Debugging

(This bug is on display at the Smithsonian!)

Core Debugging Principles

- Computers are deterministic machines: every problem has a cause
- Methodically eliminating possibilities will eventually determine cause
- Checking the program state narrows down where to search.
  - Add print statements of variable values
  - Study error messages closely

Famous Error Message: Apollo 11 1202 & 1201 alarm

Error Messages

- Very helpful for diagnosing problems
- Are always *downstream* of the real issue
- Often, they *do not* identify the root problem

Printing variables and reading error messages let you bracket where the error could be

What to look for

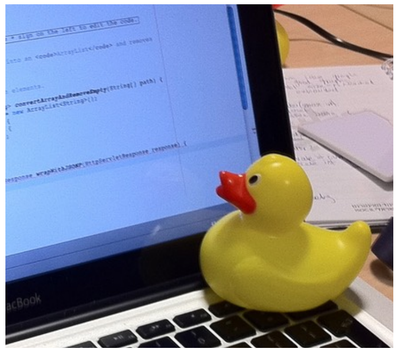
- **Syntax errors:**
  bracket or parenthesis mismatch,
  incorrect form if/elif/else,
  missing commas,
  misspelt keyword,
  incorrect indentation level
- **Wrong operator used** (example: ^ instead of **)
- **Misspelt variable or function name**
- **Incorrect function usage**
- **Logic error in program design**

Getting help

- Explain the issue to a rubber duck!
  "Rubber duck debugging" helps you understand the issue better and solve it!
  en.wikipedia.org/wiki/Rubber_duck_debugging

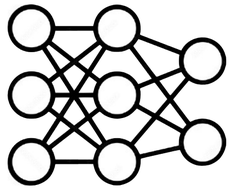
- Ask your friends—a fresh set of eyes can be helpful!

- **Consult the internet via:**



Web search (Google)

- Answers & advice in context



AI (LLMs)

- Solutions identified via internet-scale approximate recall

What to do when you get stuck

- Take a breath / don’t panic!
- Slow down: understanding the code more important than a quick fix
- Online search for issue / ask for help
- Look over the documentation

Exercises: tinyurl.com/compassdebug