Math.round(`*value*`)`	Returns *value* rounded to the nearest whole number
`Math.sqrt(`*value*`)`	Returns the square root of *value*
`Math.pow(`*base*`, `*exp*`)`	Returns *base* raised to the *exp* power

Reminders

- Creative Project 1 is out, due Tue Oct 15th
- Resubmission Cycle 0 released, due Thu Oct 17th
- Getting help today?
  - Matt's office hours: 12:30-1:20 (in-person/Zoom)
  - Brett's office hours: 1:30-2:30 (Zoom)
  - IPL (until 6:30)