CS 162
Intro to CS II

Structs vs. Classes
Odds and Ends...

• Assignment #1 due Sunday, 11:59pm
  – Upload a assign1.tar to TEACH!!!
• Print Worksheet prior to recitation next week!
Unix Redirection Demo...

```
flip2 ~/cs162/private/sec2 169% arr_struct
OR
10
OR
10
4
3
2
1
0
flip2 ~/cs162/private/sec2 170% arr_struct < state.txt
Oregon
1000
4
3
2
1
0
flip2 ~/cs162/private/sec2 171%
```
Unix Redirection Demo...

```
flip2 ~/cs162/private/sec2 178% cat new.txt
Oregon
1000
4
3
2
1
0
flip2 ~/cs162/private/sec2 179% arr_struct < state.txt >> new.txt
flip2 ~/cs162/private/sec2 180% cat new.txt
Oregon
1000
4
3
2
1
0
Oregon
1000
4
3
2
1
0
flip2 ~/cs162/private/sec2 181%
```
Structs vs. Classes

• Differences
  - Structs do not contain functions
  - The members are public by default
  - Classes do contain functions
  - The members are private

• Similarities
  * You are making a user-defined type
  * They both contain members
  * The name by itself is the container, not the address
Class vs. Object

- Class declaration is type.
- Object is an instance of a class.
- 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 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

• Why do this?
• How do we set/get private member variables?
  – Accessor and Mutator Functions
• Accessor: get...
• Mutator: set...
Write the Mutator and Accessor Functions...

class Point {
public:
    //Accessor Functions
    //Mutator Functions
private:
    int x;
    int y;
};
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;
}
How do we write an Accessor Function?

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

int main () {
    Point p1;
    p1.set_xy(1, 1);
    std::cout << p1.get_x(); << "\t" << p1.get_y() << "\n";
    return 0;
}

int Point::get_x() {
    return x;
}

int Point::get_y() {
    return y;
}
Over the Weekend
Create a Date Class...

• Determine the members and their types for the date class.

• Write the accessor and mutator functions...