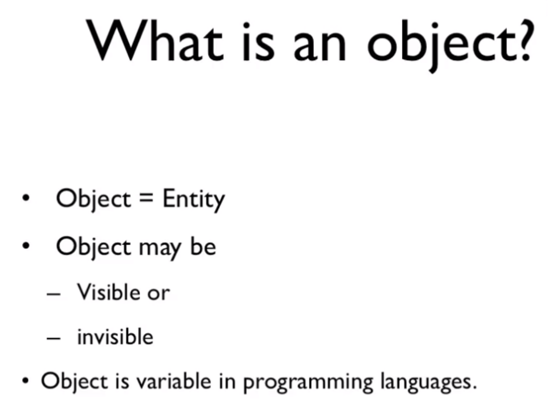
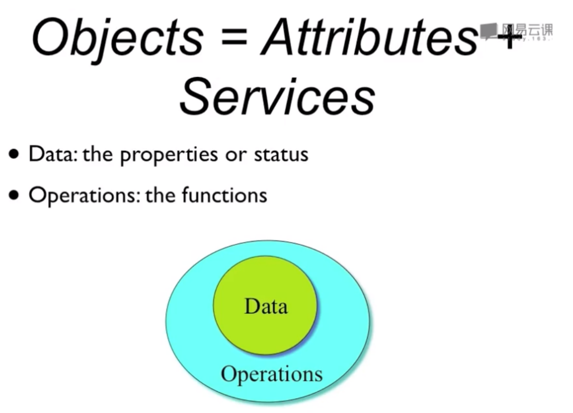
面向对象



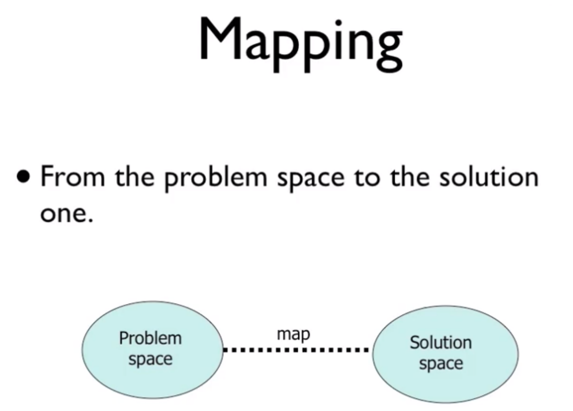
What is an object?

- Object = Entity
- Object may be
  - Visible or
  - invisible
- Object is variable in programming languages.



Objects = Attributes + Services

- Data: the properties or status
- Operations: the functions



Mapping

- From the problem space to the solution one.

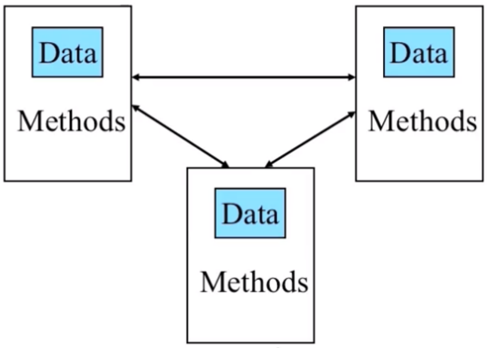
C vs. C++

```
typedef struct point3d {
    float x;
    float y;
    float z;
} Point3d;

void Point3d_print(const
Point3d* pd);

Point3d a;
a.x = 1; a.y = 2; a.z=3;
Point3d_print(&a);
```

```
class Point3d {
public:
    Point3d(float
x,float y,float z);
    print();
private:
    float x;
    float y;
    float z;
} ;

Point3d a(1,2,3);
a.print();
```

Object Oriented Programming

- Objects send and receive messages (objects do things!)

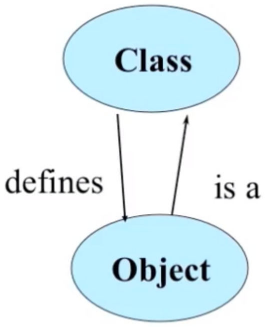


Objects send messages

- Messages are
  - *Composed* by the sender
  - *Interpreted* by the receiver
  - *Implemented* by methods
- Messages
  - May cause receiver to change state
  - May return results

Object vs. Class

- Objects (cat)
  - Represent things, events, or concepts
  - Respond to messages at run-time
- Classes (cat class)
  - Define properties of instances
  - Act like types in C++



OOP Characteristics

1. Everything is an object.
2. A program is a bunch of objects telling each other what to do by sending messages.
3. Each object has its own memory made up of other objects.
4. Every object has a type.
5. All objects of a particular type can receive the same messages.