As of the completion of this assignment, I found the most difficult part of the assignment to be properly installing a plugin to use for analysis of the cyclomatic complexity of my code. I was unable to do this in the time provided, so I opted to use an online cyclomatic complexity analyzer called Lizard. This can be found at the following link: http://www.lizard.ws/. Other than this, the assignment was completed without issue, and there are no missing parts.

The following are some important statistics about my code:

Ant.java:
Lines of code: 11
Cyclomatic Complexity of constructor: 1
Cyclomatic Complexity of move(): 1

Bat.java:
Lines of code: 29
Cyclomatic Complexity of constructor: 1
Cyclomatic Complexity of eat(): 3
Cyclomatic Complexity of fly(): 1
Cyclomatic Complexity of move(): 1

Creature.java:
Lines of code: 25
Cyclomatic Complexity of constructor: 1
Cyclomatic Complexity of eat(): 1
Cyclomatic Complexity of whatDidYouEat(): 2

Fly.java:
Lines of code: 26
Cyclomatic Complexity of constructor: 1
Cyclomatic Complexity of eat(): 2
Cyclomatic Complexity of fly(): 1
Cyclomatic Complexity of move(): 1

Flyer.java:
Lines of code: 4

TestCreature.java:
Lines of code: 40
Cyclomatic Complexity of main(): 3

Thing.java:
Lines of code: 20
Cyclomatic Complexity of constructor: 1
Cyclomatic Complexity of toString(): 2

Tiger.java:
Lines of code: 11

Cyclomatic Complexity of constructor: 1
Cyclomatic Complexity of move(): 1

AntTest.java:
Lines of code: 9
Cyclomatic Complexity of testName(): 1

BatTest.java:
Lines of code: 9
Cyclomatic Complexity of testName(): 1

FlyTest.java:
Lines of code: 9
Cyclomatic Complexity of testName(): 1

ThingTest.java:
Lines of code: 9
Cyclomatic Complexity of testName(): 1

TigerTest.java:
Lines of code: 9
Cyclomatic Complexity of testName(): 1

Unit Test Coverage using EclEmma Java Code Coverage 3.1.2 Eclipse Plugin: