

Problem Solving and Algorithms

Fran Trees (fran.trees@cs.Rutgers.edu)



Problems with Problem Solving in CS

- We need to consider student background and experiences.
- We tend to focus on coding.
 - Lots of task overloading
 - Language syntax
 - IDE

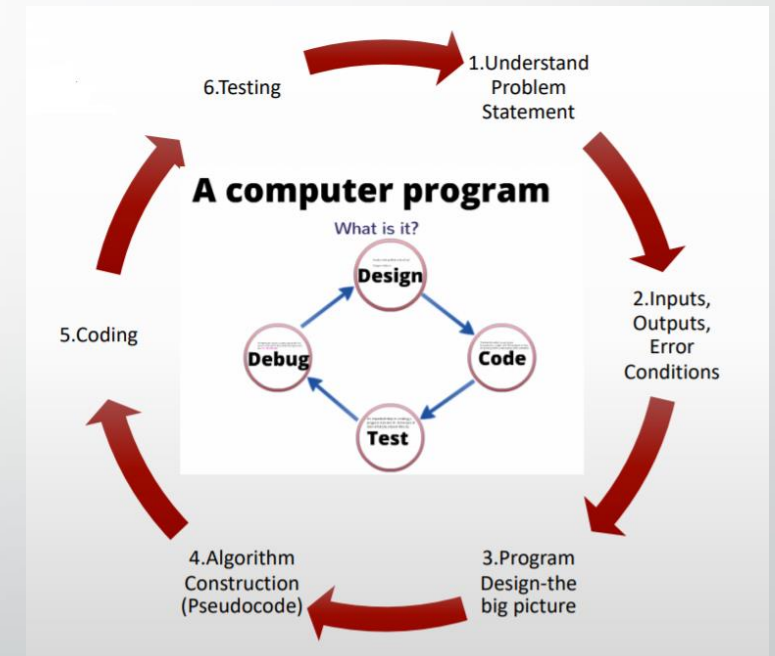
Computational Thinking

- **Problem solving methods (what we want to encourage):**
 - Understanding the problem
 - Express the solution in some language (English)
 - Translate the solution to an algorithm (Abstraction)
 - Convert the algorithm to a program that the computer can solve (Automation)
 - Test the program (Analyze)
- **What the students end up doing!**
 -
 -
 -
 - Write a program
 -

Program Development Cycle

- **Problem Development (3 Cs)—**
OUR responsibility!:
 - Communication
 - Clarity
 - Completeness

If this were brought to a court of law, what would the outcome be?



First (somewhat boring) Problem

- **Problem.** Ellen and Ana are moving in together. Determine how many cats they will have in total?

Easy?

What is CS, REALLY?

```
public class Cats {  
    public static void main(String[] args){  
        int anaCats = Integer.parseInt(args[0]);  
        int ellenCats = Integer.parseInt(args[1]);  
        if(anaCats < 0 || ellenCats < 0){  
            System.out.println("error");  
        }  
        else {  
            int totalCats = anaCats + ellenCats;  
            System.out.println("Total cats = " + totalCats);  
        }  
    }  
}
```

Java Solution

NOT Easy!



- Motion
- Looks
- Sound
- Events
- Control
- Sensing
- Operators
- Variables
- My Blocks



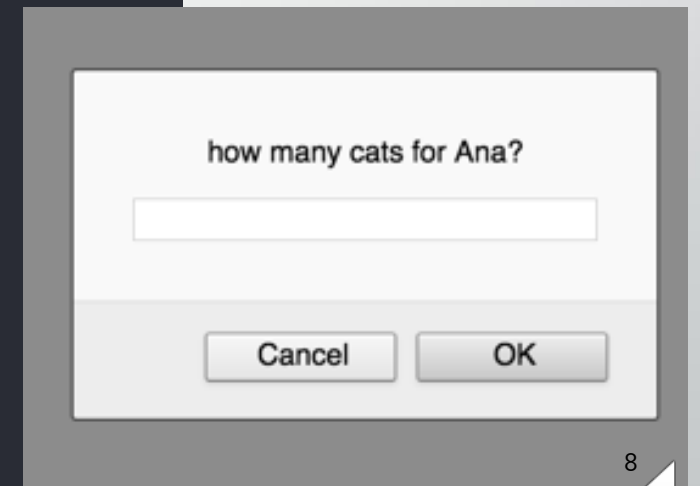
Scratch Solution

NOT Easy!

