CSE 121 Lesson 10: User Input (Scanner) and while loops

Matt Wang & Brett Wortzman

Autumn 2024

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				

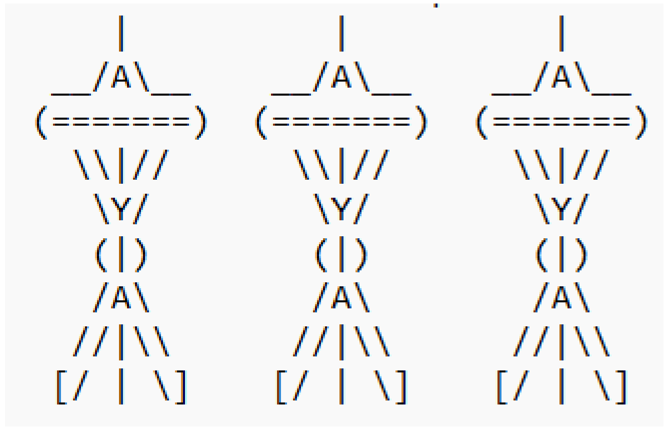
sli.do #cse121

Today's playlist:
121 24au lecture tunes

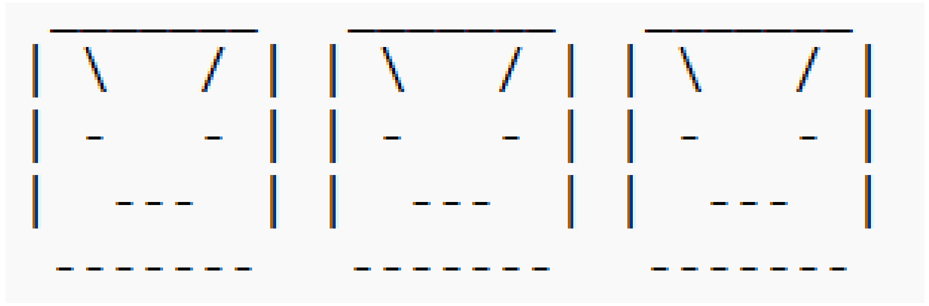
The typical announcements & reminders

- Creative Project 2 (C2) due tonight (October 30th)
- Resubmission Cycle 2 (R2) due tomorrow (October 31st)
  - Eligible: **C0**, P0, C1, P1
  - R3 opens tomorrow, due November 7th; eligible: **P0**, C1, P1, C2
- Programming Assignment 2 (P2) open Friday (November 1st), due November 12th
- Quiz 1 next week (November 7th); topics include up through today
  - more practice materials & resources coming soon!

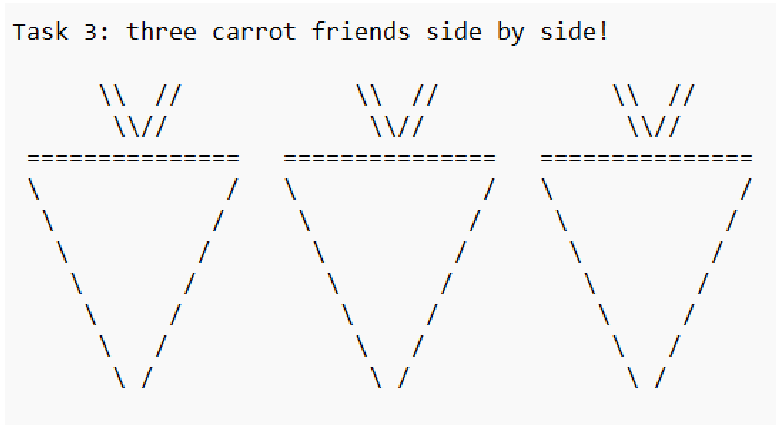
Some C1 Art!



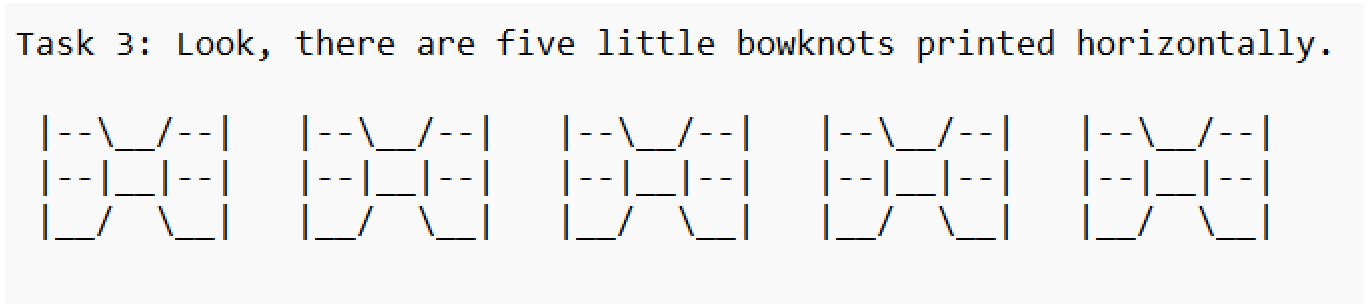
```
    |          |          |
 __/A\__    __/A\__    __/A\__
(=======)  (=======)  (=======)
  \\|//      \\|//      \\|//
   \Y/        \Y/        \Y/
   (|)        (|)        (|)
   /A\        /A\        /A\
  //|\\      //|\\      //|\\
 [/ | \]    [/ | \]    [/ | \]
```



```
 _______    _______    _______
| \   / |  | \   / |  | \   / |
| -   - |  | -   - |  | -   - |
|  ---  |  |  ---  |  |  ---  |
 -------    -------    -------
```

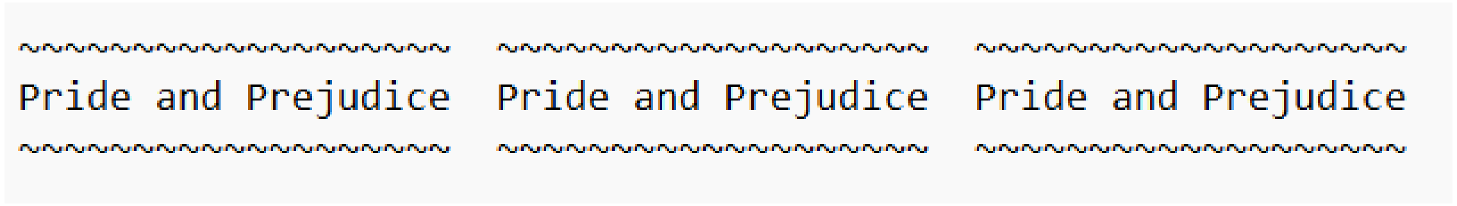


```
Task 3: three carrot friends side by side!

      \\  //            \\  //            \\  //
       \\//              \\//              \\//
 ===============   ===============   ===============
 \             /   \             /   \             /
  \           /     \           /     \           /
   \         /       \         /       \         /
    \       /         \       /         \       /
     \     /           \     /           \     /
      \   /             \   /             \   /
       \ /               \ /               \ /
```

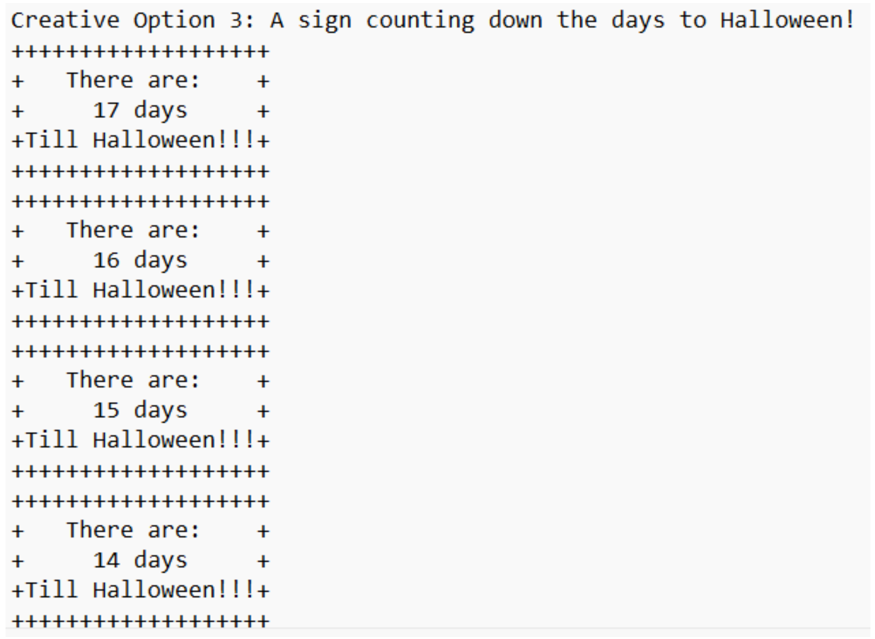


```
Task 3: Look, there are five little bowknots printed horizontally.

 |--\__/--|   |--\__/--|   |--\__/--|   |--\__/--|   |--\__/--|
 |--|__|--|   |--|__|--|   |--|__|--|   |--|__|--|   |--|__|--|
 |__/  \__|   |__/  \__|   |__/  \__|   |__/  \__|   |__/  \__|
```

