Java SE 8 Programming Basics

CS50 Seminar

Bill Zhang

Background

- Released by Sun Microsystems in 1995
- Object-oriented programming language
- Runs on Java Virtual Machine
- Java SE 8 released in 2014
- Java SE 9 released in September, 2017

Benefits of Java

- Object oriented
    - Procedural (C) and Functional (Python)
- Encapsulation
- Platform independent
- Robust
- Simple
- Secure
- It’s everywhere!

Downsides of Java

;

Roadmap

1. Basic Java Syntax
2. Good Class Design
3. Encapsulation

Object-Oriented Programming

```
public class Animal { }
```

Object-Oriented Programming

```
public class Animal {
    String name;
}
```

Object-Oriented Programming

```
public class Animal {
    String name;
}
```

Object-Oriented Programming

```
public class Animal {
    String name;
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
    public void setName(String newName) {
        name = newName;
    }
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
    public void setName(String newName) {
        name = newName;
    }
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
    public void setName(String newName) {
        name = newName;
    }
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
    public void setName(String newName) {
        name = newName;
    }
}
```

Fields and Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
    public void setName(String newName) {
        name = newName;
    }
}
```

Classes and Files

```
public class Animal { }
```

Classes and Files

```
public class Animal { }
```

Classes and Files

```
public class Animal { }
```

Classes and Files

```
public class Animal { }
```

```
$ javac Animal.java
```

```
$ java Animal
```

Will anything happen?

The main() Method

```
public class Zoo {
    public static void main(String[ ] args) {

    }
}
```

The main() Method

```
public class Zoo {
    public static void main(String[ ] args) {

    }
}
```

The main() Method

```
public class Zoo {
    public static void main(String[ ] args) {

    }
}
```

The main() Method

```
public class Zoo {
    public static void main(String[ ] args) {

    }
}
```

The main() Method

```
public class Zoo {
    public static void main(String[ ] args) {
        System.out.println(args[0]);
        System.out.println(args[1]);
    }
}
```

The main() Method

```
public class Zoo {
    public static void main(String[ ] args) {
        System.out.println(args[0]);
        System.out.println(args[1]);
    }
}
```

The main() Method

```
public class Zoo {
    public static void main(String[ ] args) {
        System.out.println(args[0]);
        System.out.println(args[1]);
    }
}
```

The main() Method

```
public class Zoo {
    public static void main(String[ ] args) {
        System.out.println(args[0]);
        System.out.println(args[1]);
    }
}
```

The main() Method

```
$ javac Zoo.java

$ java Zoo Bronx Zoo

Bronx
Zoo
```

The main() Method

```
$ javac Zoo.java

$ java Zoo “San Diego” Zoo

San Diego
Zoo
```

The main() Method

```
$ javac Zoo.java

$ java Zoo.java Zoo

ZooException in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 1
at mainmethod.Zoo.main(Zoo.java:7)
```

Primitives and Objects

```
String name = new String(“Bill Zhang”);
```

Primitives and Objects

```
String name = new String(“Bill Zhang”);
```

Primitives and Objects

```
String name = new String(“Bill Zhang”);
```

Primitives and Objects

```
String name = new String(“Bill Zhang”);
```

Primitives and Objects

```
String name = new String(“Bill Zhang”);
```

Primitives and Objects

```
int number = 5;
```

Primitives and Objects

```
int number = 5;
```

Primitives and Objects

```
int number = 5;
```

Primitives and Objects

```
int number = 5;
```

Primitives and Objects

boolean	true or false	true
byte	8-bit integral value	123
short	16-bit integral value	123
int	32-bit integral value	123
long	64-bit integral value	123
float	32-bit floating point value	123.45f
double	64-bit floating point value	123.45
char	16-bit Unicode value	‘a’

Constructors

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Constructors

```
$ javac Chick.java

$ java Chick
```

What will this print?

