struct Exam 2	array Exam 2	-> Exam 2
padding Exam 2	static (variable) Exam 2	static (function) Exam 2
passed by reference Exam 2	passed by value Exam 2	Makefile Exam 2
l-value of a variable Exam 2	r-value Exam 2	NULL pointer location Exam 2
malloc Exam 2	calloc Exam 2	free Exam 2
function pointer that can point to strcat Exam 2	interpreted language Exam 2	compiled language Exam 2

3	2	1
Operator to access a member of a struct from a pointer to that struct	A homogenous collection, accessed by index	A heterogenous collection, accessed by member name

6	5	4
Modifier which makes the function "private" to that file	Modifier which has the variable be maintained for the lifecycle of the program	Used to increase the size of a struct to allow members to align with the multiple of their data type size

9	8	7
Specifies how to create a project including dependencies and other commands	How all other data types other then arrays are passed into functions	How arrays are passed into functions

12	11	10
Location 0 in memory	The data held by a variable	The address of the variable

15	14	13
Function to deallocate space in the heap	Function to grap initialized data (all 0's) from the heap	Function to grap uninitialized data from the heap

18	17	16
Language in which program translation takes place all at once before execution	Language in which program translation takes place line by line when running	char *(*cat)(char *, char*);

strongly typed language	weakly typed language	statically typed language
Exam 2	Exam 2	Exam 2
dynamically typed language		
Exam 2		

A language where type checking takes place at compilation

A language that allows flexibility with its type rules

A language that does not allow operations between different data types

A language where type checking takes place when the program is running