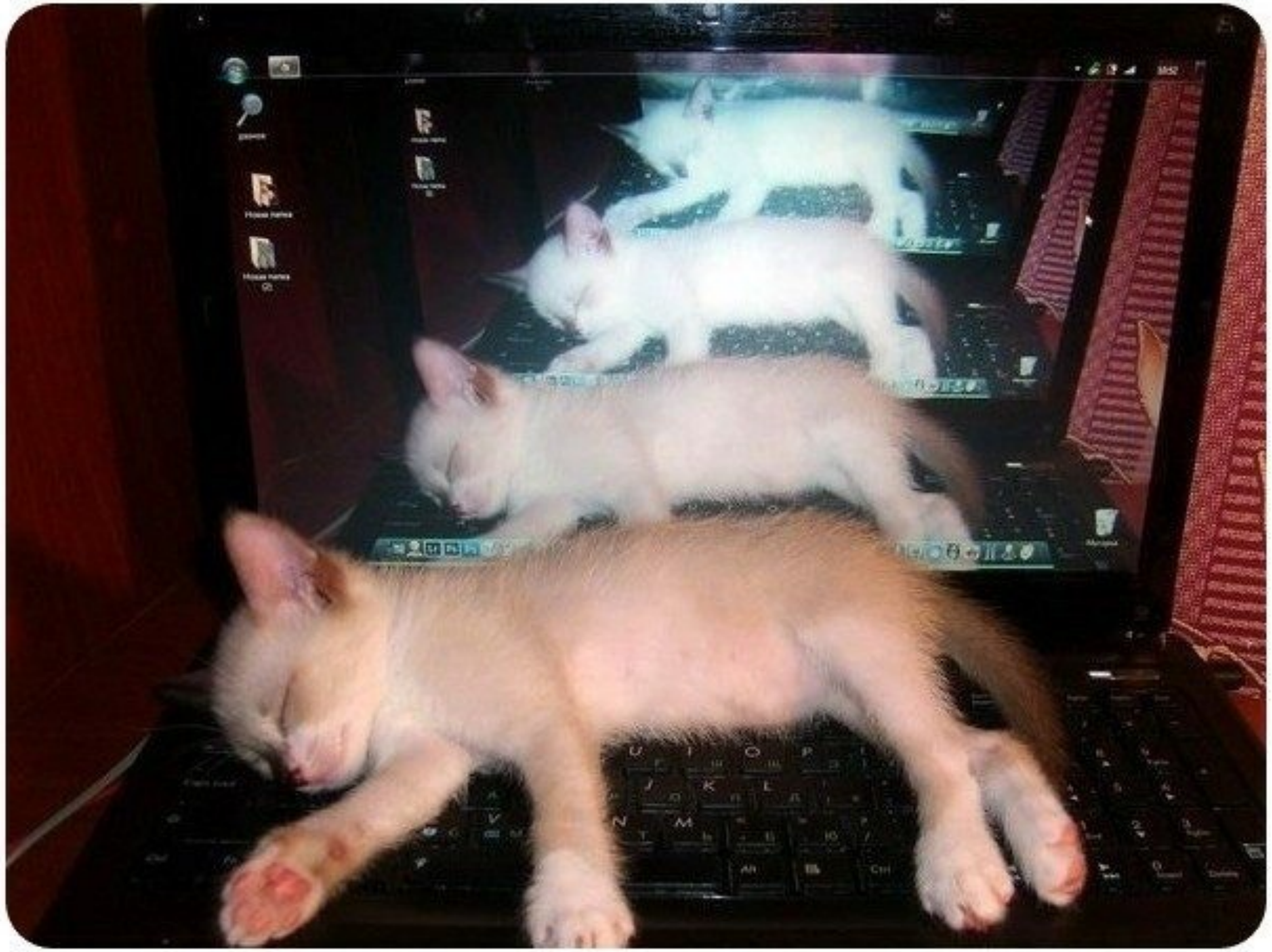
Lecture 8: Recursion

07/11/22

Today's sponsor is Recursive Cat!
Recursive cat:

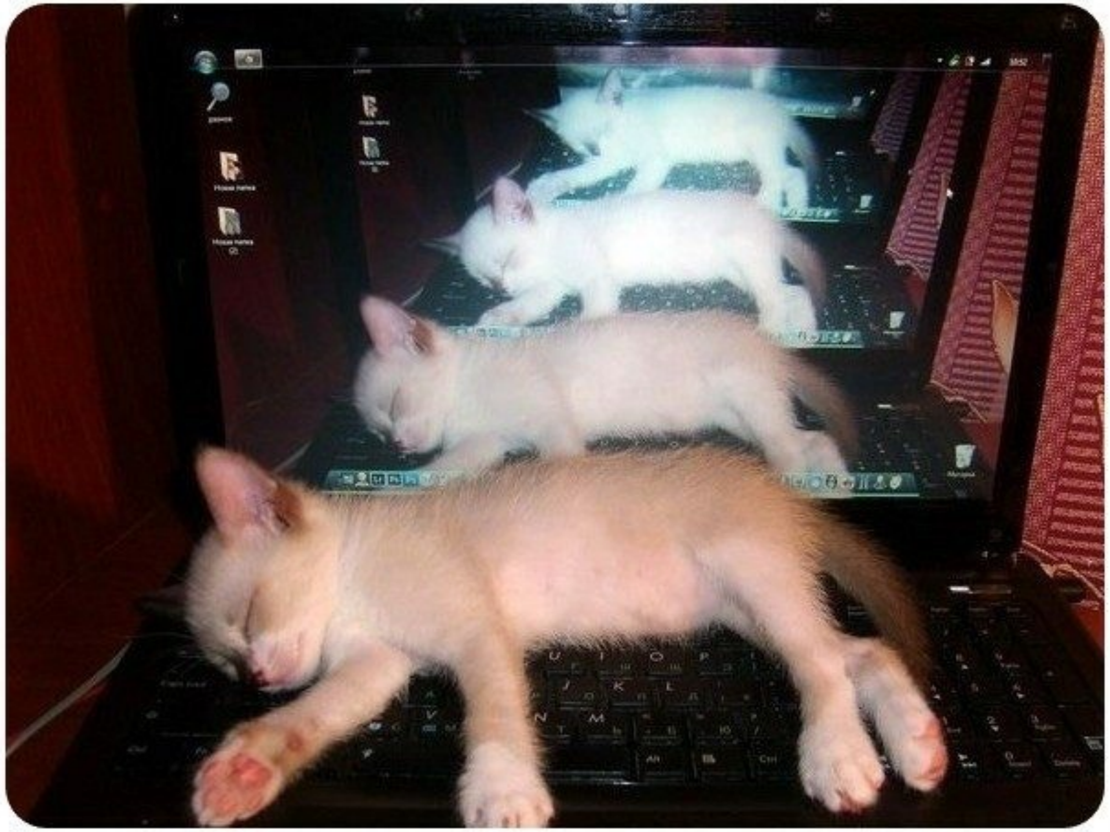


Announcements

- Husky Card required for access to GUG starting today
  - Doors at the front of the room are open if you need another way in

Recursion!

Today's sponsor is Recursive Cat!
Recursive cat:



Roadmap for the week

- Monday
  - Introduce idea of **recursion**
  - Goal: Understand idea of recursion and **read** recursive code
- Tuesday
  - Practice **reading** recursive code
- Wednesday
  - More complex recursive examples
  - Goal: Identify recursive structure in problem and **write** recursive code
- Thursday
  - Practice **writing** recursive code

Recursion

- **recursion**: A problem defined in terms of itself.
  - Solving a problem using recursion depends on solving smaller occurrences of the same problem.

- **recursive programming**: Writing methods that call themselves to solve problems recursively.
  - An equally powerful substitute for *iteration* (loops)
  - Particularly well-suited to solving certain types of problems

What row are you sitting in?

Getting down stairs

- Need to know two things:
    - How to get down one step
    - How to recognize the bottom

- Most code will look like this:

```
if (simplest case) {
    compute and return solution
} else {
    divide into similar subproblem(s)
    solve each subproblem recursively
    assemble the overall solution
}
```

Recursion and cases

Every recursive algorithm involves at least 2 cases:

- **base case:** the simplest case
- **recursive case:** does a tiny bit of work, then breaks down the problem into a smaller version of itself

Some recursive algorithms have more than one base or recursive case, but all have at least one of each.

Poll Everywhere

pollev.com/cse143

What is the output of this code?

```java
public static void main(String[] args) {
    method1(5);
}

public static void method1(int n) {
    method2(n - 1);
    System.out.println(n + " bananas");
}

public static void method2(int n) {
    System.out.println(n + " apples");
    method3(n - 1);
    System.out.println(n + " strawberries");
}

public static void method3(int n) {
    System.out.println(n + " oranges");
}
```