Announcements

Project 1A is due today at 5:00
Midterm 1 is next Friday
In this room ... arrive on time
Cover material in 1st 9 Lectures + Labs
Bring only Photo ID and a pencil/pen

Tip of the day: The most useful habit for successful computing is that of being perfectly accurate

Debugging & Troubleshooting

“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
• In most cases computers can’t recover for our errors
The standard of precision for computers is perfect, which is tough for people, but try!

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?”

Debugging Guidelines

1. Verify that the error is reproducible
2. Determine exactly what the problem is
3. Eliminate the “obvious” causes
4. Divide process into working/faulty parts
5. On reaching a dead end, reassess the information you have, trying to identify the mistake you are making
6. Work through process making predictions and checking they’re fulfilled

Reproducibility

First step: verify the error is reproducible
• Transient errors are very rare, but they do happen ... try again

Getting Out and Getting Back In

* Rebooting the operating system is advisable, 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 so obvious, the problem would have been fixed!”

- There are standard things to check:
  - Inputs
  - Connections
  - “Permissions”
  - Physical connectivity

Isolate the Problem

Fourth step: Try to divide 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

Fifth step: When everything seems to check out, don’t get frustrated... ask, “What am I misunderstanding?”

- Your goal is to see the situation as it is, not as you think it should be
  - Are you assuming too much?
  - Are you mis-reading the clues?

Sometimes, stepping back to the surrounding context is helpful

Make Prediction/Check

Sixth: 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” may help!

A Debugging Example

After building a class web page, we find it is wrong

Houston, we have a problem
Debugging Demo

FIT100: Bringing Light to Computer Users
Winter 2006

Summary

Debugging is not algorithmic, but there are guidelines to follow

* It probably pays to memorize them so they come to mind while debugging
* Watch yourself debug -- assess how you are doing, what you need to know
* Being accurate -- avoiding textual mistakes at all costs -- saves frustration

Notice how few letters mess up a whole page