**Variables**

* What is a variable?
  + An object that holds data that can be changed
* What is variable scope?
  + The portion of a program where a value is known or available
* Local vs Static vs Global
  + Local is only available in the function it is declared in
  + Static is local inside of its functions but global when declared outside of functions
  + Global is available throughout the entire file system and all functions within
* What are variable types?
  + The type of data that is stored in a variable such as Char or Int
* What is a pointer?
  + A variable that stores the memory address of another variable in memory
  + What can a pointer point to?
    - Any variable or function that matches its declared data type
* What is an array?
  + An object that holds a group of similar variable types
* What is a string?
  + An object that holds a group of char variables
* What is casting?
  + Typecasting is when you change a variable from one data type to another

**Functions**

* What is a function prototype?
  + The declaration of a function that declares its name and type
* How can you tell the return type of a function?
  + A functions return type should match its function declaration
* How do you pass arguments to a function?
  + By adding your arguments to the function prototype
* What is a struct?
  + Similar to an array it can hold a group of variables but they do not have to have the same data type like arrays

**Computer Architecture**

* What is a register?
  + Similar to variables they hold data for processing
* What are the x86 registers and what are they used for?
  + 8 general purpose registers that are used for address calculations, arithmetic, and logical calculations
* What is meant by little endianness?
  + It describes the idea that the least significant bytes are stored before the most significant bytes

**Assembly language**

* What is assembly language?
  + A low-level machine language that is meant to communicate directly with a machines hardware
* What are the sources for the values used in an assembly instruction?
* What happens when the call instruction executes?
  + The call instruction puts the data onto the stack
* What is a function prologue?
  + It is typically a few lines of code at the beginning of the function which prepare the stack for use
* How are function arguments accessed?
  + By referencing the ESP
* How are local variables accessed?
  + By referencing the EBP

**Concepts of a running process**

* What sections is a program divided into?
  + Data section
  + BSS section
  + Text section
* What kind of data is stored in each section?
  + Data declares data or constants
  + BSS declares variables
  + Test section keeps actual code
* What is the stack and how does it work?
  + An array like structure in memory where data can be stored and removed
* What is a stack frame?
  + An organization structure that helps organize local data for function calls
* How can you view the details of a stack frame in gdb?
  + Using the backtrace command

**Misc**

* What is a file descriptor?
  + A number that identifies an files within Unix
* What is a format parameter?
  + Specifies whether to return the original file to the repository rather than running the file