```
Tiny
```

Constructors

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public Chick(String name) {
        this.name = name;
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick(“Cutie”);
        System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
        private String name = "Fluffy";
        public Chick() {
                name = "Tiny";
        }
        public Chick(String name) {
                this.name = name;
        }
        public static void main(String[ ] args) {
                Chick chick = new Chick(“Cutie”);
                System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
	private String name = "Fluffy";
	public Chick() {
		name = "Tiny";
	}
	public Chick(String name) {
		this.name = name;
	}
	public static void main(String[ ] args) {
		Chick chick = new Chick(“Cutie”);
		System.out.println(chick.name); } }
```

Constructors

```
public class Chick {
        private String name = "Fluffy";
        public Chick() {
                name = "Tiny";
        }
        public Chick(String name) {
                this.name = name;
        }
        public static void main(String[ ] args) {
                Chick chick = new Chick(“Cutie”);
                System.out.println(chick.name); } }
```

Constructors

```
$ javac Chick.java

$ java Chick
```

What will this print?

```
Cutie
```

Declarations and Initializations

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Declarations and Initializations

```
public class Chick {
    private String name = "Fluffy";
    public Chick() {
        name = "Tiny";
    }
    public static void main(String[ ] args) {
        Chick chick = new Chick();
        System.out.println(chick.name); } }
```

Java Statements

if-then Statement

if-then-else Statement

while Statement

do-while Statement

for Statement

Java Statements

```
if (name == NULL) {
    System.out.println(“empty”);
}

for (int i = 0; i < 10; i++) {
    System.out.println( i );
}
```

Class Design

```
public class Animal {
    private int age;
    public int getAge() {
        return age;
    }
}
```

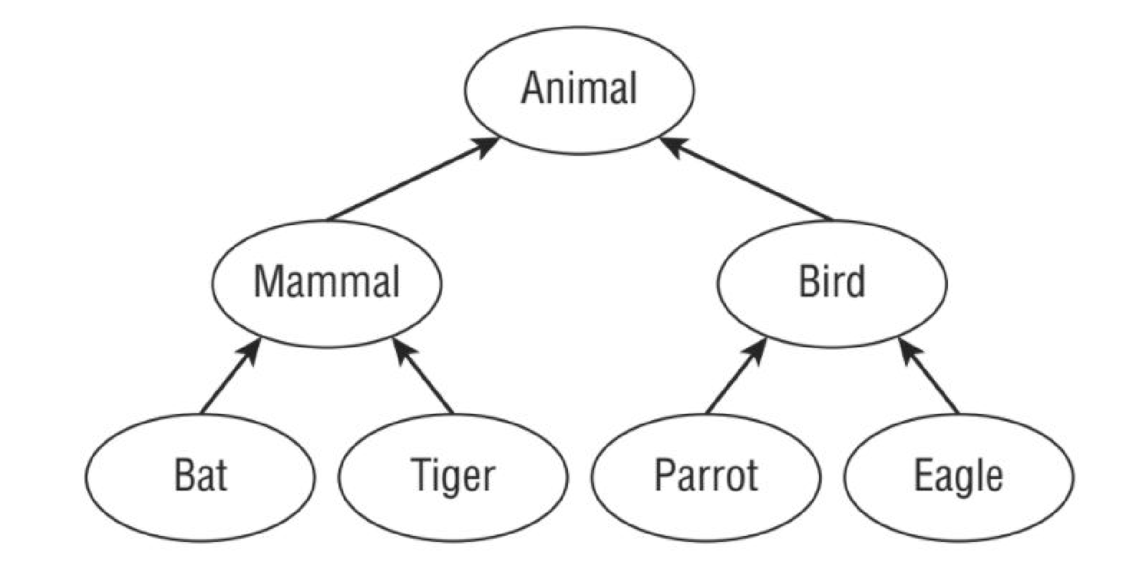
```
public class Tiger {
    private int age;
    public int getAge() {
        return age;
    }
    public void roar() {
        System.out.println(“roar”);
    }
}
```

```
public class Animal {
    private int age;
    public int getAge() {
        return age;
    }
}
```

```
public class Tiger {
    private int age;
    public int getAge() {
        return age;
    }
    public void roar() {
        System.out.println(“roar”)
    }
}
```

