#Overview

This is an overview of the [__Homework 1__](https://github.com/sssurvey/ObjectOrientedDesign/tree/master/Homework_1).

*_Haomin Shi_* @ Sep 9th, 2018

```bash
haominshi@MacBook-Pro-de-Haomin:~/GitRepo/ObjectOrientedDesign$ tree Homework_1/
Homework_1/
├── Homework_1.iml
├── buildScripts
│   ├── build.xml
│   └── personal_testing
│       └── com
│           └── haomins
│               └── main
│                   ├── Main.class
│                   └── things
│                       └── Thing.class
├── lib
│   ├── apiguardian-api-1.0.0.jar
│   ├── junit-jupiter-api-5.0.0.jar
│   └── junit-platform-commons-1.0.0.jar
└── src
    └── com
        └── haomins
            ├── main
            │   └── things
            │       ├── Ant.java
            │       ├── Creature.java
            │       ├── Thing.java
            │       ├── Tiger.java
            │       └── flyer
            │           ├── Bat.java
            │           ├── Fly.java
            │           └── Flyer.java
            └── test
                ├── TestCreature.java
                └── unitTests
                    ├── AntTest.java
                    ├── BatTest.java
                    ├── CreatureTest.java
                    ├── FlyTest.java
                    ├── ThingTest.java
                    └── TigerTest.java

15 directories, 21 files
```

Stats

- Total Lines of Code:

Total:	407	310	0.7616707616707616
	Total Lines	Source Code Lines	Source Code Line %

- Lines of Code in UnitTest:

AntTest.java	**21**	**15**	**0.7142857142857143**
BatTest.java	**41**	**34**	**0.8292682926829268**
CreatureTest.java	**37**	**28**	**0.7567567567567568**
FlyTest.java	**32**	**26**	**0.8125**
ThingTest.java	**25**	**18**	**0.72**
TigerTest.java	**26**	**19**	**0.7307692307692307**
Filenames	Lines	Code Lines	Code Lines %

- Unit Test Coverage:

com.haomins.test.unitTests	100% (9/9)	86% (19/22)	93% (59/63)
Folder	Class Coverage	Method Coverage	Lines Coverage

- Complexity report:

com.haomins.main.things.Creature.Stomach	1.0	2.0
com.haomins.test.unitTests.CreatureForTesting	1.0	2.0
com.haomins.test.unitTests.AntTest	1.0	2.0

com.haomins.test.unitTests.ThingTest	1.0	2.0
com.haomins.main.things.Ant	1.0	2.0
com.haomins.main.things.Tiger	1.0	2.0
com.haomins.test.unitTests.CreatureTest	1.0	2.0
com.haomins.main.things.Thing	1.5	3.0
com.haomins.test.unitTests.TigerTest	1.0	3.0
com.haomins.test.unitTests.FlyTest	1.0	4.0
com.haomins.main.things.Creature	1.3333333333333333	4.0
com.haomins.test.unitTests.BatTest	1.0	5.0
com.haomins.main.things.flyer.Fly	1.25	5.0
com.haomins.main.things.flyer.Bat	1.5	6.0
com.haomins.test.TestCreature	5.333333333333333	16.0
Total		60.0
Average	1.4285714285714286	4.0

With the most complex class as TestCreature.java, which is understandable, consider the work TestCreature.java is doing.

Challenges

Identify the difference between *_Thing.java_* and *_Creature.java_*. I think this part is the most difficult since I cannot find a desirable way to identify the parent class of the parent class. Since we can have a grand-grand-parent class under some circumstances.

__Current Soltuion:__

```java
// To identify Thing.java
if (thing instanceof Thing) return "is Thing";

// To identify Creature.java
if (thing.getClass().getSuperclass().equals(Creature.class)) return "is Creature"
```