JavaScript Code

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "UTF-8" />
    <title> CATS </title>
  </head>

  <body>
    <script>
      var anaCats = Number(prompt("how many cats for Ana?"));
      var ellenCats = Number(prompt("how many cats for Ellen?"));
      var totalCats;
      if(anaCats < 0 || ellenCats < 0){
        alert("Error");
      }
      else{
        totalCats = anaCats + ellenCats;
        alert("Total cats = " + totalCats);
      }
    </script>
  </body>
</html>
```



JavaScript Code

```
<body>
  <h1> CATS </h1>
  <p> How many cats for Ellen? </p>
  <input type = "text" id = "ellen" value = "" size = "20"
  onchange = "
    ellenCats = Number(ellen.value);
  "/><br><br>
  <p> How many cats for Ana? <br>
  <input type = "text" id = "ana" value = "" size = "20"
  onchange = "
    anaCats = Number(ana.value);
  "/><br><br>
  <button id="total"
  onclick = "
    if(anaCats < 0 || ellenCats < 0){
      totalBox.value = 'ERROR';
    }
    else{
      totalCats = ellenCats + anaCats;
      totalBox.value = totalCats;
    }
  " >Calculate Total Cats</button> <br><br>
  Total Cats<br>
  <input type = "text" id = "totalBox" value = "" readonly /></p>
</body>
```

NOT EASY!!!

CATS

How many cats for Ellen?

2

How many cats for Ana?

3

Calculate Total Cats

Total Cats

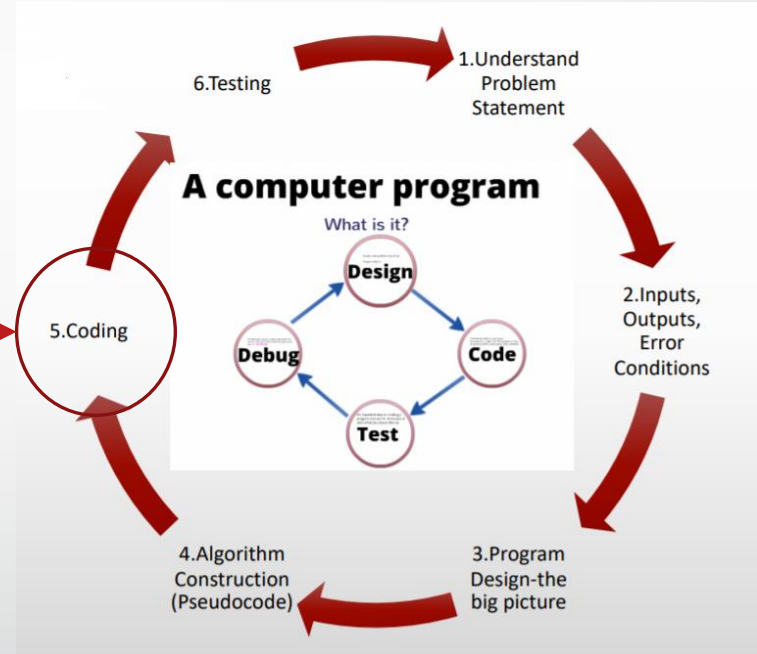
5



First (somewhat boring) Problem

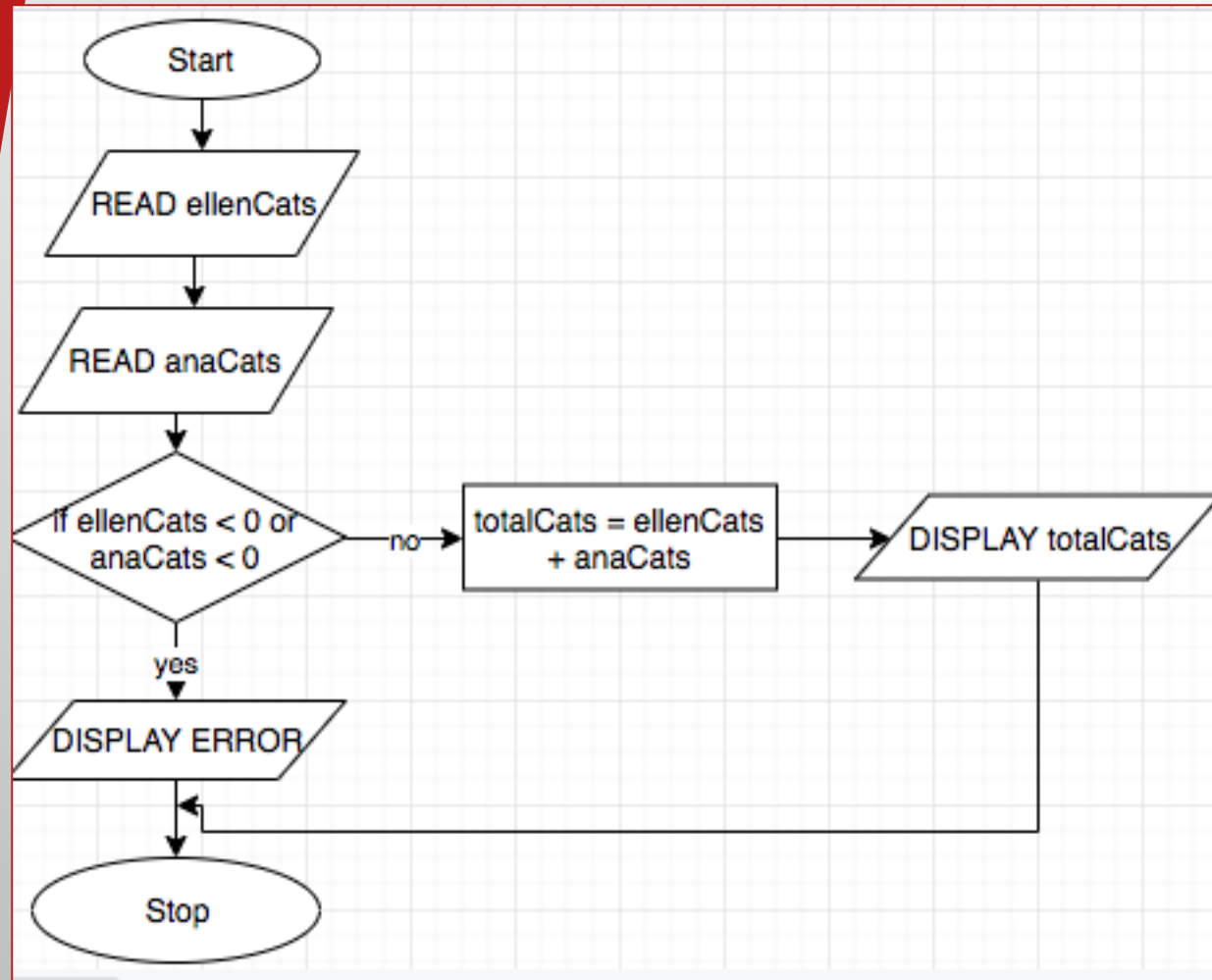
