

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

| | | | |
|--|--|---|----|
| | 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 than 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*); | |

| | | |
|---|--|--|
| <div>strongly typed language</div> <div>Exam 2</div> | <div>weakly typed language</div> <div>Exam 2</div> | <div>statically typed language</div> <div>Exam 2</div> |
| <div>dynamically typed language</div> <div>Exam 2</div> | <div>union</div> <div>Exam 2</div> | <div>heap</div> <div>Exam 2</div> |
| <div>text</div> <div>Exam 2</div> | <div>stack</div> <div>Exam 2</div> | <div>data</div> <div>Exam 2</div> |
| <div>fgets</div> <div>Exam 2</div> | <div>fgetc</div> <div>Exam 2</div> | |

21

20

19

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

24

23

22

Segment in memory where dynamically allocated memory is located

A data type in C in which all member elements share the same space in memory

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

27

26

25

Segment in memory where global memory is stored

Segment in memory where data for a function call is located

Segment in memory where executable code is located

29

28

Function to get a character from a file

Function to read a line from a file as a string