6. Object Oriented Concepts and Techniques

Lesson 3:
Inheritance

6.8. Inheritance

- Inheritance is the concept of one class of objects inheriting the data and behavior from another class of objects.

- Inheritance is the creation of one class by extending another class so that instances of the new class automatically inherit the fields and methods of its parent class. Child classes can **override** methods in their parent classes.

- Through inheritance you can express the differences among related classes. At the same time, you share the functions and member variables that implement the common features.

- Inheritance promotes the class **Reusability**. Inheritance also helps you to reuse existing code from one or more classes by simply deriving a new class from them.

6.8. Inheritance
6.8.1. Example “Car”

- Class for a basic car

```
Class Car
{
        direction;
        position;
        speed;
        Seer();
        PressGasPedal();
        PressBreak();
}
```

- Suppose you want to create a new car which has a special passing gear

```
Class PassingCar inherits from Car
{
        Pass();
}
```

- Pass() is a method that implements the new functionality.

6.8. Inheritance
6.8.2. Inheritance in Java

- In Java the extends Keyword is used to achieve Inheritance

```
public class ThreeDimentionalPoint extends TwoDimentionalPoint
```

- A sub class can have its own implementation of the Super class methods (Method Overriding).

```
public class ThreeDimentionalPoint extends TwoDimentionalPoint
{
        int z;
        public void Show( )
        {
            super.Show();
            ...
        }
}
```

- The Show Method from super class is overridden. The Show method of the super class is accessed using the super keyword.
-
- Java does not support Multiple Inheritance.

6.8. Inheritance

6.8.3. Abstract Classes

- An Abstract Class is defined to be extended to sub classes uses abstract Keyword in the definition.

- Abstract classes cannot be instantiated.

```
abstract class Shape
{
abstract Point Center();
abstract double Diameter();
abstract double Area();
}
```