Class Inheritance

- Single inheritance



```
public class Animal {
    private int age;
    public int getAge() {
        return age;
    }
}
```

```
public class Tiger extends Animal {
    public void roar() {
        System.out.println(“roar”);
    }
}
```

```
public class Animal {
    private int age;
    public int getAge() {
        return age;
    }
}
```

```
public class Tiger extends Animal {
    public void roar() {
        System.out.println(“roar”);
    }
}
```

```
public class Animal {
    private int age;
    public int getAge() {
        return age;
    }
}
```

```
public class Tiger extends Animal {
    public void roar() {
        System.out.println(“roar”);
    }

    public void age() {
        int tigerAge = getAge();
        System.out.println(tigerAge);
    }
}
```

```
public class Animal {
    private int age;
    public int getAge() {
        return age;
    }
}
```

```
public class Tiger extends Animal {
    public void roar() {
        System.out.println(“roar”);
    }

    public void age() {
        int tigerAge = getAge();
        System.out.println(tigerAge);
    }
}
```

```
public class Animal {
    private int age;
    private String name;
    public Animal(int age, String name) {
        super();
        this.age = age;
        this.name = name;
    }
    public Animal(int age) {
        this.age = age;
        this.name = null;
    }
}
```

```
public class Gorilla extends Animal {
    public Gorilla(int age) {
        super(age,"Gorilla");
    }
    public Gorilla() {
        super(5);
    }
}
```

```java
public class Animal {
    private int age;
    private String name;
    public Animal(int age, String name) {
        super();
        this.age = age;
        this.name = name;
    }
    public Animal(int age) {
        this.age = age;
        this.name = null;
    }
}
```

```java
public class Gorilla extends Animal {
    public Gorilla(int age) {
        super(age,"Gorilla");
    }
    public Gorilla() {
        super(5);
    }
}
```

```
public class Animal {
    private int age;
    private String name;
    public Animal(int age, String name) {
        super();
        this.age = age;
        this.name = name;
    }
    public Animal(int age) {
        this.age = age;
        this.name = null;
    }
}
```

```
public class Gorilla extends Animal {
    public Gorilla(int age) {
        super(age,"Gorilla");
    }
    public Gorilla() {
        super(5);
    }
}
```

```java
public class Animal {
    private int age;
    private String name;
    public Animal(int age, String name) {
        super();
        this.age = age;
        this.name = name;
    }
    public Animal(int age) {
        this.age = age;
        this.name = null;
    }
}
```

```java
public class Gorilla extends Animal {
    public Gorilla(int age) {
        super(age,"Gorilla");
    }
    public Gorilla() {
        super(5);
    }
}
```

```
public class Animal {
    private int age;
    private String name;
    public Animal(int age, String name) {
        super();
        this.age = age;
        this.name = name;
    }
    public Animal(int age) {
        this.age = age;
        this.name = null;
    }
}
```

```
public class Gorilla extends Animal {
    public Gorilla(int age) {
        super(age,"Gorilla");
    }
    public Gorilla() {
        super(5);
    }
}
```

Abstract Classes

```
public abstract class Animal {
    protected int age;
    public void eat() {
        System.out.println("Animal is eating");
    }
    public abstract String getName();
}

public class Swan extends Animal {
    public String getName() {
        return "Swan";
    }
}
```

```
public abstract class Animal {
    protected int age;
    public void eat() {
        System.out.println("Animal is eating");
    }
    public abstract String getName();
}

public class Swan extends Animal {
    public String getName() {
        return "Swan";
    }
}
```

```
public abstract class Animal {
    protected int age;
    public void eat() {
        System.out.println("Animal is eating");
    }
    public abstract String getName();
}

public class Swan extends Animal {
    public String getName() {
        return "Swan";
    }
}
```

```
public abstract class Animal {
    protected int age;
    public void eat() {
        System.out.println("Animal is eating");
    }
    public abstract String getName();
}

public class Swan extends Animal {
    public String getName() {
        return "Swan";
    }
}
```

```java
public abstract class Animal {
    protected int age;
    public void eat() {
        System.out.println("Animal is eating");
    }
    public abstract String getName();
}

public class Swan extends Animal {
    public int getAge() {
        return age;
    }
}
```

```
public abstract class Animal {
    protected int age;
    public void eat() {
        System.out.println("Animal is eating");
    }
    public abstract String getName();
}

public class Swan extends Animal {
    public int getAge() {
        return age;
    }
}
```

```
public abstract class Animal {
    protected int age;
    public void eat() {
        System.out.println("Animal is eating");
    }
    public abstract String getName();
}

public abstract class Swan extends Animal {
    public int getAge() {
        return age;
    }
}
```

