

# **CSE 143**

## **Lecture 14 (A)**

More Recursive Programming

reading: 12.2 - 12.3, 12.5

slides created by Marty Stepp

<http://www.cs.washington.edu/143/>

# Exercise

- Write a method `crawl` accepts a `File` parameter and prints information about that file.
  - If the `File` object represents a normal file, just print its name.
  - If the `File` object represents a directory, print its name and information about every file/directory inside it, indented.

```
cse143
  handouts
    syllabus.doc
    lecture_schedule.xls
  homework
    1-sortedintlist
      ArrayIntList.java
      SortedIntList.java
      index.html
      style.css
```

- **recursive data:** A directory can contain other directories.

# File objects

- A `File` object (from the `java.io` package) represents a file or directory on the disk.

| Constructor/method                 | Description  |
|------------------------------------|--|
| <code>File(<b>String</b>)</code>   | creates <code>File</code> object representing file with given name |
| <code>canRead()</code>             | returns whether file is able to be read                            |
| <code>delete()</code>              | removes file from disk   |
| <code>exists()</code>              | whether this file exists on disk                                   |
| <code>getName()</code>             | returns file's name  |
| <code>isDirectory()</code>         | returns whether this object represents a directory                 |
| <code>length()</code>              | returns number of bytes in file                                    |
| <code>listFiles()</code>           | returns a <code>File[]</code> representing files in this directory |
| <code>renameTo(<b>File</b>)</code> | changes name of file   |

# Public/private pairs

- We cannot vary the indentation without an extra parameter:

```
public static void crawl(File f, String indent) {
```

- Often the parameters we need for our recursion do not match those the client will want to pass.

In these cases, we instead write a pair of methods:

- 1) a public, non-recursive one with the parameters the client wants
- 2) a private, recursive one with the parameters we really need

# Exercise solution 2

```
// Prints information about this file,  
// and (if it is a directory) any files inside it.  
public static void crawl(File f) {  
    crawl(f, "");    // call private recursive helper  
}  
  
// Recursive helper to implement crawl/indent behavior.  
private static void crawl(File f, String indent) {  
    System.out.println(indent + f.getName());  
    if (f.isDirectory()) {  
        // recursive case; print contained files/dirs  
        for (File subFile : f.listFiles()) {  
            crawl(subFile, indent + "    ");  
        }  
    }  
}
```