

ABSTRACTION

- concealing details of implementation
- using incomplete things to build other things
 - the incomplete things are hidden from the "user"
 - abstract classes + interfaces

ABSTRACT CLASSES

abstract class =
can't
instantiate! →

```
abstract class Animal
- props, methods, const's
abstract void speak();
```

```
class Toad extends Animal
void speak() { sayout "croak"; }
```

INTERFACES

- a contract - if you implement me, you must do x(), y(), z()
- USUALLY contains only abstract methods
- use "implements", not "extends"
- you CAN implement multiple interfaces

```
interface Student
```

```
void study();
```

```
interface Employee
```

```
void work();
```

```
class Mike imp's Student,
Empl.
```

```
void study() { }
```

```
void work() { }
```