Review Session: Exceptions, GUI's.

Exceptions: "About the Final" handout, what you have in your "About the Final" handout, what you need to know.

- Array/String Index Out Of Bounds - trying to access an index that doesn't exist
- Arithmetic Exception - trying to divide by zero
- NullPointerException - trying to access fields or methods of a null object
- Number Format Exception - trying to parse a string that doesn't represent a number

Exceptions can (should) be caught by a method that knows how to deal w/ the problem.

Example:

```
Exception:

Some code on part of input

Causes code on part of input

```

Who should catch this?

This method, because it "knows" the input was bad, can ask user for new input.

(process input).

- How can processInput throw an Exception?
  - try to divide by zero, access null field
  - system creates one (e.g., by divide by zero)
  - throw new Exception(); (some pre-existing or a new Exception class)
  - throw new Exception("processInput threw a problem...");

- How can processInput catch an Exception?
  - inside processInput
    - try
      - processInput();
    - catch (Exception re) {

What if no catch?

What if throw Exception? How do we catch class Exception? Exception?
(0) Consider the following methods:
runEverything(...): creates a GUI that responds to user requests. Calls processInput.
processInput(...): gets and processes entire input from the user. Calls calcAllInfo
to preprocessAllInfo(...): collates info about result of preprocessing each piece of user input. Calls
processFirstBit(...).
processFirstBit(...): preprocesses the first bit of user input.

We consider what (should) happen if processFirstBit throws an Exception.

(1) (From Spring 2010's final) We have forgotten how to find the length of a string s, and we are in a
hurry. We do remember that s.charAt(k) throws a StringIndexOutOfBoundsException if k
is not the index of a character s. So we (meaning you) write the function below, using a loop (with init-
ialization) that successively evaluates s.charAt(0), s.charAt(1), s.charAt(2), ... until the
exception is thrown, at which time k will be the length! Write the body of the function. You will need a
try-statement.

```java
/** = length of string s */
public static int length(String s) {
    int k = 0;
    // inv: s[0..k-1] exists.
    while (true) {
        try {
            s.charAt(k); // totally useless; just check if access allowed
        } catch (StringIndexOutOfBoundsException e) {
            // bad case, if true.
            return k;
        }
        k = k + 1;
    }
}
```

(2) How were the 3 steps for getting something to listen to an event actually implemented in the acm
package we used for A7? (Remember we said that these things were done "under the hood").

1. Real 3 steps:
   - add what should happen if event occur
     - Some class (e.g. GUI) has to contain a method for handling the event
   - all Java objects of this class specify what should happen
     - add "implement_eventListener" to class header
   - for a component when an event could occur,
     - register a listener as a listener

   ```java
cgent.addMouseMotionListener(d);
```
a GCanvas, part of a Graphics Pane (which Bracket Excludes)

- JFrame
- BorderLayout
  - add (button, BorderLayout.EAST)
- JPanel
  - add (button)
- Box : BoxLayout
  - constructor : Box (BoxLayout.X_AXIS)
  
  - JButton
  - JLabel
  - JTextField
  - JTextArea