Capstone Course

Clean Error Handling

Admin

- How is choosing a project going?
-

Take-away on error handling

- Clean error handling synopsis
  - Error handling is important, but if it obscures program logic it is no good.
  - If the error handling logic is growing everywhere over and in your code like kudzu – you have trouble

Kudzu Error handling in C

```
int init_abc()
    {
        if (!init_a())
            goto err_a;
        if (!init_b())
            goto err_b;
        if (!init_c())
            goto err_c;
        return 1;
     err_c:
        cleanup_b();
     err_b:
        cleanup_a();
     err_a:
        return 0;
    }
```

credit https://news.ycombinator.com/item?id=3883310

Even more simply

- def foo():
- var = get_data()
- if var is not None:
- do_something(var)
- else:
- handle_var_error()
-
- Still control flow is dominated by error checking
- Note: Golang thinks they have an approach that works

So perror?

- Same advice from earlier in the semester
  - Prefer exceptions to error codes
  - If you are a systems programming in C – so be it
  - Otherwise use your Stroustrup given exceptions
    - Or Learning Research Group – given if you want to go all smalltalk purist.

Using exceptions

- Again, not too many scope levels in one method
  - And do one thing per method
  - Exception generating code is code that does one thing
  - Exception handling code is code that does another thing
  - Note that code on page 106 of clean code needs refactoring.

Exception handling

- Exception handling is like a database transaction
  - Try attempts to make a change of state
    - Entire change must be successful to complete.
  - If change is successful, then new state is consistent
  - If change was not successful, then exception handler (catch/except etc) must make the state consistant.

Java exceptions

- In Java what are checked vs unchecked exceptions?
- Which kind does python have? (or both?)

Java exceptions

- In Java what are checked vs unchecked exceptions?
  - Checked exceptions: those that the compiler knows about – you must catch them or program fails to compile
  - Unchecked exceptions: the compiler doesn't know about them. You need to catch for good running of program
- Which kind does python have? (or both?)
  - unchecked

Checked Exceptions

- Are checked exceptions a good thing?
  - Why or why not? (it has been debated -though your book believes the debate is done.
  - Lets discuss

Exception context

- Exception objects have default descriptions
  - And of course correct exception types help too
  - If you throw exception base class you belong in the dunk tank
    - Yes we've all done it for academic code
  - But every exception allows a string parameter – make it clear what went wrong!!

Special Case Pattern

- When you would normally return an object
  - Sometimes there is an error or special case – can't return usual object
  - Return special case object
  - Create object which also extends abstract base class or implements interface to handle special case.

NULL what is it good for

- Null null nil NULL
  - Its all a bunch of nothing
  - Book – don't return null.
  - I like this rule. Requires more work on the part of the method
    - Must detect that null would be returned
      - Null just becomes another error code
      - What should be do instead?
    - But it buys you better code

Passing null

- Now this is just the mark of a bad programmer
  - Personal admission story
  - And so what do you think happened?

Passing null

- Now this is just the mark of a bad programmer
  - Personal admission story
  - And so what do you think happened?
  - Yup – compiled just fine – then a great big steaming pile of crash!

Reading

- Read chapter 7 in clean code.