```
Starting:
        gdb
        gdb <file>
        gdb -h          (lists command line options)

Exiting:
        quit
        Ctrl-d
        Note: Ctrl-C does not exit from gdb, but halts the current
                gdb command


General commands

        run             (start your program)
        kill            (stop the program)


Breakpoints

        break FUNCTION  (set a breakpoint at the entry to the function)
        break *ADDRESS  (set a breakpoint at the specified address)

        disable <NUM>   (disable the breakpoint with that number)
        enable <NUM>    (enable the breakpoint with that number)

        clear FUNCTION  (clear any breakpoints at the entry to the function)

        delete <NUM>    (deletes the breakpoint with that number)
        delete          (deletes all breakpoints)


Working at breakpoints

        stepi           (execute one instruction)
        stepi <NUM>     (execute NUM instructions)

        nexti           (like stepi, but proceed through subroutine calls)
        nexti <NUM>

        until LOCATION  (continue running until LOCATION is reached)

        continue        (resume execution)
        continue <NUM>  (continue, ignoring this breakpoint NUM times)

        finish          (run until the current function returns)

        backtrace       (print the current address and stack backtrace)
        where           (print the current address and stack backtrace)

Examining code

        print/a $pc     (print the program counter)
        print $sp       (print the stack pointer)
        disas           (display the function around the current line)
        disas ADDR      (display the function around the address)
        disas ADDR1 ADDR2 (display the function between the addresses)

Examining data

        print $eax      (print the contents of %eax)
        print/x $eax    (print the contents of %eax as hex)
        print/a $eax    (print the contents of %eax as an address)
        print/d $eax    (print the contents of %eax as decimal)
        print/t $eax    (print the contents of %eax as binary)
        print/c $eax    (print the contents of %eax as a character)
```

```
        print 0x100     (print decimal repr. of hex value)
        print/x 555     (print hex repr. of decimal value)

        x ADDR          (print the contents of ADDR in memory)
        x/NFU ADDR      (print the contents at ADDR in memory:
                                N = number of units to display
                                F = display format
                                U = b (bytes), h (2 bytes), w (4 bytes))

Autodisplaying information

        display $eax    (print contents of %eax every time the
                        program stops)
        display         (print the auto-displayed items)
        delete display <NUM>    (stop displaying item NUM)

Useful information commands
        help info
        info program    (current status of the program)
        info functions  (functions in program)
        info stack      (backtrace of the stack)
        info frame      (information about the current stack frame)
        info scope      (variables local to the scope)
        info variables  (global and static variables)
        info registers  (registers and their contents)
        info breakpoints (status of user-settable breakpoints)
        info address SYMBOL     (use for looking up addresses of functions)

Running gdb in emacs
        M-x gdb
        C-h m to see the features of GDB mode
```