Object-oriented programming

Gianluca Campanella

Classes and objects

Class

- The blueprint for an object
- Encapsulates state and behaviour

Object

- An instance of a class
- Holds data and functions

Abstraction

Classes should...

- Provide only essential information to the external world
- Hide any internal implementation details

Inheritance and polymorphism

Classes can…

- Reuse code by inheriting from another class
- Redefine functions through polymorphism
- Define the meaning of operators through overloading

Defining a new class

Definition

```
class Rectangle {
    int width, height;
    public:
        Rectangle(int, int);
        int area(void);
}; // Mind the semicolon!
```

Defining a new class

Implementation

```
Rectangle::Rectangle(int w, int h) {
    width = w;
    height = h;
}

Rectangle::area() {
    return width * height;
}
```

Don’t overuse classes and objects!

- OOP can be overkill (especially for small programs)
- Classes make your code more verbose
- Try `struct`s if you just need ‘data containers’