- **Problem.** Ellen and Ana are moving in together. Determine how many cats they will have in total?
- **Solution Design.** Identify inputs, outputs and error conditions.
- **Answer:**
 - **Inputs.** ellenCats, anaCats
 - **Output.** totalCats
 - **Error conditions.** Bad input (e.g. Ana's cats is -5)

Not only



What is CS?

A Common Language



READ ellenCats

READ anaCats

IF ellenCats < 0 **OR** anaCats < 0 **THEN**

DISPLAY ERROR

ELSE

COMPUTE totalCats **AS** ellenCats + anaCats

DISPLAY totalCats

ENDIF

Let's talk CS!

A Common Language

READ ellenCats

READ anaCats

IF ellenCats < 0 **OR** anaCats < 0 **THEN**

DISPLAY ERROR

ELSE

COMPUTE totalCats **AS** ellenCats +
anaCats

DISPLAY totalCats

ENDIF

- **Inputs.** ellenCats, anaCats **Output.** totalCats
Error Conditions
 - [1, 3][4]
 - [0, 2] [2]
 - [4, 0] [4]
 - [-2, 3] [ERROR]

A Common Language

It's about counting!

READ ellenCats

1 oper.

READ anaCats

1 oper.

IF ellenCats < 0 **OR** anaCats < 0 **THEN**

2 oper.

DISPLAY ERROR

1 oper.

ELSE

COMPUTE totalCats AS ellenCats + anaCats

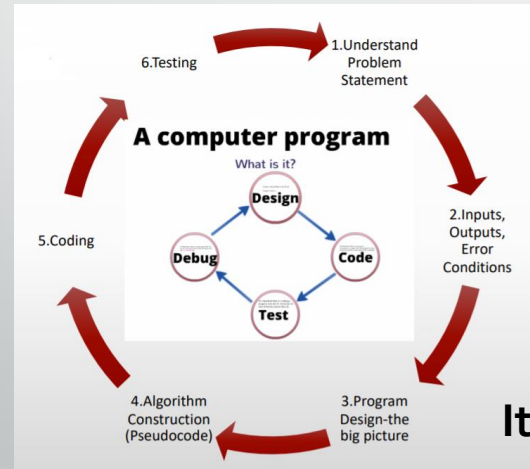
1 oper.

DISPLAY totalCats

1 oper.

ENDIF

- **Inputs.** ellenCats, anaCats
- **Output.** totalCats
- **Error Conditions**
 - [1, 3][4]
 - [0, 2] [2]
 - [4, 0] [4]
 - [-2, 3] [ERROR]
 - [2, -1] [ERROR]
- **Number of operations:**
 - **MIN: 5**
 - **MAX: 6**



It's all about counting!

Different Algorithms to solve the same problem

