Program DEVELOPMENT & DEBUGGING

Kaminski/cs3310

These are skills you are expected to be further developing by doing the course assignments. As a reminder, here are some basic software engineering practices which introduce fewer bugs into your program in the first place, and make them easier to detect and fix when they are in your program. For example,

- Incrementally develop and implement the program/project in a modular fashion so you can more easily pinpoint which code caused the problem to occur. The program should have been working correctly (and you should be 100% sure of this) BEFORE a NEW module is added. Incremental development also means you pretty much always have your project in a "working state" where it at least "does something" that's demonstratable. [That'll guarantee you have "something" to hand in when the due date comes up and you've already used up your grace days].

- Use good, standard, self-documenting naming of variables, methods, classes, objects, programs, ... Use the same naming as described in the requirements specs. Use commonly-used naming (the "everybody knows" rule). Uphold the "truth in advertising" rule - e.g., a method should do what it's name says it does.

- Modularize the project using a page/screen or less size methods (and classes, maybe?). Use OOP to modularlize if your language permits. Static classes are useful for organizing a program to group methods together, even if there are no objects involved.

- Do a thorough mental and paper/pencil “walk-through” of your algorithms and pseudo-code at the design stage, and then again on each new module that's introduced.

- In debugging, it may help to leave your computer for awhile and just sit down with a printout of your program to do a thorough “desk-check” of a “walk-through” of the code. And don’t make leaps and assumptions - examine the entire flow! [The code you leaped over is usually the spot where the bug is].

- Narrow down where the problem is by determining where in the code is the last place where you’re SURE that things are working correctly? And be 100% sure by examining the contents of variables using an interactive debugger or inserting WriteLine's in the code. Then gradually move the WriteLine down a line - or step through the code line-by-line with the interactive debugger.

- Consider whether it’s the PROGRAM itself that's wrong or the DATA which caused the problem. This is particularly true when dealing with I/O and files - e.g., did you run your update program earlier, which changed the contents of the file itself, so no wonder . . .?

- Use the commonly acceptable format/style/practices (see my FormatSpecs document) to reduce the potential for bugs: e.g., good naming, aligning/indenting, “if/else/…” for mutually exclusive conditions, “while” for event-controlled looping vs. “for” for count-controlled looping, parameter-passing rather than global variables, keep re-using i for forLoops rather than "going through the alphabet", break long code into callable methods, use a hierarchical control structure (vs. a goto-style looping, e.g., with recursion).

- Check for common types of errors: e.g., “edge cases” (first/last record/node in a file/array/data structure), starting at 1 or 0 as appropriate (when the opposite should be used), using the wrong read/loop structure (read/process or process/read), mix up of “and” and “or” in while condition.