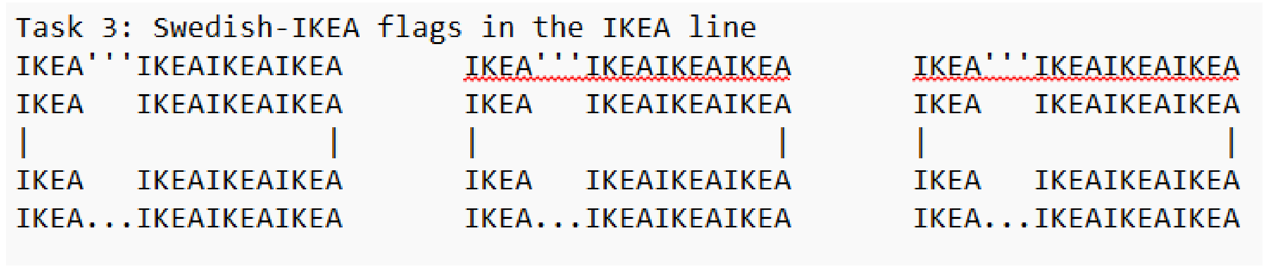
Some C1 Art!



```
~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~
Pride and Prejudice  Pride and Prejudice  Pride and Prejudice
~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~
```

