CS 161
Intro to CS I

Arrays vs. Structs

0. Exercises are entered - make sure you have grades 1-7
1. Tests are entered
2. Duncan is 001
   Connor is 002
3. Be here Fri.
4. No office hours for Jennifer on Fri.
5. Assign #5 is due tomorrow!
6. Assign #6 released today!

OSU Oregon State University
Structures

• Data Structures So Far...
  – Variables simple
  – Arrays little more complex

• What if we want mixed types?
  – Record: name, age, weight, etc.
  – Use struct type user-defined types
Struct/Members

```c
struct doc_record {
    char name[50];
    int age;
    float weight;
};
```

- What does this do?
- How do we use it?

*Keyword: defining as a user-defined type*
struct doc_record{
    char name[50];
    int age;
    float weight;
}; //creates a user defined type, doc_record

int main() {
    doc_record jen; //use it as a type
    ...
}
using namespace std;

// This defines the user-defined type of doc_rec that contains three pieces of info of mixed types
struct doc_rec {
    char name[25];
    int age;
    float weight;
};

int main(){
    struct doc_rec austin; // create a doc_rec structure called austin

    cout << "Enter name: ";
    cin >> austin.name;
    cout << "Enter age: ";
    cin >> austin.age;
    cout << "Enter weight: ";
    cin >> austin.weight;

    cout << austin.name << endl;
    cout << austin.age << endl;
    cout << austin.weight << endl;

    return 0;
}
Why is it good to have an array of structs?

• What happens if you have two arrays with first names and last names, and you want to sort by first name?

• What happens if you put the first name and last name in a struct?
Things to think about…

- Describe an example structure you might define.
- How would you return a struct from a function?
- How would you pass a struct to a function?
- How would you create an array of these structures?
Returning Pointers, Arrays, Structs…

```cpp
int * create_1darray(int );

int main() {
    int *array;
    ...
    array = create_1darray(5);
    ...
}

int * create_1darray(int n) {
    return new int[n];
}
```
Returning Pointers, Arrays, Structs...

```c
int ** create_2darray(int, int);

int main() {
    int **array;
    ...
    array = create_2darray(5, 5);
    ...
}

int ** create_2darray(int n, int m) {
    //What goes in here?
}
```
Returning Pointers, Arrays, Structs...

```cpp
struct contact_info {
    std::string name;
    std::string address;
    unsigned int phone;
};
...
int main() {
    contact_info address_book[50];
    ...
    address_book[0] = create_contact();
    ...
}
contact_info create_contact() {
    contact_info contact;
    contact.name = "Jennifer";
    return contact;
}
```
What about passing structs to functions?

```cpp
#include <iostream>

using namespace std;

// This defines the user-defined type of doc_rec that contains three pieces of info of mixed types
struct doc_rec {
    char name[25];
    int age;
    float weight;
};

// We can return a doc_rec struct from a function
doc_rec set_info() {
    doc_rec person;

    cout << "Enter name: ";
    cin >> person.name;
    cout << "Enter age: ";
    cin >> person.age;
    cout << "Enter weight: ";
    cin >> person.weight;

    return person;
}
```
What about passing structs to functions?

```c++
24 } // We can pass the address of (C style) or by reference (C++ style)
25 // to change the contents of the structure austin in main
26 void set_info(doc_rec *person){
27     cout << "Enter name: ";
28     cin >> person->name; // same as (*person).name for dereference
29     cout << "Enter age: ";
30     cin >> person->age;
31     cout << "Enter weight: ";
32     cin >> person->weight;
33 }
34 int main(){
35     struct doc_rec austin; // create a doc_rec structure called austin
36     // austin = set_info(); // capture returned info into structure
37     set_info(&austin); // pass address of to change contents
38
39     cout << austin.name << endl;
40     cout << austin.age << endl;
41     cout << austin.weight << endl;
42     return 0;
43 }
```