Debugging Programs

Finding and Eliminating Errors

What is a 'bug'?

- a software bug is an error, flaw, failure, or fault in a computer program or system
- a bug causes the program to produce an incorrect or unexpected result, or to behave in unintended ways
- 'bug reports' detail the bugs in a program
- 'debugging' is the process of identifying and eliminating bugs

Much of your time as a computer programmer will likely be spent debugging. This reality was quickly discovered by the early computer programmers:

"As soon as we started programming, we found to our surprise that it wasn’t as easy to get programs right as we had thought. We had to discover debugging. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs."

(Maurice Wilkes, 1949)

Why Learn to To Debug

- few programs are written correctly the first time
- some errors can be solved by inspection (i.e., just looking at the code), but many cannot
- without a debugging strategy, some programmers will make random changes
  - if lucky, this might fix a simple problem in a simple program
  - usually, this will make the problem worse, or add more problems

Print Debugging or Tracing

- use output statements to report the current state of the program at key locations in the code
  - values of important variables
  - reaching important locations, particularly within if/else branches or exiting loops
- make sure output statements are not part of your final product!
  - you may comment them out, rather than delete them completely

Rubber Duck Debugging

- carry around a rubber duck (or teddy bear, or some other inanimate object)
    - also works with another programmer
- when code is not working, explain the code, line-by-line, to the rubber duck
- this forces the programmer to slow down and take the time to think about what is happening in their code
- programmers will often identify and solve their own problems this way

Divide and Conquer

- remove (or comment out) sections of code where the problem might exist
  - usually recently added code
- one piece or section at a time, add in code, testing each time
- usually helpful if considerable new code has been added without any validation along the way

Manual Walkthrough or Tracing

- usually done with paper and pencil
- record and update variable values while working through the lines of code
- may also track conditional statements
  - if/else statements
  - loop enter & exit conditions
- can be done without running the program on a computer (e.g., written tests)
- does not depend on specific code language

Debugging Software Tools

- many <u>integrated development environments</u> (IDEs) include the ability to control and inspect the program as it is running
- trace code execution one line at a time
- display variable values
- set breakpoints at key locations
  - program runs normally until hitting breakpoint
  - inspect variables at breakpoint
  - continue normally or step through from there