Creating Multidimensional Arrays

Initialized and uninitialized multidimensional arrays are created as follows:

```
initializedMultArray = type arr[][dim2Size]…[dimNSize]
= {{element1, element2,…},{element1, element2, …}, …};
uninitializedMultArray = type arr[dim1Size][dim2Size]…
[dimNSize];
```

strcat()

Two strings can be concatenated using the `strcat()` function.

Creating Strings

Strings can be created by initializing an array of `char`s.

String Length

Arrays are static, therefore the length of a string cannot be modified.

Iterating Through Arrays

Arrays can be iterated through using `while` loops or `for` loops.

What is an array?

An array is used to store many elements of the same type in contiguous blocks of memory

sizeof()

Array size can be found using the `sizeof()` function

Accessing Characters in a String

Characters in a string can be accessed and modified using indices, the same technique used with arrays.

Creating Uninitialized Arrays

An uninitialized array is created as follows:

```
type arr[array_size];
```

Null Character

All strings terminate with a null character (`'\0'`).

First and Last Array Elements

The first and last elements in the array can be found at the following indices:

```
firstElement = arr[0];
lastElement = arr[arraySize - 1];
```

Creating an Initialized Array

An initialized array is created as follows:

```
type arr[] = {element1, element2, element3, …};
```

strcpy()

A string can be copied into an empty `char` array (empty string) using the `strcpy()` function.

Accessing Array Elements

You can access the array element at index `idx` as follows:

```
arr[idx];
```

Invalid Array Access

Attempting to access or modify an element at an index greater than the length of the array will cause the program to behave unpredictably.

strlen()

You can find the length of a string using the `strlen()` function.