Interfaces

- Multiple inheritance

```
public interface HasTail {
    public abstract int getTailLength();
}

public class Mouse implements HasTail {
    public int tailLength = 5;
    public Mouse(int tailLength) {
        this.tailLength = tailLength
    }
    public int getTailLength() {
        return tailLength;
    }
}
```

```
public interface HasTail {
    public abstract int getTailLength();
}

public class Mouse implements HasTail {
    public int tailLength = 5;
    public Mouse(int tailLength) {
        this.tailLength = tailLength
    }
    public int getTailLength() {
        return tailLength;
    }
}
```

```
public interface HasTail {
    public abstract int getTailLength();
}

public class Mouse implements HasTail {
    public int tailLength = 5;
    public Mouse(int tailLength) {
        this.tailLength = tailLength
    }
    public int getTailLength() {
        return tailLength;
    }
}
```

```
public interface HasTail {
    public abstract int getTailLength();
}

public class Mouse implements HasTail {
    public int tailLength = 5;
    public Mouse(int tailLength) {
        this.tailLength = tailLength
    }
    public int getTailLength() {
        return tailLength;
    }
}
```

```
public interface HasTail {
        public abstract int getTailLength();
}
public interface HasWhiskers {
        public int getNumberOfWhiskers();
}
public class Mouse implements HasTail, HasWhiskers {
        public int tailLength = 5;
        public Mouse(int tailLength) {
                this.tailLength = tailLength
        }
        public int getTailLength() {
                return tailLength;
        }
}
```

```
public interface HasTail {
        public abstract int getTailLength();
}
public interface HasWhiskers {
        public int getNumberOfWhiskers();
}
public class Mouse implements HasTail, HasWhiskers {
        public int tailLength = 5;
        public Mouse(int tailLength) {
                this.tailLength = tailLength
        }
        public int getTailLength() {
                return tailLength;
        }
}
```

```
public interface HasTail {
        public abstract int getTailLength();
}
public interface HasWhiskers {
        public int getNumberOfWhiskers();
}
public class Mouse implements HasTail, HasWhiskers {
        public int tailLength = 5;
        public Mouse(int tailLength) {
                this.tailLength = tailLength
        }
        public int getTailLength() {
                return tailLength;
        }
}
```

```
public interface HasTail {
    public abstract int getTailLength();
}
public interface HasWhiskers {
    public int getNumberOfWhiskers();
}
public class Mouse implements HasTail, HasWhiskers {
    public int tailLength = 5;
    public Mouse(int tailLength) {
        this.tailLength = tailLength
    }
    public int getTailLength() {
        return tailLength;
    }
}
```

```
public interface HasTail {
        public abstract int getTailLength();
}
public interface HasWhiskers {
        public int getNumberOfWhiskers();
}
public abstract class Mouse implements HasTail, HasWhiskers {
        public int tailLength = 5;
        public Mouse(int tailLength) {
                this.tailLength = tailLength
        }
        public int getTailLength() {
                return tailLength;
        }
}
```

```
public interface HasTail {
	public int getTailLength();
}

public interface HasWhiskers {
	public int getNumberOfWhiskers();
}
```

```
public class Mouse implements HasTail, HasWhiskers {
	public int tailLength = 5;
	public int numWhiskers = 10;
	public Mouse(int tailLength) {
		this.tailLength = tailLength;
	}
	public int getTailLength() {
		return tailLength;
	}
	public int getNumberOfWhiskers() {
		return numWhiskers;
	}
}
```

```
public interface HasTail {
        public int getTailLength();
}

public interface HasWhiskers {
        public int getNumberOfWhiskers();
}
```

```
public class Mouse implements HasTail, HasWhiskers {
        public int tailLength = 5;
        public int numWhiskers = 10;
        public Mouse(int tailLength, int numWhiskers) {
                this.tailLength = tailLength;
                this.numWhiskers = numWhiskers;
        }
        public Mouse(int tailLength) {
                this.tailLength = tailLength;
        }
        public int getTailLength() {
                return tailLength;
        }
        public int getNumberOfWhiskers() {
                return numWhiskers;
        }
}
```