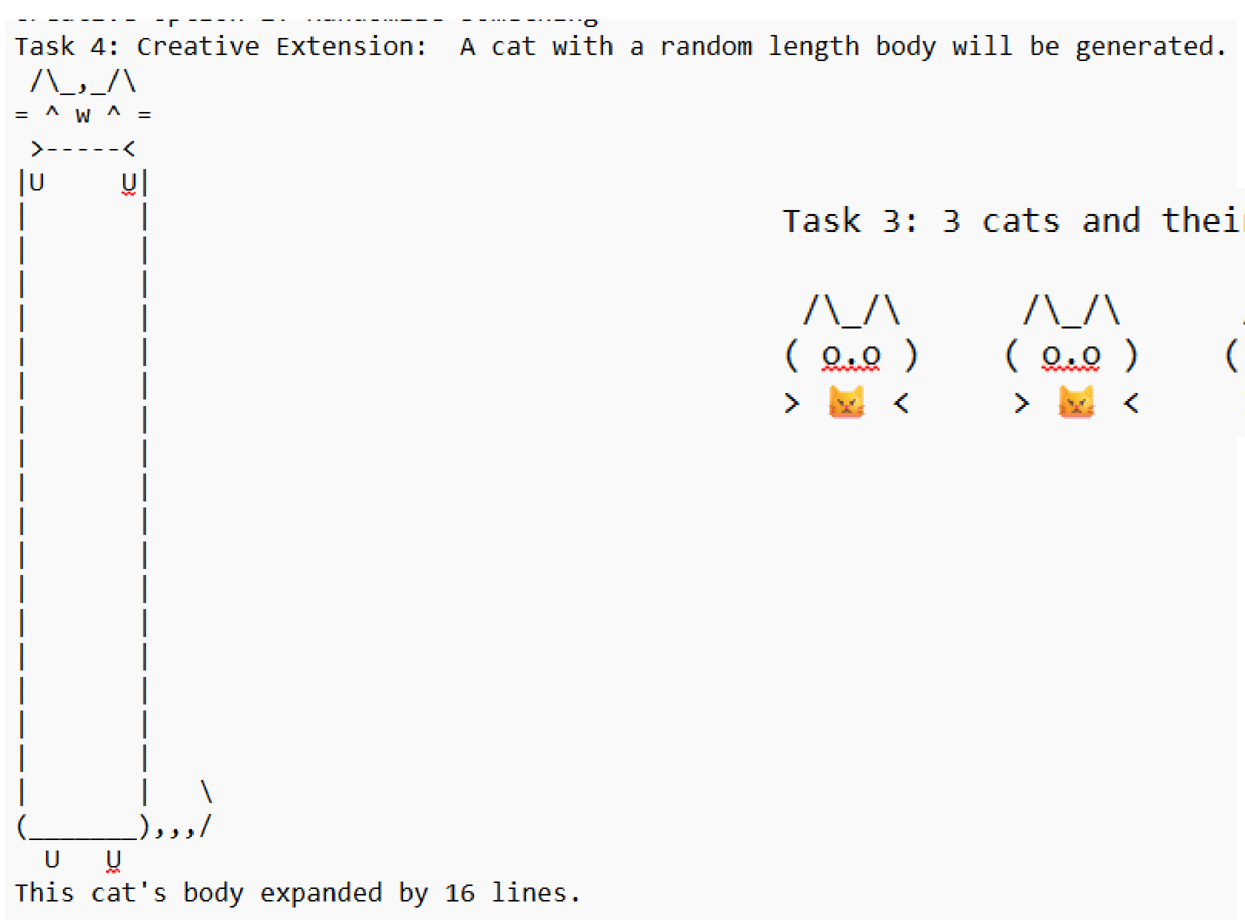


```
Creative Option 3: A sign counting down the days to Halloween!
+++++++++++++++++++
+   There are:    +
+     17 days     +
+Till Halloween!!!+
+++++++++++++++++++
+++++++++++++++++++
+   There are:    +
+     16 days     +
+Till Halloween!!!+
+++++++++++++++++++
+++++++++++++++++++
+   There are:    +
+     15 days     +
+Till Halloween!!!+
+++++++++++++++++++
+++++++++++++++++++
+   There are:    +
+     14 days     +
+Till Halloween!!!+
+++++++++++++++++++
```

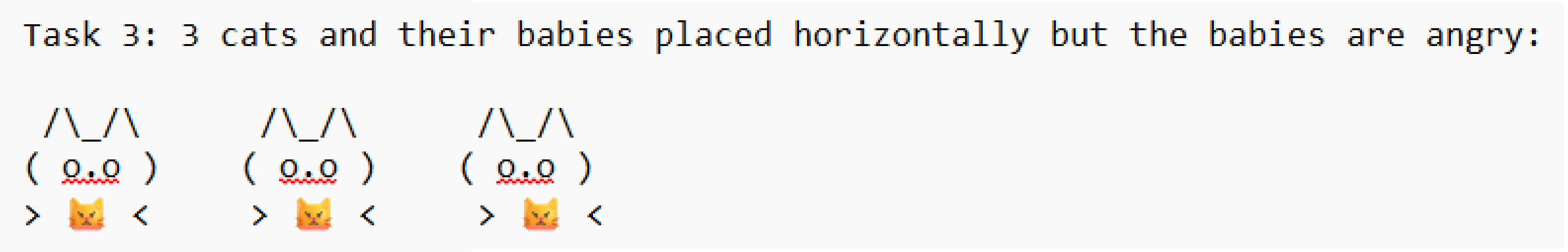


```
Task 3: Swedish-IKEA flags in the IKEA line
IKEA'''IKEAIKEAIKEA       IKEA'''IKEAIKEAIKEA       IKEA'''IKEAIKEAIKEA
IKEA   IKEAIKEAIKEA       IKEA   IKEAIKEAIKEA       IKEA   IKEAIKEAIKEA
|                 |       |                 |       |                 |
IKEA   IKEAIKEAIKEA       IKEA   IKEAIKEAIKEA       IKEA   IKEAIKEAIKEA
IKEA...IKEAIKEAIKEA       IKEA...IKEAIKEAIKEA       IKEA...IKEAIKEAIKEA
```