Find the oldest person in a room

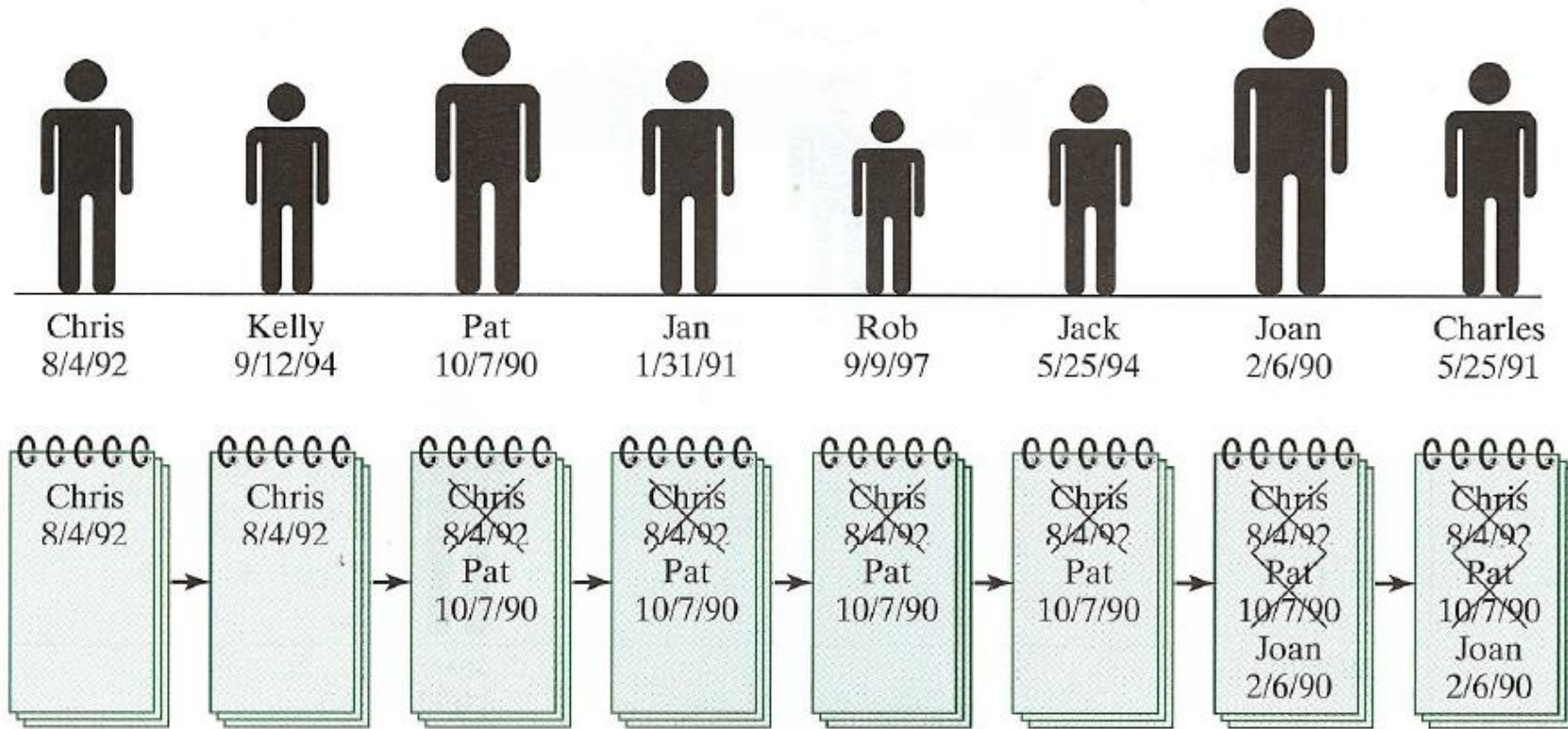
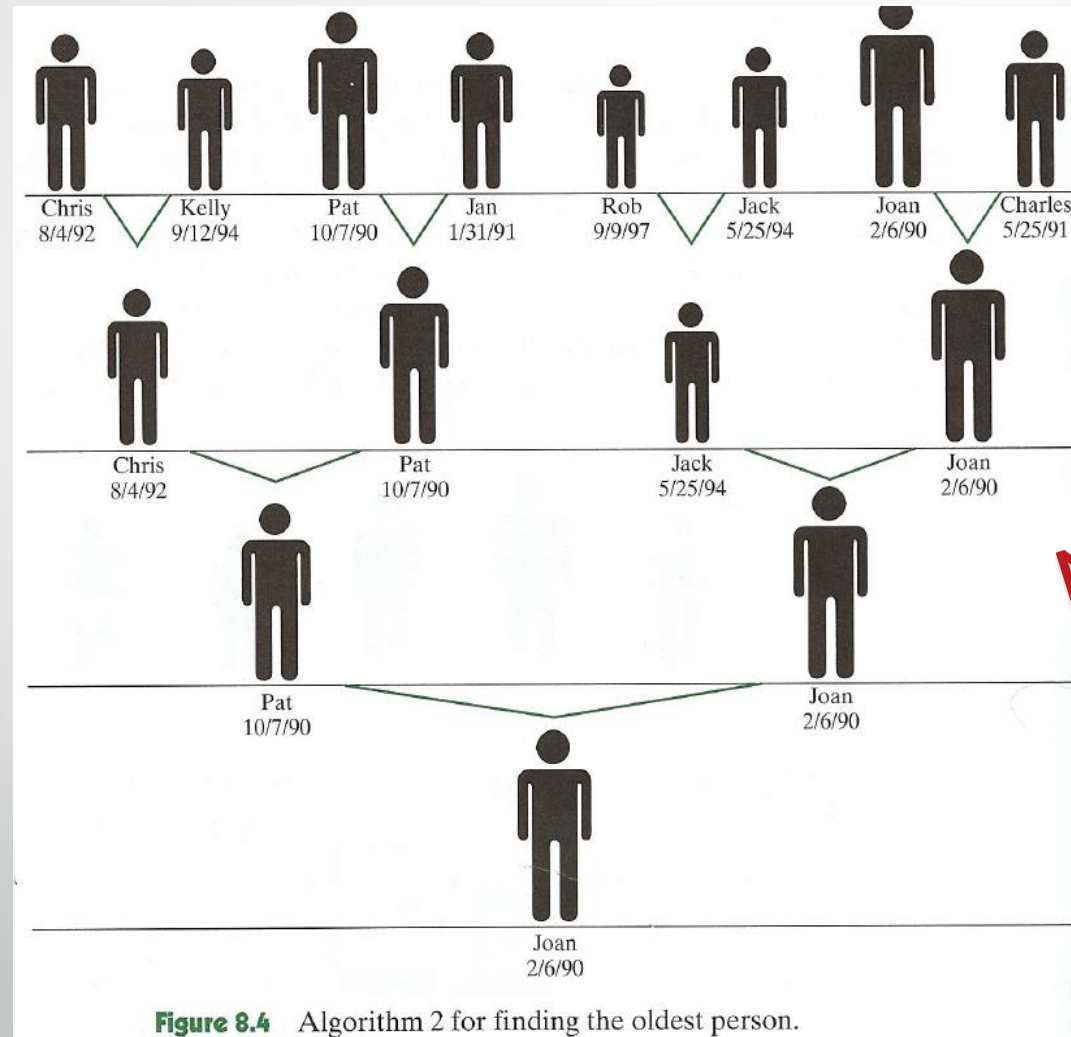


Figure 8.3 Algorithm 1 for finding the oldest person.

Number of operations?

Different Algorithms to solve the same problem

Find the oldest person in a room



Number of operations?

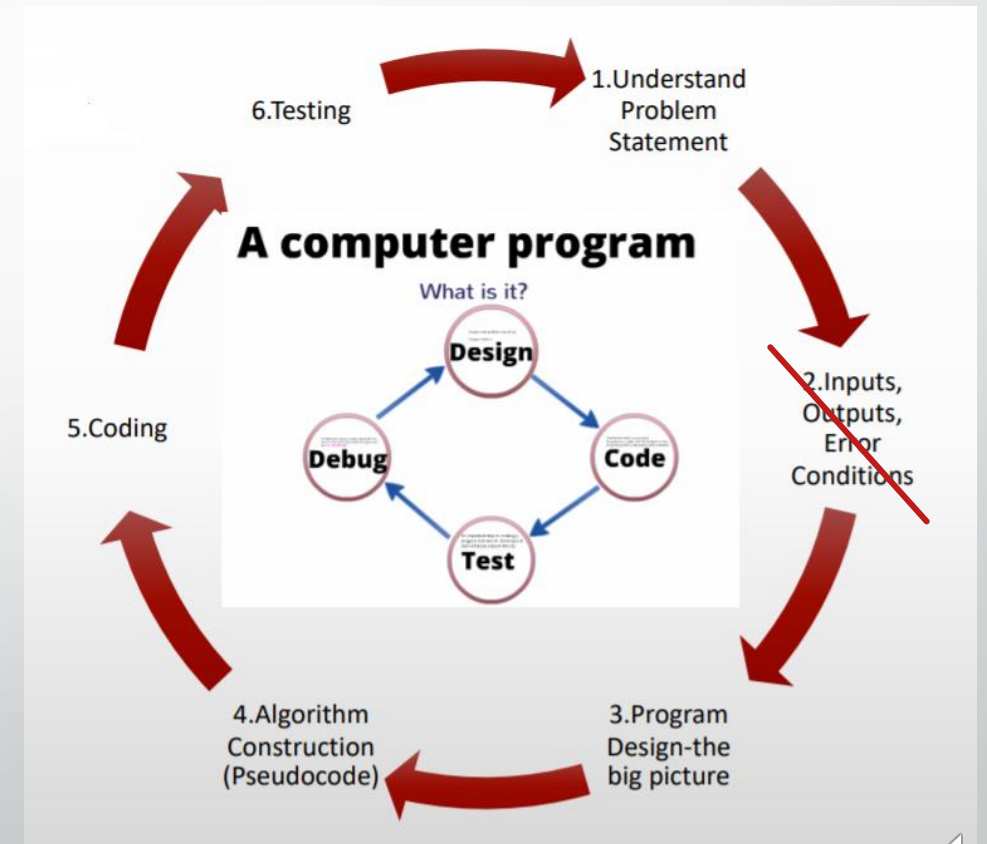
Not only about counting!
Not only about coding!

Is there more than ONE way to solve a problem?

I want to fill a 20-element container with random integers from 1 to 100?

Problem Development (3 Cs):

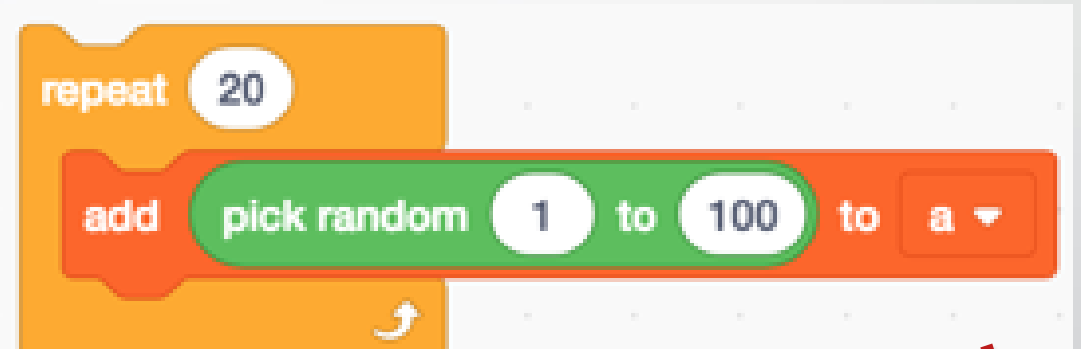
- Communication
- Clarity
- Completeness



I want to fill a 20-element container with random integers from 1 to 100?

