

## CS100J 22 February 2005

**More on Methods.** Developing methods

**Read section 2.5 on stepwise refinement**

**Listen to PLive activities 2.5.1 -- 2.5.4!!**

### Quotes that relate to specifying a method before writing it.

A verbal contract isn't worth the paper it's written on.

What is not on paper has not been said.

If you don't know where you are going, any road will take you there.

If you fail to plan you are planning to fail.

1

## Prelim on Thursday evening, 7:30 to 9:00

On Thursday, Gries is out of town (at a conference of SIGCSE --Special Interest Group on Computer Science Education.)

Two TAs will take his place. Thursday's lecture will be devoted to a review for the prelim. You can ask questions; if you don't, the TAs will lecture on what they think is important. They know what is on the test.

The test will be graded Sunday afternoon and the grades posted directly after that.

This week's lab: devoted to the development of functions that deal with Strings. It is useful practice for the prelim.

2

## Anglicizing an integer

*/\*\* = the English equivalent of n, for 1 <= n < 1,000*

e.g. `ang(3)` is "three"

`ang(412)` is "four hundred twelve"

`ang(762)` = "seven hundred sixty two" *\*/*

```
public static String ang(int n)
```

Hint at how to do it. When we add 5 + 3 + 2 + 8, we start with  $x = 0$  and add one value to  $x$  at a time, step by step:

$x = x + 5;$   $x = x + 3;$   $x = x + 2;$   $x = x + 8;$

**Definition of  $x$ :**  $x$  is the sum of the values added so far.

Can we start with String variable  $s = ""$  and step by step catenate pieces on to the end of it?

What's the definition of  $s$ ?

3

## Anglicizing an integer

*/\*\* = the English equivalent of n, for 1 <= n < 1,000*

e.g. `ang(3)` is "three"

`ang(641)` is "six hundred forty one" *\*/*

```
public static String ang(int n)
```

Start with String variable  $s = ""$  and step by step catenate pieces on to the end of it? Use two local variables,  $s$  and  $k$ .

	$s$	$k$
start:	<code>""</code>	641
	<code>"six hundred"</code>	41
	<code>"six hundred forty"</code>	1
end	<code>"six hundred forty one"</code>	0

**Definition of  $s$  and  $k$ :**

`ang(n)` is  $s + \text{ang}(k)$

**To find `ang(n)`, anglicize  $k$  and append the result to  $s$ .**

4

## Anglicizing an integer

*/\*\* = the English equivalent of n, for 1 <= n < 1,000*

e.g. `ang(3)` is "three"

`ang(641)` is "six hundred forty one" *\*/*

```
public static String ang(int n) {
```

```
    // ang(n) is s + ang(k)
```

```
    String s = "";
```

```
    int k = n;
```

```
}
```

This definition of  $s$  and  $k$  is *very important*. This definition will drive the development. Whenever we append something to  $s$ , we have to change  $k$  to keep the definition true. The definition helps us develop the method body and helps the reader understand it.

Whenever you declare a local variable whose value will change often over execution of the method, **write a comment near its declaration to define it!!**

**You** will use the definition often as you develop the method. Without the definition, you will forget what the variable means and will make mistakes.

5

## Anglicizing an integer

*/\*\* = the English equivalent of n, for 1 <= n < 1,000*

e.g. `ang(3)` is "three"

`ang(641)` is "six hundred forty one" *\*/*

```
public static String ang(int n) {
```

```
    // ang(n) is s + ang(k)
```

```
    String s = "";
```

```
    int k = n;
```

```
}
```

The rest of this lecture is devoted to the development of this algorithm, using DrJava. The final program will be put on the course website before 7AM Wednesday morning.

6