Abstract class

Abstract class

- A class which contains the **abstract** keyword in its declaration is known as abstract class.
- Abstract classes may or may not contain *abstract methods*, i.e., methods without body (public void move();)
- However, if a class contains at least one abstract method, then the class **must** be declared abstract.
- If a class is declared abstract, it cannot be instantiated.
- To use an abstract class, you have to inherit it from another class, provide implementations to the abstract methods in it.
- If you inherit an abstract class, you have to implement all the contained abstract methods.

Purpose of Abstract Classes

- An abstract class's purpose is to provide an appropriate superclass from which other classes can inherit and forced to have mandatory methods to share a common design.
- For example Shape class

Concrete classes

- Classes that can be used to instantiate objects are called **concrete classes**. Such classes provide implementations of *every* method they declare (some of the implementations can be inherited).

Abstract Methods

- If you want a class to contain a particular method but you want the actual implementation of that method to be determined by child classes, you can declare the method in the parent class as an abstract.

Contd.,

- **abstract** keyword is used to declare the method as abstract.
- You have to place the **abstract** keyword before the method name in the method declaration.
- An abstract method contains a method signature, but no method body.
- Instead of curly braces, an abstract method will have a semi colon (;) at the end.

Example

```
 public abstract class Employee
{
    private String name;
    private String address;
    private int number;

// Abstract method
public abstract double computeSalary();

// Rest of class definition
}
```

Consequences

Declaring a method as abstract has two consequences −

- The class containing it must be declared as abstract.
- Any class inheriting the current class must either override the abstract method or declare itself as abstract.

Suppose Salary class inherits the Employee class, then it should implement the **computeSalary()** method as follow example

Example

- public class Salary extends Employee

{

private double salary; // Annual salary public double computeSalary()

{

System.out.println("Computing salary for " +getName());

return salary/52;

}

// Rest of class definition }

Practice

Create the abstract class Employee with computSalary method.

Create concrete SalariedEmployee class and hourlySalariedEmployee.

Class SalariedEmployee extends class Employee and overrides *abstract* method earnings which will make SalariedEmployee a *concrete* class

- // abstract method must be overridden by concrete subclasses
- public abstract double earnings();
- // no implementation in super abstract class

Subclass SalariedEmployee

- // calculate earnings; override abstract method earnings in Employee
- @Override
- public double earnings()
- {
- return getWeeklySalary();
- }