```
int[] a = new int[20];
for(int k = 0; k < MAXELTS; k++){
    a[k] = (int) (Math.random()*100 + 1);
}
```

```
<body>
<script>
    var a = [];
    var text = "";
    for(i = 0; i < 20; i++){
        var num = Math.random() * 100 + 1;
        a[i] = Math.floor(num);
        text += (a[i] + " ");
    }
</script>
<button id="go"
onclick = "
    arrayBox.value = text;
    " >See values</button> <br><br>
<input type = "text" id = "arrayBox" value = "" size = "50"/>
</body>
```



Let's NOT code!

The challenge :

(great for pair/team collaboration)

PROBLEM: Fill a 20-element container with random integers from 1 to 100 inclusive **ensuring that there are no duplicate values in the container.**

Think of **TWO** algorithms to solve this problem!

While doing so, remember to talk CS! (space, time to execute, etc.)

Solution 1 :

```
for (int k = 0; k < 20; k++)
```

```
do
```

```
    Generate random integer(1-100)
```

```
    Check all elements in array that are filled already until random  
    number is not a dupe
```

```
while random num is a dupe
```

```
Put the non-dupe in array
```

8

4
8
6
23
*

Solution 1 :

```
for (int k = 0; k < 20; k++)
```

```
do
```

```
    Generate random integer(1-100)
```

```
    Check all elements in array that are filled already until  
    random number is not a dupe
```

```
while random num is a dupe
```

```
Put the non-dupe in array
```

8

4
8
6
23
*

What about

- TIME
- SPACE

Solution 2:

- Create an additional array, *b*, with 101 elements (random choices are from 1-100) and fill it with 0s.

```
for (int k = 0; k < 20; k++)
```

```
do
```

```
    Choose a random number, r
```

```
    while b[r] != 0
```

```
    Put r in your array
```

```
    Put 1 in b[r] to mark it "taken"
```

Solution 2 : a picture

```
for (int k=0; k < 20; k++)
```

```
do
```

```
    Choose a random number between 1 and 100, r 8
```

```
    while b[r] != 0
```

```
    Put r in your array
```

```
    Put 1 in b[r] to mark it "taken"
```

What about

- TIME
- SPACE

b		a
b[0]	0	4
b[1]	0	8
b[2]	0	
b[3]	0	
b[4]	1	
b[5]	0	
b[6]	0	
b[7]	0	
b[8]	1	
b[9]	0	
b[10]	0	
b[11]	0	

Solution 3:

- Create an additional array, b , with 101 elements (random choices are from 1-100) and fill it with indices of b ($b[1] = 1, b[5] = 5$). Set $n = 100$;

for (int k=0;k<20;k++)

 Choose a random number, r , in the range $[1, n]$

 Put $b[r]$ in your array

$b[n]$ replaces $b[r]$

 Decrement n

Solution 3 : a picture

`n = 100`

```
for (int k=0; k < 20; k++)
```

Choose a random number,

r , in the range $[1, n]$ **4**

Put $b[r]$ in your array

$b[n]$ replaces $b[r]$

Decrement n **$n = 99$**

What about

- TIME
- SPACE

b[0]	0
b[1]	1
b[2]	2
b[3]	3
b[4]	4 100
b[5]	5
b[6]	6
b[7]	7
b[8]	8
b[9]	9
b[10]	10
b[11]	11

A
4

Solution 4:

- Create an array, `b`, with 100 elements (random choices are from 1-100) and fill it with (index + 1) (`b[0] = 1, b[4] = 5`).

`for (int k=0;k< 20;k++)`

Choose a random number, `r`, in the range `[0, 99]`

Swap `b[k]` with `b[r]`

Solution 4 : a picture

