Debugging and Troubleshooting

INFO/CSE 100, Spring 2005
Fluency in Information Technology

http://www.cs.washington.edu/100
Readings and References

• Reading

  » Fluency with Information Technology
  • Chapter 7, To Err is Human

  “To err is human, but it takes a computer to really foul things up”
Using Computers...

- In IT, stuff goes wrong ... debugging is the process of finding the error
  » Term coined by Grace Murray Hopper
- Best solution ... make no mistakes!
  » Be accurate ... get it right the 1st time
  » Follow a process that makes it easier to get it right
  » Computers can’t make "common sense" decisions about what we really meant. They do what we say, not what we mean.
When You Debug...

Debugging is not algorithmic: no guaranteed process

• There are guidelines for debugging…

  Rather than trying things aimlessly and becoming frustrated, think of yourself as solving a mystery

  • Be objective: What are my clues? What is my hypothesis? Do I need more data?
  • Consciously ‘watch’ yourself debug -- its an out-of-body experience
  • When stumped, don’t become frustrated, but ask, “What am I misunderstanding?”

  Become Sherlock Holmes
Debugging Guidelines

» Verify that the error is reproducible
» Determine exactly what the actual failure is
» Eliminate the “obvious” causes by checking
» Divide process into working/faulty parts
» On reaching a dead end, reassess the information you have, trying to identify the mistake you are making
» Work through process making predictions and checking they’re fulfilled
Reproducibility

• First step: verify the error is reproducible
  » You can't find something that you can't reproduce
  » Get out and get back in. Does it still happen?
    • Restart the application.
    • Try a different application
    • Reboot the operating system. Sometimes this is appropriate, especially for errors involving peripheral devices (printers, modems)
Determine the Problem

- Second step: figure out what’s wrong
  » Often there is a sequence of steps following an error and propagating it … work backwards looking to see where the error first occurred
Eliminate the Obvious

• Third step: eliminate obvious causes
  “If the cause were obvious, the problem would have been fixed!” - Yeah, right.

» There are standard things to check:
  • Inputs
  • Connections
  • “Permissions”
  • Physical connectivity
  • Requirements

My fan is broken, it won't turn on!
Isolate the Problem

• Try to “partition” the situation into working and non-working parts
  • Form a hypothesis of what’s wrong
  • Make as few assumptions as possible
  • Take nothing for granted

The goal is to eliminate as many things from consideration as possible
At a Dead End, Reassess

• When everything seems to check out, don’t get frustrated

• Instead, ask yourself “What am I overlooking or misunderstanding?”
  » Your goal is to see the situation as it is, not as you think it should be
    • Am I assuming too much?
    • Am I misreading the clues?
    • What can I eliminate or simplify?

• Explain the situation to a friend
Make Predication/Check

• Beginning with the isolated part, step through the process, predicting the outcome and verifying it
  » A prediction that is not fulfilled shows…
    • A possible bug
    • A possible misunderstanding
    • A chance to narrow the search

‘Sleeping on it’ often helps!
Summary

- Debugging is not algorithmic, but there are guidelines to follow
  - Stay calm - high blood pressure clouds your brain
  - Be organized as you investigate and fix things
  - Recognize that you may feel a little embarrassed when you finally figure out the problem.
    - If we were perfect, we would never make mistakes ...
    - A little humility is a good thing for all of us
  - Watch yourself debug -- assess how you are doing, what you need to know