Methods and Encapsulation

Packages

C:\temp\packagea\ClassA.java

```
package packagea;
public class ClassA {
}
```

C:\temp\packageb\ClassB.java

```
package packageb;
import packagea.ClassA;
public class ClassB {
    ClassA a;
}
```

Packages

C:\temp\packagea\ClassA.java

```
package packagea;
public class ClassA {
}
```

C:\temp\packageb\ClassB.java

```
package packageb;
import packagea.ClassA;
public class ClassB {
	ClassA a;
}
```

Packages

C:\temp\packagea\ClassA.java

```
package packagea;
public class ClassA {
}
```

C:\temp\packageb\ClassB.java

```
package packageb;
import packagea.ClassA;
public class ClassB {
      ClassA a;
}
```

Packages

C:\temp\packagea\ClassA.java

```
package packagea;
public class ClassA {
}
```

C:\temp\packageb\ClassB.java

```
package packageb;
import packagea.ClassA;
public class ClassB {
        ClassA a;
}
```

Making Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
    public void setName(String newName) {
        name = newName;
    }
}
```

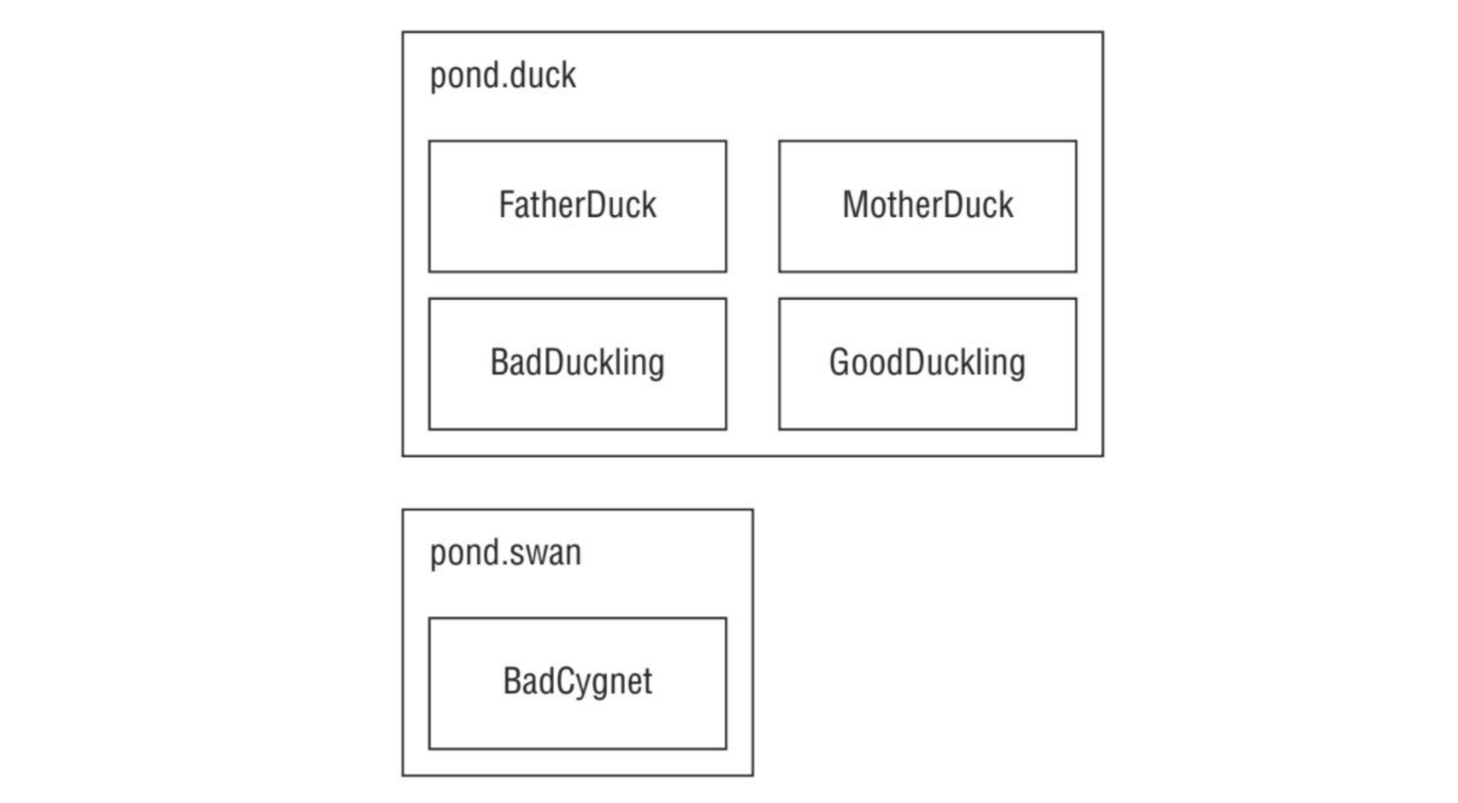
Making Methods