```
for (int k=0; k < 20; k++)
```

Choose a random number,

r , in the range $[0, 99]$ 4

Swap $b[r]$ with $b[k]$

copy the first 20
elements of b to an array
of size 20 and
assign b to this reference.

to swap:

```
int temp = a[i];
```

```
a[i] = a[j];
```

```
a[j] = temp;
```

What about

- TIME
- SPACE

b[0]	1 5
b[1]	2
b[2]	3
b[3]	4
b[4]	5 1
b[5]	6
b[6]	7
b[7]	8
b[8]	9
b[9]	10
b[10]	11
b[11]	12

Solution 5 :

- Assign to each of the first 100 elements of an `ArrayList`, `b`, the value of `index + 1`.
- Create a new empty `ArrayList` (or 20-element array), `a`.

```
for(int x = 1; x <= 20; x++)
```

Choose a random integer in the range `[0, b.size() ]`

Remove this element from `b` and add it to `a`.

Solution 5 : A picture

- Assign to each of the first 100 elements of an ArrayList, b, the value of index + 1.
- Create a new empty ArrayList, a.

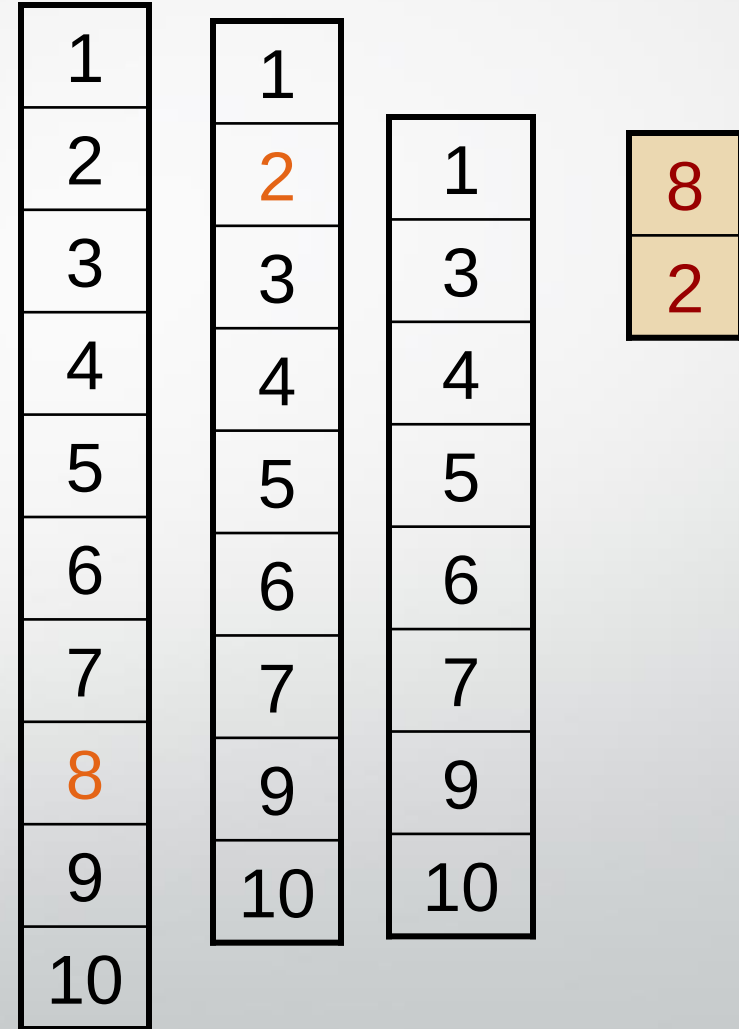
```
for(int x = 1; x <= 20; x++)
```

Choose a random integer in the range [0, b.size()]

Remove this element from b and add it to a.

What about

- TIME
- SPACE



You should be able to



- Give advantages and disadvantages of each solution.
- **Discuss time and space considerations.**
- **Think about number of operations.**
- Code one of them!
- Think about readability.

Let's teach more than coding!

Problem Solving and Algorithms

Fran Trees

fran.trees@cs.rutgers.edu



Resources

- <https://introcs.cs.princeton.edu/java/stdlib/> (Standard input/output libraries)

QUESTIONS????

THANKS MUCH!!!!