Debugging

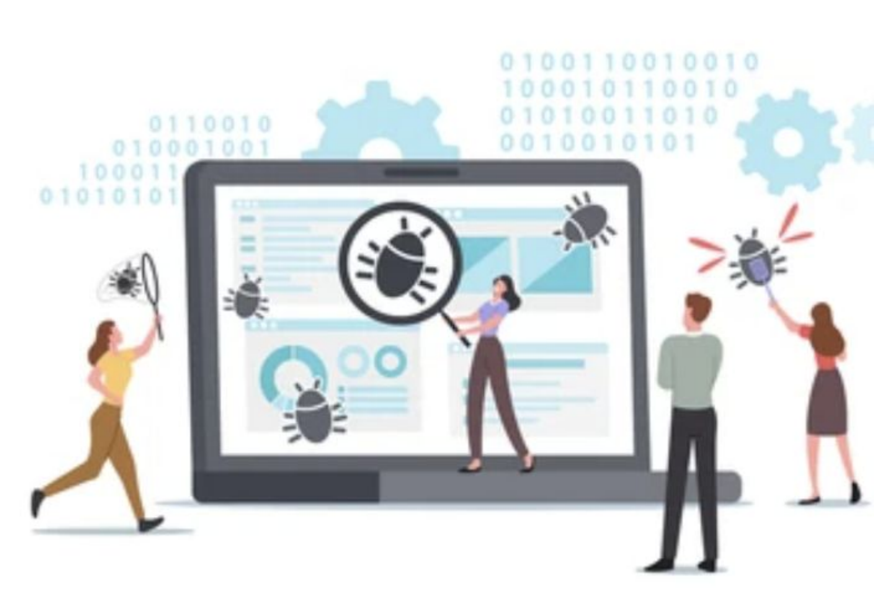
CS101 - Allen Jue

What is debugging?

Debugging is a systematic way to identify and remove bugs from your code.

It comes in a few flavors:

- Prints/Logs
- Debugger
- Examining the stack trace
- Rubber Duck :)
- And more!



Prints and Logs

Advantages:

- Basic way to do debugging, pretty good

Disadvantages:

- Is a “brute force” way to do debugging
- Not a silver bullet

Advice:

- It’s OK to use peppered in your code
- Craft meaningful debugging messages
- Ex: log show –last 10s

Debugger

A debugger is a set of tools provided to make finding and removing bugs easier.

Ex: GDB

Advantages:

- Can make it much easier to *step* through a program
- Offers flexibility and precise approach to debugging
- Gives you a lot of tools

Disadvantages:

- Still can be tedious
- Does not always reveal logical errors

GDB tools

Common commands:

- To start: use ‘gdb *executable*’ then ‘run’
- ‘break *file line #*’ - sets a breakpoint or just ‘b *file line*’
- ‘disable *breakpoint #*’ - removes a specified breakpoint
- ‘next’ or ‘n’ - goes to next breakpoint
- ‘step’ or ‘s’ - fine-grained, goes to next line
- ‘print *expression*’ - print some variable

Stacktraces

Report of active stack frames during a program’s execution.

In particular, it’s basically a print of the current process’s function call stack

Advantages:

- It tells you exactly where your program exploded
- Sometimes the problem occurred elsewhere, stacktrace provides a ‘map’
- Very detailed description and gives description of error

Disadvantages:

- Can be daunting to navigate initially
- Issues may occur in multiple places

Anatomy of a stacktrace

```
Exception in thread "main" java.lang.NullPointerException
        at com.example.myproject.Book.getTitle(Book.java:16)
        at com.example.myproject.Author.getBookTitles(Author.java:25)
        at com.example.myproject.Bootstrap.main(Bootstrap.java:14)
```

error

Call stack

Any other ways?

Sometimes, instead of heading straight into the code and debugger, it may me useful to just **talk!**

Rubberducking

Rubberducking is a debugging method where you articulate your code to an inanimate object or someone else

Advantages:

- You may say something that trips a mental alarm that an error has occurred
- If you talk to someone else, you can get collaborative feedback
- You are forcing yourself to **doubt** the validity of your code

Disadvantages:

- Sometimes bugs need to be approached in more detail (need to see values)
- You might seem crazy!

GDB Tutorial

We’re going to try and debug a bubble sort algorithm.

You can find the files at https://github.com/AllenJue/Debugging-Tutorial

Questions?

Thanks for coming!