```
public class Animal {
    String name;
    public String getName( ) {
        return name;
    }
    public void setName(String newName) {
        name = newName;
    }
}
```

Access Modifiers

- private
- default (package private)
- protected
- public

A Duck Study


pond.duck
FatherDuck
MotherDuck
BadDuckling
GoodDuckling
pond.swan
BadCygnet

```
package pond.duck;
public class FatherDuck {
    private String noise = “quack”;
    private void quack() {
        System.out.println(noise);
    }
    private void makeNoise {
        quack();
    }
}
```

```
package pond.duck;
public class FatherDuck {
    private String noise = “quack”;
    private void quack() {
        System.out.println(noise);
    }
    private void makeNoise {
        quack();
    }
}
```

```
package pond.duck;
public class FatherDuck {
    private String noise = “quack”;
    private void quack() {
        System.out.println(noise);
    }
    private void makeNoise {
        quack();
    }
}
```

```
package pond.duck;
public class BadDuckling {
    public void makeNoise() {
        FatherDuck duck = new FatherDuck();
        duck.quack();
        System.out.println(duck.noise);
    }
}
```

```
package pond.duck;
public class BadDuckling {
    public void makeNoise() {
        FatherDuck duck = new FatherDuck();
        duck.quack();
        System.out.println(duck.noise);
    }
}
```

```
package pond.duck;
public class BadDuckling {
    public void makeNoise() {
        FatherDuck duck = new FatherDuck();
        duck.quack();
        System.out.println(duck.noise);
    }
}
```

```
package pond.duck;
public class MotherDuck {
    String noise = "quack";
    void quack() {
        System.out.println(noise);
    }
    private void makeNoise() {
        quack();
    }
}
```

```
package pond.duck;
public class MotherDuck {
    String noise = "quack";
    void quack() {
        System.out.println(noise);
    }
    private void makeNoise() {
        quack();
    }
}
```

```
package pond.duck;
public class MotherDuck {
    String noise = "quack";
    void quack() {
        System.out.println(noise);
    }
    private void makeNoise() {
        quack();
    }
}
```

```
package pond.duck;
public class GoodDuckling {
    public void makeNoise() {
        MotherDuck duck = new MotherDuck();
        duck.quack();
        System.out.println(duck.noise);
    }
}
```

