Trouble-shooting Java code

Syntax issues
   Fix the first error message first
   (others may appear or disappear)
   The error may be before Java finds it
   Indent rigorously (make the errors stand out)
Get a second set of eyes
Get some coaching (office hours)

File Scanning Loops

A common kind of “for each x” loop:
   for each line in file ...
   for each word on line ...

Finding Palindromes

Suppose we want to find the longest palindrome in a long list of words
   * Such as one of the word lists used by spelling checkers
We already have a palindrome checking method
   static boolean isPalindrome( String s )

Design the logic for finding the longest palindrome in the list (in pseudocode)
   (do this on the blackboard)

Now let’s turn it into Java ...

Get a Scanner for a named file
Loop through each word in the file
Opening a File

Scanner in = new Scanner(  
    new File( fileName )  
);

What if there is no such file?

Finding the methods we need ...

The methods we need ...

boolean hasNext()  
Returns true if this scanner has another token in its input.

String next()  
Finds and returns the next complete token from this scanner.
So we can translate into Java ...

pseudocode “for each word in file ...”
becomes

```
while ( in.hasNext() ) {
    String candidate = in.next();
    ...
}
```

Peek ahead: Arguments to “main”

```
public static void main(String[] args) throws FileNotFoundException {
    final String defaultFile = "words.txt";
    String fileName;
    if (args.length < 1) {
        fileName = defaultFile;
    } else {
        fileName = args[0];
    }
```

Command line arguments

```
public static void main(String[] args) { ... }
```

Then to run the program:

```
$ java Palindrome words.txt
```

“words.txt” becomes the value of args[0]

Scan for longest

```
static String findLongestPalindrome(Scanner in) {
    String longest = "";
    int maxLen = 0;
    while ( in.hasNext() ) {
        String candidate = in.next();
        if (candidate.length() > maxLen && isPalindrome( candidate )) {
            maxLen = candidate.length();
            longest = candidate;
        }
    }
    return longest;
```
Two patterns for extrema (1)

If there is a good default value
   Something that will definitely be replaced by
   the first iteration of the loop ...

int biggest = impossiblySmallValue;
while ( ... ) {
   if (thisOne > biggest) { ... } ...
}

If there is no good default ...

if ( at least one input ) {
   biggest = next input;
   while ( more inputs ) {
      current = next input;
      if (current > biggest) {
         biggest = current;
      }
   }
} // else handle case of no inputs

Scanning Line by Line, Word by Word

while (infile.hasNextLine()) {
   String line = infile.nextLine();
   Scanner fields = new Scanner ( line );
   String muppetName = fields.next();
   ...

   class C {
      public C() { ... } // constructor
      void foo() { ... };
   }

   C myObject = new C();
   myObject.foo();
Formatted output

System.out is a PrintStream object
PrintStream has methods for formatted output

see documentation for java.util.Formatter

System.out.format("%-10s %3d%n", name, changes);

Method Design: Making Change

There is a lot of repetition in code to make change for each coin.
Let’s make it better.

Summary

File I/O: by line, by word
Sometimes nested loops with Scanner
Formatted output
Patterns to precisely control
printing of each field