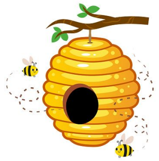


<- This means this lab is on the hive

Lab 2: C Debugging

61c Fall 23

QR code for these slides

Please do this lab in order according to the spec

- Exercises 2, 3, 4 are on the same file and if you do more than the instructions ask you to do for any of the exercises it might mess up the autograder

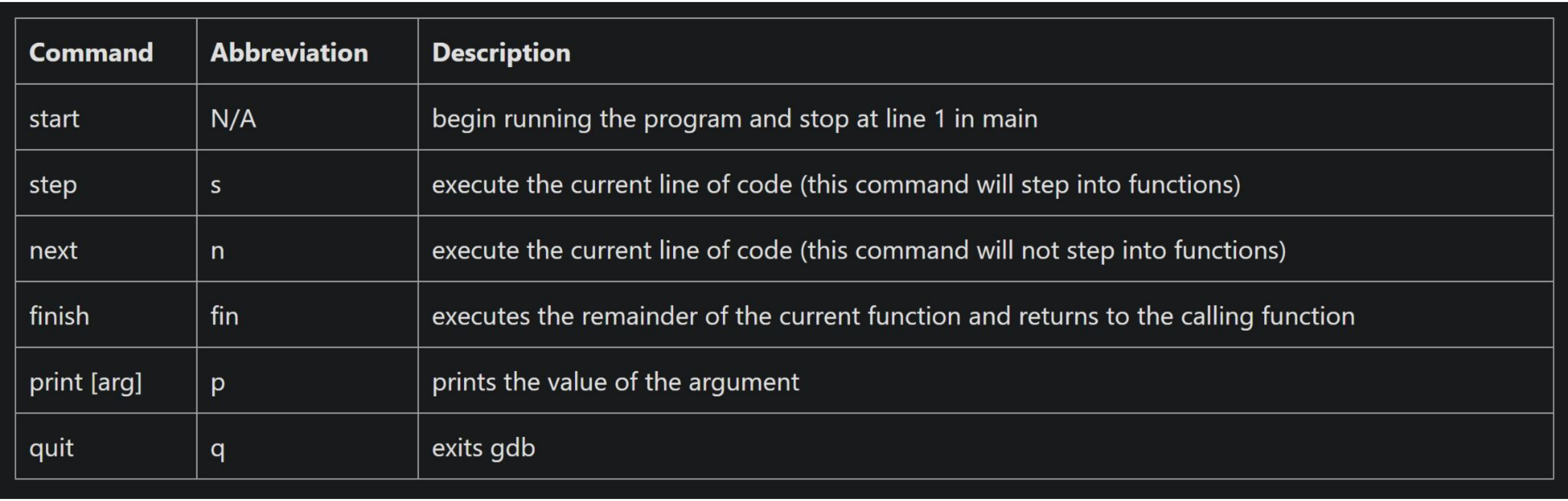
Compiler Warnings

- Compiler warnings are generated to help you find potential bugs in your code
- Make sure that you fix all of your compiler warnings before you attempt to run your code
    - This will save you a lot of time debugging in the future because fixing the compiler warnings is much faster than trying to find the bug on your own.

GDB

- GDB, the GNU Project debugger, allows you to see what is going on 'inside' another program while it executes -- or what another program was doing at the moment it crashed.
- In this class, we will be using CGDB which provides a lightweight interface to gdb to make it easier to use.

GDB Commands From Lab Spec

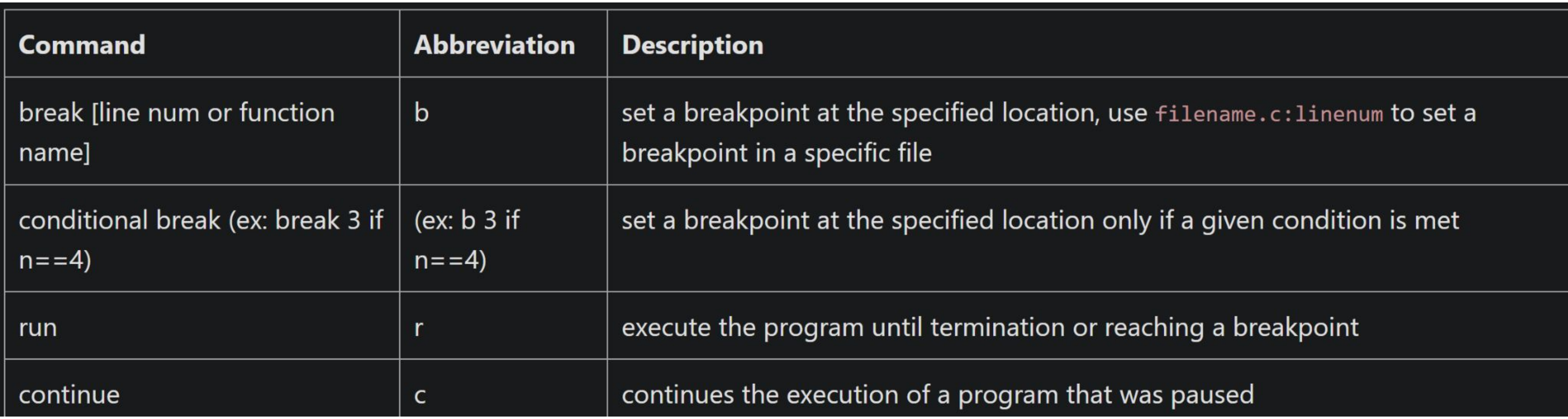


Command	Abbreviation	Description
start	N/A	begin running the program and stop at line 1 in main
step	s	execute the current line of code (this command will step into functions)
next	n	execute the current line of code (this command will not step into functions)
finish	fin	executes the remainder of the current function and returns to the calling function
print [arg]	p	prints the value of the argument
quit	q	exits gdb

Some Notes

- You can repeat some commands with numbers
  - If you wanna go 3 lines ahead you can do n 3
- The print feature is very versatile
  - It can print strings, numbers, pointers, etc

GDB Commands Continued



Command	Abbreviation	Description
break [line num or function name]	b	set a breakpoint at the specified location, use `filename.c:linenum` to set a breakpoint in a specific file
conditional break (ex: break 3 if n==4)	(ex: b 3 if n==4)	set a breakpoint at the specified location only if a given condition is met
run	r	execute the program until termination or reaching a breakpoint
continue	c	continues the execution of a program that was paused

Valgrind

- A tool for debugging “bohrbugs” and “heisenbugs”
- Allows you to see invalid reads/writes
- Checks for memory leaks
    - --leak-check=full