Quick Reference Guide to gdb

**Common gdb commands:**

- `run` -- run the program
- `run args` -- run program with command line args.
- `break function` -- set breakpoint at function entry
- `break linenum` -- set breakpoint at line
- `break *addr` -- set breakpoint at address
- `break … if cond` -- set breakpoint; break if condition
- `clear funct` -- remove breakpoint at function entry
- `delete bnum` -- delete breakpoint bnum
- `disable bnum` -- disable breakpoint bnum
- `enable bnum` -- enable breakpoint bnum
- `condition bnum` -- set conditions for breakpoint bnum
- `commands bnum` -- set commands for breakpoint bnum
- `cont` -- continue execution to next break point
- `next` -- step next source level statement or function
- `nexti` -- step next machine instruction or function
- `step` -- step next source level statement
- `stepi` -- step next machine instruction
- `print expr` -- print value of expression including $n for machine registers
- `print/f expr` -- print value of expression according to format specified by f: x hexadecimal, d decimal, u unsigned decimal, o octal, a address, c character, f single precision floating point.
- `x/sf addr` -- Examine memory of size s bytes in format f: s = b one byte, s = h halfword, s = w word, s = g double word; x hexadecimal, d decimal, u unsigned decimal, o octal, a address, c character, f single precision floating point, s ascii string, I machine instruction
- `display/f expr` -- p/sf, print every gdb command
- `display/sf expr` -- x/sf, examine every gdb command
- `undisplay n` -- remove item n from display list.
- `jump *addr` -- execute next instruction at address addr.
- `printf string, expr` -- formatted output, similar to printf in C but without the parentheses surrounding the arguments.
- `info data` -- information about break, display, registers, functions, variables
- `list` -- list ten source lines
- `where` -- show call stack
- `q` -- exit gdb

Note: pressing return repeats the last command.