

## Basics of Classes.

- Way to define your own types  
(e.g. for complex numbers.)

Example: a class to hold a square:

```
class square {  
    public:
```

```
    // member variables:
```

```
    // what data is needed to
```

```
    // represent a square?
```

```
    double side; // side length
```

```
    double x; // x coord of center
```

```
    double y; // y coord of center
```

```
    // (assume squares aren't rotated...)
```

```
    // member functions
```

```
    double area();
```

```
    void translate(double x, double y);
```

```
};
```

---

```
int main() {
```

```
    square s;
```

```
    s.x = 0;
```

```
    s.y = 0;
```

```
    s.side = 10;
```

```
    }
```

```
}
```

How to write member functions?

Want: `s.area()` to return the area. How to write it?

Non-member func:

```
double area(square s) {  
    return s.side * s.side;  
}
```

Note: in a call like

`"s.area()"`

`this == &s`

member func:

```
double square::area() {  
    return (*this).side *  
           (*this).side;  
}
```

\*this  
↓ implicitly  
here.

// note: this is of type `square*`.

// Short hand:

`return this->side * this->side`

$\square \rightarrow \Delta \equiv (*\square). \Delta$

// Even shorter:

`return side * side;`

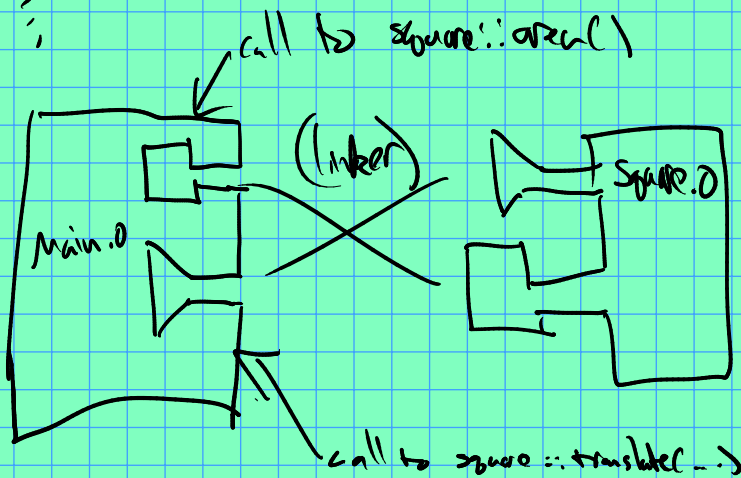
Note: the header file (containing the class defn.)  
is enough to compile something using `square`:

Say `main.cpp` uses `square`:

// `main.cpp`

#include "square.h"

;



## Constructors

You can write special functions that will be called automatically upon a variable's creation.

Syntax:

```
class square {  
    public:  
        square(); // ①  
        square(double ix, double iy, double iside); // ②  
};
```

// main.cpp

```
square s; // const. ① called  
square t(0,0,10); // const. ② called
```

---

Special constructor syntax:

if you are just setting member variables, do this:

```
square(double ix, double iy, double iside):  
    x(ix), y(iy), side(iside) {};
```

≡

```
square(double ix, double iy, double iside) {  
    x = ix; y = iy; side = iside; };
```