

Object-Oriented Programming *

Dr. Gerald Baumgartner

1 Objects:

```
struct S {  
    int x;  
    float y;  
};
```

```
S a;
```

```
a.x = 42;  
a.y = 3.14;
```

```
S* p = &a;
```

```
S* q = new S;
```

```
int i = p->x; // (*p).x;
```

```
int j = q->x;
```

*Typeset by Michael Miceli

2 Methods:

```
struct S {  
    int x;  
    float y;  
    int getX() {  
        return x;  
    }  
};
```

```
S* p = new S;  
p->x = 42; // (*p).x;  
p->y = 3.14; int i = p->getX(); // (*p).getX();
```

3 Constructors:

```
struct S {  
    int x;  
    float y;  
    S(int a, float b) {  
        x = a;  
        y = b;  
    }  
    int getX() {  
        return x;  
    }  
};
```

```
S* p = new S(42,3.14);  
int i = p->getX();
```

4 Visibility Specifiers:

```
struct S {
    private:
        int x;
        float y;
    public:
        S(int a, float b) {
            x = a;
            y = b;
        }
        int getX() {
            return x;
        }
}

S* p = new S(42,3.14);
int i = p->getX(); // (*p).getX();
int j = p->x; // error x is private
```

5 struct vs. class:

```
struct S {  
    private:  
    .  
    .  
    .  
};
```

is the same as

```
class S {  
    .  
    .  
    .  
};
```

6 struct vs. class:

```
class C {  
    public:  
    .  
    .  
    .  
};
```

is the same as

```
struct C {  
    .  
    .  
    .  
};
```

7 Inheritance:

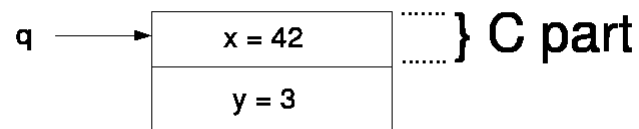
```
class C {
    int x;
public:
    C(int a) {
        x = a;
    }
    int getX(void) {
        return x;
    }
};

class D : public C {
    int y;
public:
    D(int a, int b) : C(a) {
        y = b;
    }
};

C* p = new C(42);
D* q = new D(42,3);
```

8 Subtyping:

```
class D : public C {...};  
C* q = new D(42,3);
```



A D object can be used where a C object is expected

9 Virtual Methods:

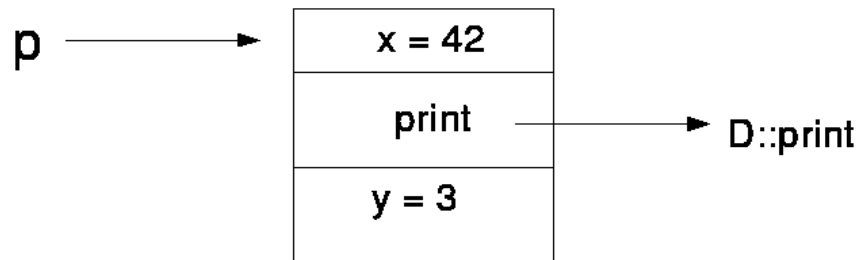
```
class C {
    int x;
public:
    C(int a) {
        x = a;
    }
    virtual void print(void) {
        cout << x;
    }
};

class D : public C {
    int y;
public:
    D(int a, int b) : C(a) {
        y = b;
    }
    virtual void print(void) {
        cout << '(' << x << ", " << y << ')';
    }
};

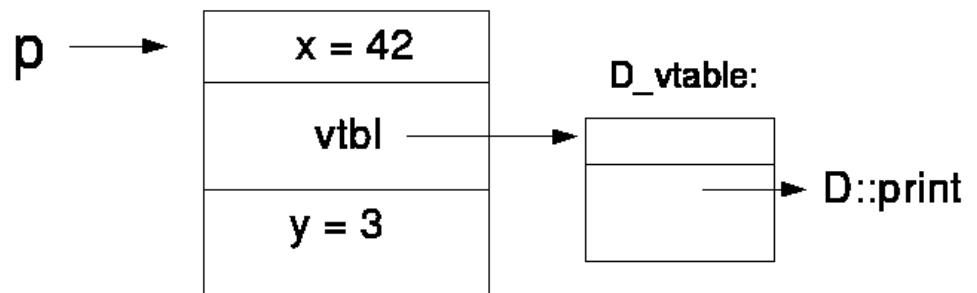
C* p = new D(42,3);
p->print();
```

10 Virtual Methods:

Conceptually:



Actual Implementation:



`p->print() → ((*p->vtbl())[1])(p)`