```
package pond.duck;
public class GoodDuckling {
    public void makeNoise() {
        MotherDuck duck = new MotherDuck();
        duck.quack();
        System.out.println(duck.noise);
    }
}
```

```
package pond.swan;
import pond.duck.MotherDuck;
public class BadCygnet {
    public void makeNoise() {
        MotherDuck duck = new MotherDuck();
        duck.quack();
        System.out.println(duck.noise);
    }
}
```

```
package pond.swan;
import pond.duck.MotherDuck;
public class BadCygnet {
    public void makeNoise() {
        MotherDuck duck = new MotherDuck();
        duck.quack();
        System.out.println(duck.noise);
    }
}
```

Protected Access

```java
package pond.shore;
public class Bird {
    protected String text = "floating";
    protected void floatInWater() {
        System.out.println(text);
    }
}
```

```
package pond.shore;
public class Bird {
    protected String text = "floating";
    protected void floatInWater() {
        System.out.println(text);
    }
}
```

```
package pond.shore;
public class Bird {
    protected String text = "floating";
    protected void floatInWater() {
        System.out.println(text);
    }
}
```

```
package pond.goose;
import pond.shore.Bird;
public class Gosling extends Bird {
    public void swim() {
        floatInWater();
        System.out.println(text);
    }
}
```

```
package pond.goose;
import pond.shore.Bird;
public class Gosling extends Bird {
    public void swim() {
        floatInWater();
        System.out.println(text);
    }
}
```

```
package pond.goose;
import pond.shore.Bird;
public class Gosling extends Bird {
    public void swim() {
        floatInWater();
        System.out.println(text);
    }
}
```

```
package pond.goose;
import pond.shore.Bird;
public class Gosling extends Bird {
    public void swim() {
        floatInWater();
        System.out.println(text);
    }
}
```

```
package pond.shore;
public class BirdWatcher {
    public void watchBird() {
        Bird bird = new Bird();
        bird.floatInWater();
        System.out.println(bird.text);
    }
}
```

```
package pond.shore;
public class BirdWatcher {
    public void watchBird() {
        Bird bird = new Bird();
        bird.floatInWater();
        System.out.println(bird.text);
    }
}
```

```java
package pond.shore;
public class BirdWatcher {
    public void watchBird() {
        Bird bird = new Bird();
        bird.floatInWater();
        System.out.println(bird.text);
    }
}
```

```
package pond.inland;
import pond.shore.Bird;
public class BirdWatcherFromAfar {
    public void watchBird() {
        Bird bird = new Bird();
        bird.floatInWater();
        System.out.println(bird.text);
    }
}
```

```
package pond.inland;
import pond.shore.Bird;
public class BirdWatcherFromAfar {
    public void watchBird() {
        Bird bird = new Bird();
        bird.floatInWater();
        System.out.println(bird.text);
    }
}
```

```java
package pond.inland;
import pond.shore.Bird;
public class BirdWatcherFromAfar {
    public void watchBird() {
        Bird bird = new Bird();
        bird.floatInWater();
        System.out.println(bird.text);
    }
}
```

```
package pond.inland;
import pond.shore.Bird;
public class BirdWatcherFromAfar {
    public void watchBird() {
        Bird bird = new Bird();
        bird.floatInWater();
        System.out.println(bird.text);
    }
}
```

```
public class Animal {
    private int age;
    public int getAge() {
        return age;
    }
}
```

```
public class Tiger extends Animal {
    public void roar() {
        System.out.println(“roar”);
    }

    public void age() {
        System.out.println(age);
    }
}
```

```
public class Animal {
    private int age;
    public int getAge() {
        return age;
    }
}
```

```
public class Tiger extends Animal {
    public void roar() {
        System.out.println(“roar”);
    }

    public void age() {
        System.out.println(age);
    }
}
```

WHY?

99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.

127 little bugs in the code...

Thanks for Watching!

Contact: billzhang@college.harvard.edu