Pointers and classes

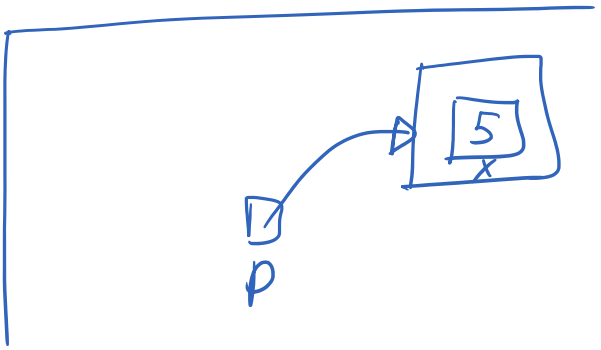
Friday, August 30, 2019 4:48 PM

- Pointer to Class

Example:

```
class Car{
public:
    int x
    void drive(…){…}
};

Car *p = new Car;
p->x = 5;
p->drive(…);
```



operator →

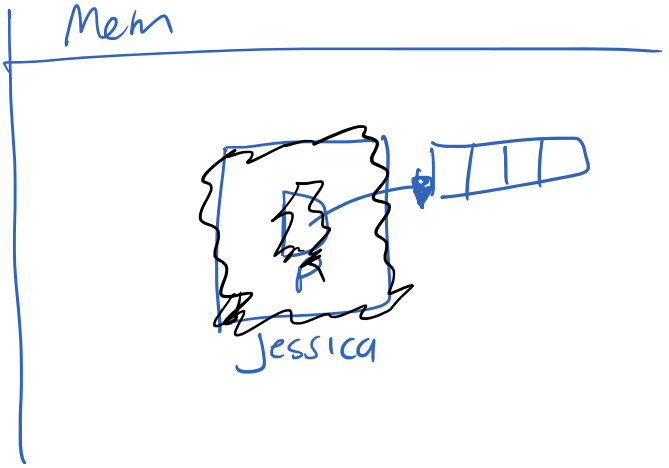
- Classes with Pointers

Example:

```
class Person {
public:
    int *p;
    Person(int n){
        p = new int[n];
    }
};

void foo()
{
…
 Person jessica(4);
…
}
```

what happens here?



- the this pointer

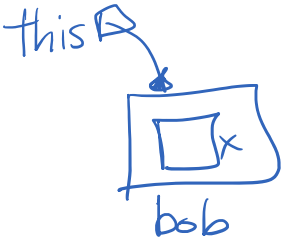
Example:

```
class IceCream{
public
    int x;
    void foo(IceCream* cont this, int n){
        x = n;
```

the this pointer points to the calling object of a member function

```
        this -> x = n;
    }
};

IceCream bob, alice, tim;
bob.foo(5);
```



- the "Big-3" Destructor, Copy Constructor, operator=

- Destructor :- called automagically when an object ends its life; goes out of scope {...}

- Copy Constructor :- called when an object is declared and initialized by other object

```
MyClass a = b;
MyClass a(b);
```

- when passing and returning objects by value.

```
foo ( MyClass c )
{
}
```

don't do

rather

```
foo(const MyClass &c)
```

```
MyClass a;
foo(a);
```

- operator equals

```
bob = tim.
```

```
bob = tim = fido = yogi;
```

Example:

```
class Dog{
public:
   int *p;
   Dog(){
      p = new int[3];
   }
   ~Dog() {}  //destructor

   Dog( const Dog &rhs ) : p(rhs.p) {} //copy-cons

   const Dog& operator=( const Dog &rhs )// operator=
```

automatic code

```
   Dog( const Dog &rhs ) : p(rhs.p) {} //copy-cons

   const Dog& operator=( const Dog &rhs )// operator=
   {
      if( this != &rhs ){  //alias test
         p = rhs.p;
      }
      return *this;
   }
};
```

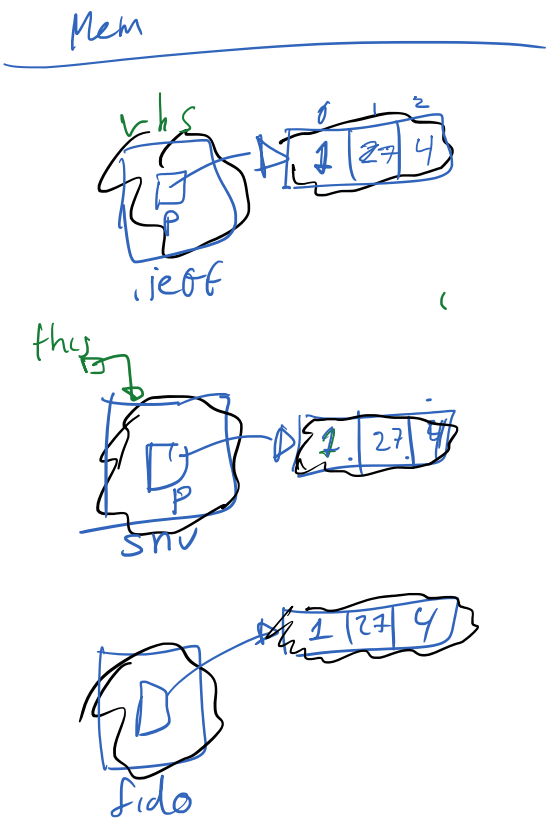
Example:

```
class Dog{
public:
   int *p;
   Dog(){
      p = new int[3];
   }
   ~Dog() {  //destructor
      delete [] p;
   }

   Dog( const Dog &rhs ) { //copy-cons
      p = new int[3];
      *this = rhs;
   }

   const Dog& operator=( const Dog &rhs )// operator=
   {
      if( this != &rhs ){  //alias test
         for(int k=0; k<3; k++)
            p[k] = rhs.p[k];
      }
      return *this;
   }
};

int foo(){
 … …
 Dog jeff;
 Dog fido;
 Dog snu = jeff;
 fido = jeff;
}
```



Whenever you have a class with pointer member, always consider overwriting the Big-3

Stay tuned to your e-mails, HW #1 to be posted soon.

= Dynamic Arrays"

int **p;

double ***** m;

Stay tuned to e-mails, HW #1 to be posted soon.

= Dynamic Arrays"

int **p;

double ***** m;