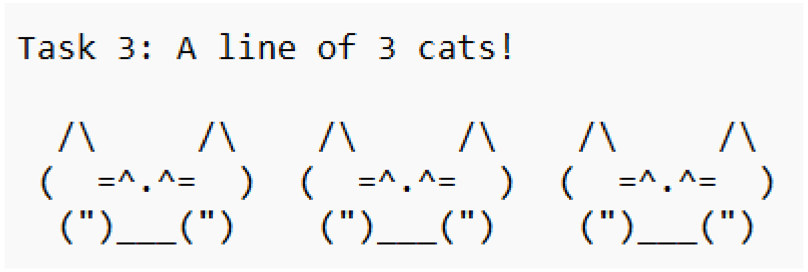
Some C1 Art!



```
Task 4: Creative Extension:  A cat with a random length body will be generated.
 /\_,_/\
= ^ w ^ =
 >-----<
|U     U|
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |
|       |   \
(_______),,,/
  U   U
This cat's body expanded by 16 lines.
```



```
Task 3: 3 cats and their babies placed horizontally but the babies are angry:

 /\_/\      /\_/\      /\_/\
( o.o )    ( o.o )    ( o.o )
> 😾 <     > 😾 <     > 😾 <
```

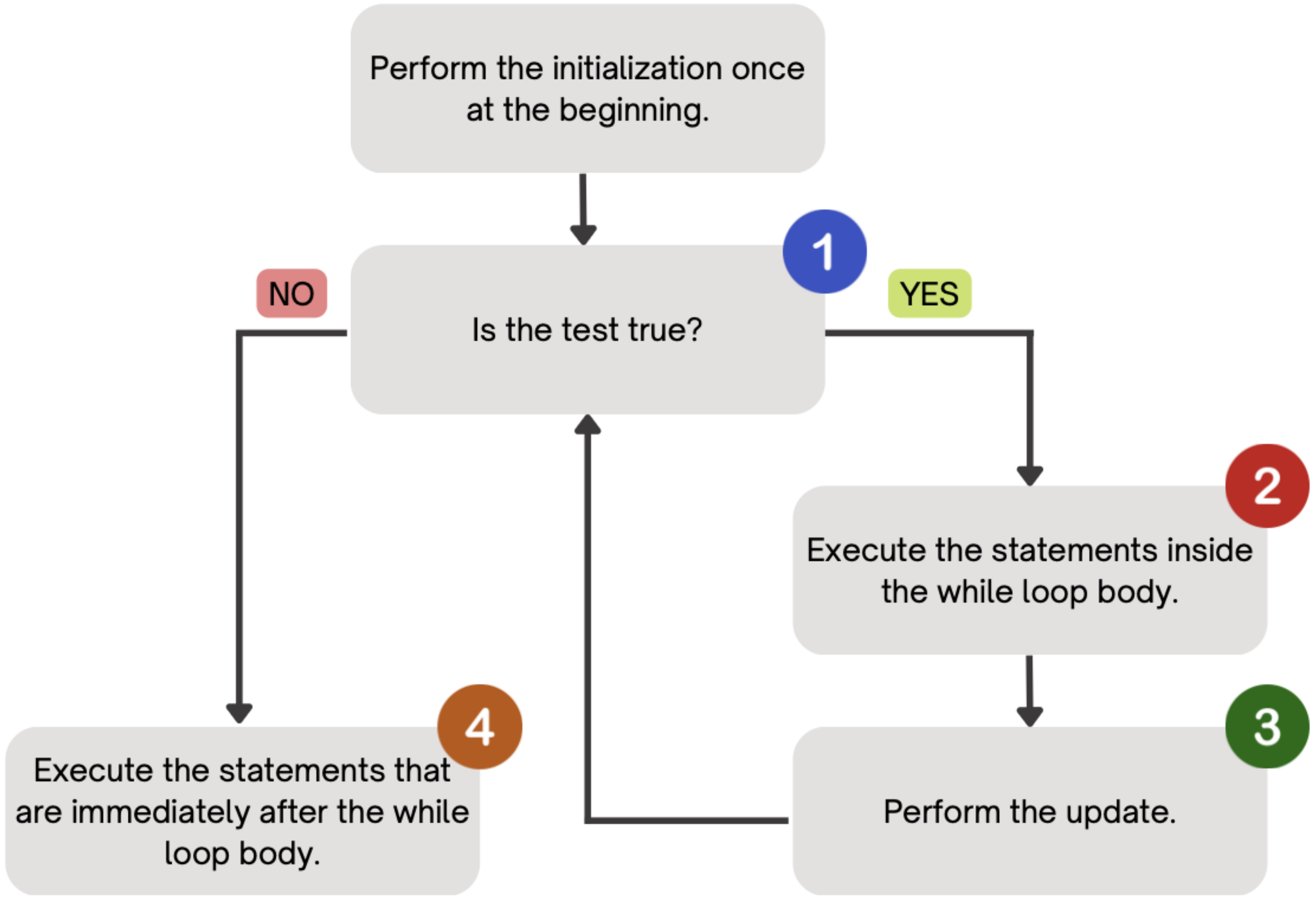


```
Task 3: A line of 3 cats!

  /\     /\    /\     /\    /\     /\
 (  =^.^=  )  (  =^.^=  )  (  =^.^=  )
  (")___(")    (")___(")    (")___(")
```

