Design Principles

1 Identify the aspects of your application that vary and separate them from what stays the same.

- ➔ This means: identify those parts in your software that will change (because of new requirements) and encapsulate them, so that you can change or extend them later.

2 Program to an interface, not an implementation

- ➔ This means: you use polymorphism by programming to a supertype so that the actual runtime object isn't locked into the code.

We know this kind of code:

```
Dog d = new Dog();
d.bark();
```

This is programming to an implementation.

Task: how could we change the code, so we are programming to an interface?
Hint: we don't want to hard-code the instantiation of dog object.

3 Favor composition over inheritance.

Using composition gives you much more flexible code. Why?