

Abstract Classes

Spoken Tutorial Project

<http://spoken-tutorial.org>

National Mission on Education through ICT

<http://sakshat.ac.in>

Narration: Priya K

**Script: Joms Antony, Amal Jyothi College of
Engineering**



16 December 2015



Learning Objectives



Learning Objectives

We will learn about

- **Abstract Methods and Concrete Methods**



Learning Objectives

We will learn about

- Abstract Methods and Concrete Methods
- **Abstract Classes and Concrete Classes**



Learning Objectives

We will learn about

- Abstract Methods and Concrete Methods
- Abstract Classes and Concrete Classes
- **Using Abstract Classes**



Software Requirements



Software Requirements

- **Ubuntu 12.04**



Software Requirements

- **Ubuntu 12.04**
- **JDK 1.7**



Software Requirements

- **Ubuntu 12.04**
- **JDK 1.7**
- **Eclipse 4.3.1**



Prerequisite



Prerequisite

- **Basic knowledge of Java and Eclipse IDE**



Prerequisite

- Basic knowledge of Java and Eclipse IDE
- Knowledge of Subclassing



Prerequisite

- **Basic knowledge of Java and Eclipse IDE**
- **Knowledge of **Subclassing****
- **For relevant tutorials, please visit <http://www.spoken-tutorial.org>**



What is an Abstract Method?



What is an Abstract Method?

- **An abstract method is a method that is declared without an implementation**



What is an Abstract Method?

- An **abstract method** is a method that is declared without an implementation
- An **abstract method is declared using the abstract keyword**



What is an Abstract Method?

- An **abstract method** is a method that is declared without an implementation
- An **abstract method** is declared using the **abstract** keyword
- **No opening and closing parenthesis**



Abstract Method

Example:

```
//abstract method  
abstract public void showDetails();
```



Concrete Method

A concrete method is defined and implemented completely.

Example:

```
//concrete method  
public void showBasicDetails(){  
    //Implementation
```



Abstract Class



Abstract Class

- **An abstract class usually contains at least one abstract method**



Abstract Class

- An **abstract class** usually contains at least one **abstract method**
- **Abstract methods can be added only to abstract classes**



Abstract Class

Example:

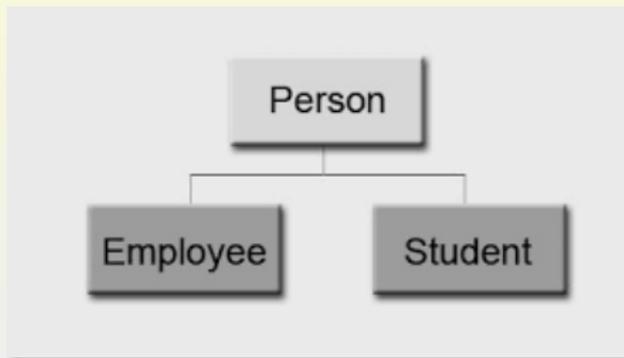
```
abstract public class Person{  
    // variables...  
    //concrete methods...  
    //abstract method(s)  
    abstract public void showDetails();  
}
```



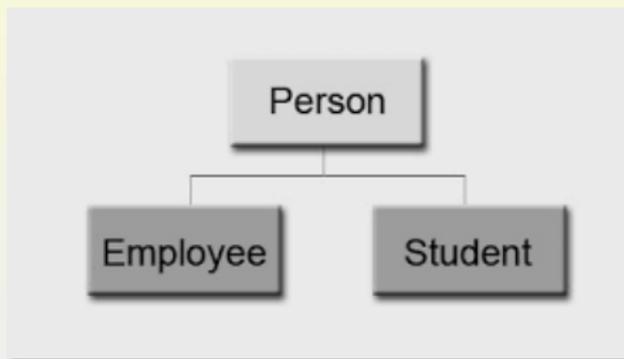
Abstract Class - Example



Abstract Class - Example



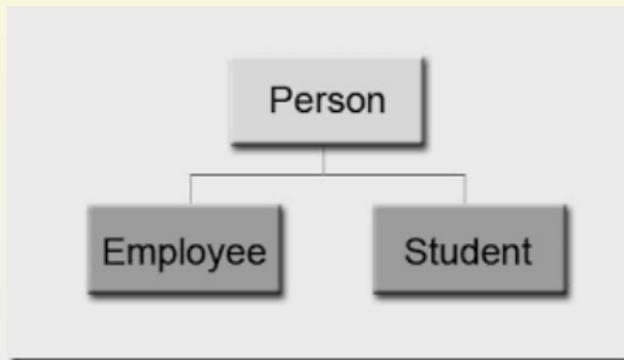
Abstract Class - Example



Emp ID, Salary



Abstract Class - Example



Emp ID, Salary



Stud Reg.no, Grade



Concrete Class

All the methods in a concrete class are to be concrete (completely implemented)

```
public class Employee{  
    // variables...  
    //concrete methods...  
    public void showDetails(){  
        //implementation  
    }  
}
```



Summary

- **Abstract Methods and Concrete Methods**
- **Abstract Classes and Concrete Classes**
- **How to use Abstract Classes**



Assignment

- Create an abstract class **Vehicle** which contains an abstract method **run()**



Assignment

- Create an abstract class **Vehicle** which contains an abstract method **run()**
- Create a subclass **Car** which



Assignment

- Create an abstract class **Vehicle** which contains an abstract method **run()**
- Create a subclass **Car** which
 - Extends the **Vehicle** class and



Assignment

- Create an abstract class **Vehicle** which contains an abstract method **run()**
- Create a subclass **Car** which
 - Extends the **Vehicle** class and
 - **Implements the run method that prints "Car is running on 4 wheels"**



Assignment

- Create another subclass **Bike** which



Assignment

- Create another subclass **Bike** which
 - Extends the **Vehicle** class and



Assignment

- Create another subclass **Bike** which
 - Extends the **Vehicle** class and
 - **Implements the run method that prints "Bike is running on 2 wheels"**



Assignment

- Create another subclass **Bike** which
 - Extends the **Vehicle** class and
 - Implements the **run** method that prints **"Bike is running on 2 wheels"**
- Create a **Demo** class containing the **main** method to verify the results



About the Spoken Tutorial Project

- Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- It summarises the Spoken Tutorial project
- If you do not have good bandwidth, you can download and watch it



Spoken Tutorial Workshops

The Spoken Tutorial Project Team

- Conducts workshops using spoken tutorials
- Gives certificates to those who pass an online test
- For more details, please write to contact@spoken-tutorial.org



Acknowledgements

- Spoken Tutorial Project is a part of the Talk to a Teacher project
- It is supported by the National Mission on Education through ICT, MHRD, Government of India
- More information on this Mission is available at <http://spoken-tutorial.org>



[/NMEICT-Intro](http://spoken-tutorial.org/NMEICT-Intro)



About the contributor

- This script has been contributed by:
Dept. of Information Technology
Amal Jyothi College of Engineering
<http://www.ajce.in/it>
- Thanks for joining

