CH7

7.1

Arrays

Hold multiple values of the same type

Declarations

Data type, name, [size decelerator]

Size decelerators must be constants. Typically named constants.

Stored together in memory

Not necessarily in order

**Slide 20 very important**

**Local variables – stored in the stack frame**

**Stack writing outside the array – studio will error if we go out of bounds in stack**

**Arrays stored contiguously in memory**

**4 bytes per each element in an array**

7.2

Local – uninitialized by default

Global – initialized by default

Input one element at a time, using the slots index

7.3

7.4

Initialization list {initializer, initializer, initializer};

If we have more numbers than slots, then compile time error

If we use a global array with less numbers than slots, rest are zero

Array vs C String

Char str = { ‘c’ , ‘a’ , ‘t’ }

Char str = { ‘c’ , ‘a’ , ‘t’ , ‘\0’ }

7.6

Sending the name of a char array to console out will print the whole array

Only exception

7.8

Main()

{

Int tests[20]; tests contains address of first element

functionDefinition (tests) function call on variable tests

}

functionDefintion (int arr[]) address pass by value, **we don’t change**

arr[3] = 10; we **ARE CHANGING A ARRAY VALUE**

arr[3] = 10; arr address [ initial address + (3 \* the data type)] = ten

Slide 71 know how the code changing the numbers works

7.9

Homework Notes

Program 1

Program 2

Use multiple functions

Parallel Array’s

Load Student Data into Two Dimensional Array

Modulus

Processing Loop

Student Processing loop

Quiz Notes

Main()

{

Int tests[20]; tests contains address of first element

functionDefinition (tests) function call on variable tests

}

functionDefintion (int arr[]) address pass by value, **we don’t change**

arr[3] = 10; we **ARE CHANGING A ARRAY VALUE**

arr[3] = 10; arr address [ initial address + (3 \* the data type)] = ten

**int size = 20;**

**int array [size]; won’t wory, must be constant**

**size decelerator types -**

**Subscripts types-variables, expressions (result must be between 0 and size minus 1)**

**Out of bounds**