Abstract Classes

A company has two kinds of employees – hourly workers who work 40 hours a week at a certain wage per hour, and salaried workers who work for an annual salary. The hourly workers get paid every week; the salaried workers once a month -- let’s say every fourth week.

I want to write a system that has a list of the company’s employees; each week it runs through the list looking at each employee’s data and printing a statement about how much that person should be paid.

How do we arrange the classes to make this easy?

Answer: Make a parent class Employee , with subclasses HourlyWorker and SalariedWorker. The staff list can be an ArrayList<Employee> or even Employee[ ]

Our payEveryone method will have a loop like this:

```
for (Employee x: staffList)
        ( <cast x into its right type>).pay()
```

If we give Employee a pay( ) method that the two subclasses override, then we don’t have to cast the list variable into appropriate subclass; the runtime environment will call the subclass’s method automatically.

Now, what body do we give the pay() method in class Employee?

Answer: we DON'T give it a body. This company has no generic employees, so we should never construct an element of the employee class. We make pay() an ***abstract*** method of the Employee class, which makes the class itself abstract.

The declaration of pay( ) is

```
public abstract void pay();
```

If a class is abstract it must be declared so:

```
public abstract class Employee
```

You cannot construct an object of an abstract class.

An abstract class must be extended by subclasses that override its abstract methods.

A class is abstract (and must be declared as such) if it has at least one abstract method.

- See example:
- Class Employee, SalariedWorker, HourlyWorker and StaffExample

Advantages of abstract classes:

1. They provide a common parent class for similar but distinct classes.
2. They force the subclasses to instantiate essential methods.
3. They provide a template that can be instantiated in very different ways for very different extending classes.