CS 161
Intro to CS I

Structs vs. Classes
Odds and Ends...

• Assignment 6 questions
• Last week to demo Assignment 5
• Final Exam Wednesday
  – 001: 9:30-11:30am
  – 002: 12-2pm
```cpp
#include <iostream>

using namespace std;

struct book {
    char name[50];
    int version;
    int pages;
    double cost;
    char author[50];
};

int main() {
    book cs;

    cout << "name of book: ";
    cin >> cs.name;

    cout << cs.name;
    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?
```cpp
#include <iostream>

using namespace std;

struct book {
  char name[50];
  int version;
  int pages;
  double cost;
  char author[50];
};

int main() {
  book cs[10];

  for(int i=0; i<10; i++) {
    cout << "name of book: ";
    cin >> cs[i].name;
    cout << cs[i].name;
  }

  return 0;
}
```
Returning 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;
}
```
Passing Structs Demo...
Structs vs. Classes

- Structs only have state/attributes/member variables
- Classes have state/attributes/member variables

PLUS

- Classes have behavior/member functions
Class vs. Object

• Class is the definition.
• Object is the variable of a class type.
• Example:

class Point {
public:
    int x;
    int y;
};

int main() {
    Point p1, p2;

    p1.x=10;  p1.y=20;
    p2.x=5;  p2.y=6;

    return 0;
}
class Point {
public:
    int x;
    int y;
    void translate(int dx, int dy);  //Translates to a new x, y location given distance
};

int main () {
    Point p1, p2;
    p1.x=10;   p1.y=20;
    p2.x=5;   p2.y=6;

    p1.translate(-1, 3);
    p2.translate(2, -2);
    return 0;
}

void Point::translate(int dx, int dy) {
    x += dx;
    y += dy;
}
Can we set the values for x and y?

class Point {
public:
    int x = 0;  //This is not allowed!!!
    int y = 0;  //This is not allowed!!!
    void translate(int dx, int dy);  //Translates to a new x, y location given distance
};

int main () {
    Point p1, p2;
    p1.x=10;  p1.y=20;
    p2.x=5;   p2.y=6;

    p1.translate(-1, 3);
    p2.translate(2, -2);
    return 0;
}

void Point::translate(int dx, int dy) {
    x += dx;
    y += dy;
}
What if we made states private?

class Point {
public:
    void translate(int dx, int dy);
private:
    int x;
    int y;
};

int main () {
    Point p1, p2;
    p1.x=10;   p1.y=20;    //This is not allowed!!!
    p2.x=5;   p2.y=6;        //This is not allowed!!!

    p1.translate(-1, 3);
    p2.translate(2, -2);
    return 0;
}

void Point::translate(int dx, int dy) {
    x += dx;
    y += dy;
}
Encapsulation/ADTs

• How do we set private member variables?
• Accessor and Mutator Functions
• Access: get...
• Mutator: set...
Encapsulation

class Point {
public:
    void set_xy(int theX, int theY); //Mutator Function
private:
    int x;
    int y;
};

int main () {
    Point p1, p2;
    p1.set_xy(1, 1);
    p2.set_xy(2, 2);
    return 0;
}

void Point::set_xy(int theX, int theY) {
    x = theX;
    y = theY;
}