Errors
Types of errors

• There are three general types of errors:

1. **Syntax** (or “compile time”) errors
   – Syntax errors are “grammatical” errors and are detected when you compile the program
   – Syntax errors prevent your program from executing

2. **Runtime errors**
   – Runtime errors occur when you tell the computer to do something illegal
   – Runtime errors may halt execution of your program

3. **Logic errors**
   – Logic errors are not detected by the computer
   – Logic errors cause your results to be wrong
Syntax errors

• A syntax error is a “grammatical” error – bad punctuation, misuse of keywords, wrong return type, etc.

• Syntax error messages tell you two things:
  1. The line number at which an error was detected
     – Usually, this is the line containing the error
     – In some cases, the actual error is earlier in the program text
     – Use an editor that shows line numbers!
  2. What the compiler thinks the problem is
     – Since the compiler cannot know what you meant, the message is only a “best guess,” and is sometimes misleading

• Syntax errors can cascade: An early error can cause spurious error messages later in the program
  – Always fix the earliest message first
  – If later messages don’t make sense, try recompiling
Runtime errors

• A **runtime error** occurs when your program does something illegal
  – Runtime errors typically stop your program
  – Runtime errors can be “caught” by your program, thus allowing the program to continue running
  – E.g., catch indexing out of array bounds

• Runtime errors are usually caused by something the program did (or failed to do) earlier in execution
  – Because the *cause* of the error is somewhere else in the program, runtime errors are usually harder to solve
Null pointer exception

• The `NullPointerException` is one of the most common runtime errors
  – It occurs when you send a message to a null variable (a non-primitive variable that doesn’t refer to an actual object)
  – The null variable causing the problem is *always* just before a dot
• Eclipse example: `ArrayOfNull.java`
Logic Errors

- A logic error is when your program compiles and runs just fine, but does the wrong thing.
- In very simple programs, logic errors are usually easy to find and fix, if they occur at all.
- In all but the simplest of programs,
  - 10% of your time is spent writing the program and fixing the syntax errors (more if you are still learning the syntax)
  - 90% of your time is spent finding and fixing runtime and logic errors
- Logic errors can take hours, or even days, to find
  - Allocate your time accordingly!
How to minimize errors

• While you cannot *avoid* making errors, you can *prepare* for them
  – Keep your programs as *simple* as possible
  – *Indent* properly so you can see the structure of your code
  – *Comment* your programs so you can find things again
  – Write *test cases* and use them
  – “Write a little, test a little”
  – Write assert statements for the things that you “know” cannot possibly happen

• Programming is a creative activity, and can be very enjoyable and satisfying
  – For most of us, debugging is *not* the fun part (well, maybe sometimes...)
Approaches to finding errors

• Put in **print** statements
• Use the **Debugger**
• Explain your code to a friend (even if they can’t program)
  – Forces you to look at and think about what you wrote
• Write tests before you write code!
  – Use JUnit to build and run your test suites