(PCM) While Loops

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

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


Perform the initialization once at the beginning.
1
Is the test true?
NO
YES
2
Execute the statements inside the while loop body.
3
Perform the update.
4
Execute the statements that are immediately after the while loop body.

`for` loops vs. `while` loops ⚔️

For loops and while loops are quite similar! This is the first (but certainly not the last) time where <u>you</u> 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

How would you describe what the variable x calculates?

```java
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

(PCM) Scanner

Scanner	console	=	new Scanner(System.in);
type	name		Scanner construction code

An **object** that we can use to *read in input*

In the `java.util` "package"!

Methods	Description
`nextInt()`	Reads the next token from the user as an `int` and returns it.
`nextDouble()`	Reads the next token from the user as a `double` and returns it.
`next()`	Reads the next token from the user as a `String` and returns it.
`nextLine()`	Reads an *entire line* from the user as a `String` and returns it.

(PCM) Tokens

A unit of user input, as read by the `Scanner`

- Tokens are separated by *whitespace* (spaces, tabs, new lines)

```
23   John Smith
       42.0     "Hello world"  $2.50 "  19
```

Poll in with your answer!

When calling the following method, which of these user inputs would not cause an error? (choose multiple)

sli.do #cse121

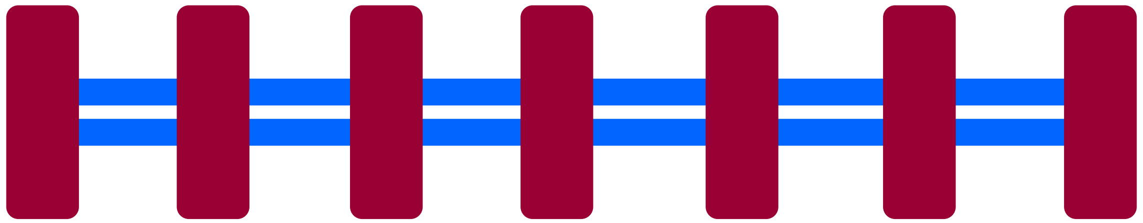
```
public static void cornbear() {
  Scanner console = new Scanner(System.in);
  int amt = console.nextInt();
  String firstName = console.next();
  String secondName = console.next();
  double price = console.nextDouble();
}
```

A. 6 Lucy's Treats $12.48
B. 3 Oatmilk Latte 16.47
C. 2 The Hunger Games 21.98
D. 4 Gigis 900.24
E. 2 Grammy Awards 90095

Fencepost Pattern

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

`L-a-u-f-e-y-!`



Quick Meals for Thought (Names)

What assumptions are we making here?

```
String firstName = console.next();
String lastName = console.next();
```

1. All first and last names have no spaces
2. All people only have one first or last name
3. All people have at least one first or last name

Interesting readings: Falsehoods Programmers Believe About Names, For Afghans, Name and Birthdate Census Questions Are Not So Simple

Quick Meals for Thought (Inputs)

Another assumption: all computer users have a keyboard & mouse!

- many blind & low-vision users only use keyboards (no mice)
- some users cannot use keyboards and use alternatives
  - e.g. "switch access" – famously used by Stephen Hawking

This isn't "just" about disability:

- your user might be on a phone, tablet, gaming console, or "smart" TV!
- your user could be using text-to-speech!
- your user's keyboard or mouse might be broken!