CS 162
Intro to CS II

Classes
Odds and Ends...

- Assignment 2 questions
- ACM-W meets tonight, Wed.
  - 5:30-6:30pm KEC 1005
- ACM meets tomorrow, Thurs.
  - 6-7pm KEC 1005
What is wrong with our library get_patron() accessor???
Objects, what’s wrong...

```c++
23  void print_pt(Points);
24  void print_pts(Points);
25
26  int main() {
27     Points pts;
28
29     print_pt(pts);
30     cout << pts.get_Point().get_x() << endl;
31
32     print_pts(pts);
33     for (int i=0; i<10; i++)
34         cout << pts.get_Points()[i].get_x();
35
36     return 0;
37 }
38
39  void print_pt(Points pts) {
40     cout << pts.get_Point().get_x() << endl;
41     pts.get_Point().set_xy(2, 2);
42     cout << pts.get_Point().get_x() << endl;
43 }
44  void print_pts(Points pts) {
45     cout << pts.get_Points()[0].get_x() << endl;
46     pts.get_Points()[0].set_xy(2, 2);
47     cout << pts.get_Points()[0].get_x() << endl;
48 }```

```cpp
class Points {
public:
    Point * get_Point();
    Point * get_Points();
private:
    Point pt;
    Point p[10];
};

void print_pt(Points &); void print_pts(Points);

int main() {
    Points pts;

    print_pt(pts);
    cout << pts.get_Point()->get_x() << endl;

    print_pts(pts);
    for(int i=0; i<10; i++)
        cout << pts.get_Points()[i].get_x();

    return 0;
}

void print_pt(Points &pts) {
    cout << pts.get_Point()->get_x() << endl;
    pts.get_Point()->set_xy(2, 2);
    cout << pts.get_Point()->get_x() << endl;
}
```

What is static?

• What is static?
  – Class variable or function
  – static int x; .... Point::x

• Can have a **static const int x=0**;
Static members...

- **Static variables:**
  
  ```cpp
  class math{
  public:
    static double pi;
  }
  double math::pi = 3.14;  //init once outside class
  int main() {
    math m, m1;
    m1.pi=2.0;  //since it isn’t constant, it can change
    cout << m.pi;  //changes for all members
    cout << m1.pi;
    cout << math::pi;
    return 0;
  }
  ```
Static members...

- **Static functions:**
  ```cpp
class math{
  private:
    static const double p = 3.14;
  public:
    static const double pi() {
      return p; // can only access static members
    }
  }

int main() {
  math m, m1;
  cout << m.pi();
  cout << m1.pi();
  cout << math::pi();
  return 0;
}
```
The Big “Three”

• If dynamic memory allocation in class, then...
  – Destructor
  – Copy Constructor
  – Assignment operator overload
What is a Destructor?

• Deallocate any member variable dynamically allocated...

• What would this destructor look like then?

```cpp
string::~string() {
    delete [] s; //delete ignores NULL
}
```
What is a copy constructor?

- Used in pass by value
- Returning an object from a function
- Pass the class type to a constructor

```cpp
string::string(const string &other) {
    len=other.len;
    if(len == 0) s=NULL;
    else {
        s=new char[len];
        for(int i=0; i<len; i++)
            s[i] = other.s[i];
    }
}
```