

4D > 4B > 4B > 4B > B 990

• Declare a structure using the typedef struct syntax: typedef struct { 1 2 int anInt; 3 double aDbl; 4 int anotherInt; 5 char aString[20]; } aStructure; • aStructure is the name of a new type

Structure I

Structures

Nebraska

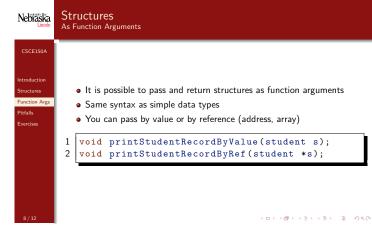
Nebraska

- Can now declare variables of type aStructure just like you'd declare a variable of type int, etc.
- Each variable of this type has 4 simple data type components

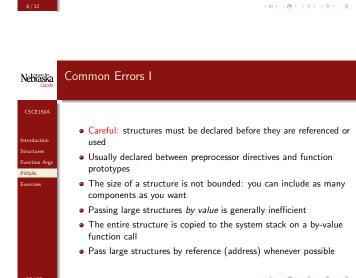
• Declare an instance of a structure like you would an atomic type: aStructure anExampleOfAStructure; aStructure anArrayOfStructures[10]; • To set or access the individual member variables we use the following syntax:

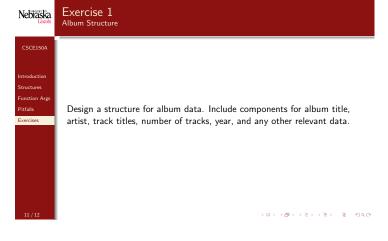
- myStructure.memberVariable
- Known as component selection operator
- Examples: anExampleOfAStructure.aDbl = 6.5 anArrayOfStructures[3].anotherInt = 4

10 > 10 > 12 > 12 > 12 > 2 900



Returning Structures CSCE150A Introduction Structures Function Args Pictalls Exercises As with arrays, we cannot return structures that are local in scope We must return a pointer to a dynamically allocated structure We haven't covered this in class, but it is in the textbook





A complex number consists of two real numbers: a real component and an imaginary component
 Complex numbers are able to handle roots of negative numbers: Examples:

Exercise 2

Nebraska

 $\sqrt{-4} = 0 + 2i$ $\sqrt[4]{-4} = 1 + i$

• Define a structure to handle complex numbers. Write a function to print them in a nice format.