```
/**
 * A class which spawns, breeds, kills, and returns information on roaches.
 *
 * @author Phil Fevry
 * @version 1.0
 */
public class RoachPopulation
{
    // instance variable
    private int roachPopulation;
    /**
     * Constructs a population of roaches (Yuck!)
     *
     * @param   initialRoachPopulation    the amount of roaches to create
     */
    public RoachPopulation(int initialRoachPopulation)
    {
        // initialise instance variable
        roachPopulation = initialRoachPopulation;
    }
    /**
     * Doubles the roach population
     *
     */
    public void breed()
    {
        roachPopulation *= 2;
    }
    /**
     * Sprays pesticide which kills a certain amount of roaches
     *
     * @param   percentage       the percentage of roaches to kill with the spra
y
     */
    public void spray(double percentage)
    {
        roachPopulation = roachPopulation - (int)(roachPopulation * percentage)/
100;
    }
    /**
     * Gets the current amount of living roaches
     *
     * @return  the current roach population
     */
    public int getRoaches()
    {
        return roachPopulation;
    }
    /**
     * Returns a string which specifies the number of the roach population
     *
     * @return  string   "Number of roaches in the population: <number of roache
s>"
     */
    public String toString()
    {
        return "Number of roaches in the population: " + roachPopulation;
    }
}
```

Blank line

80/80

```
import static org.junit.Assert.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

/**
 * The test class RoachPopulationTest for the problem
 * from the text book P3.10 Chapter 3
 * Java Concepts – Early Objects by
 * Cay Horstmann
 *
 * @author  CS 140 Instructors
 * @version 2/4/2017
 */
public class RoachPopulationTest
{

    RoachPopulation population;  // instance variable

    /**
     * Sets up the test fixture.
     *
     * Called before every test case method.
     */

    @Before
    public void setUp()
    {
        population = new RoachPopulation(200);
    }



        @Test
        public void test_getRoaches() {
            assertEquals(200, population.getRoaches() );
        }



    @Test
    public void test_breed() {
        int currentRoaches = population.getRoaches();
        population.breed();
        assertEquals(currentRoaches * 2, population.getRoaches() );
    }



    @Test
    public void test_spray() {
        int currentRoaches = population.getRoaches();
        population.spray(20);     // reduce the population by 20%
        assertEquals((int)(currentRoaches * 0.80), population.getRoaches() );
    }



    @Test
    public void test_breedAndspray() {
        int currentRoaches = population.getRoaches();
        population.breed();     // double the population
        assertEquals(currentRoaches * 2, population.getRoaches() );
```

```
        currentRoaches = population.getRoaches();
        population.spray(40);  // reduce the population by 40%
        assertEquals((int)(currentRoaches * 0.60), population.getRoaches() );

        currentRoaches = population.getRoaches();
        population.breed();     // double the population
        assertEquals(currentRoaches * 2, population.getRoaches() );

        currentRoaches = population.getRoaches();
        population.spray(30);  // reduce the population by 30%
        assertEquals((int)(currentRoaches * 0.70), population.getRoaches() );

        currentRoaches = population.getRoaches();
        population.breed();     // double the population
        assertEquals(currentRoaches * 2, population.getRoaches() );

        currentRoaches = population.getRoaches();
        population.spray(50);  // reduce the population by 50%
        assertEquals((int)(currentRoaches * 0.50), population.getRoaches() );

        currentRoaches = population.getRoaches();
        assertEquals(336, population.getRoaches());
    }



    @Test
    public void test_toString() {
        assertEquals("Number of roaches in the population: " + population.getRoa
ches(), population.toString());
    }

}
```

Discussion Log
Assignment: Project 2
Name: Phil Fevry
Date: 2/13/17

I finished this project myself without any help without any additional resources
.

What I learned from this lab:

- About Programming: I didnâM−^@M−^Yt learn anything new from this lab.
- About BlueJ: I learned about BlueJ’s testing feature and the utility of JU
nit.

Time taken to complete project:

- I finished in about 30 minutes.

Difficulties faced:

- I was confused as why the tester was failing my breed method but realized
it was because:
(a) I didn’t cast to the double to int
−and−
(b) At first, I didn’t convert the double into a percantage by dividing by
100.

I had to rename the discussion log

```
commit 0054ab413b7c859ff97ad1f1117a3aa41d70cf55
Author: Phil Fevry <pfevry@worcester.edu>
Date:   Mon Feb 13 21:45:03 2017 −0500

    Project 2 Final Version

commit e0b6868ff5226763b36b9ceaf1e13045f3bebb0d
Author: Aparna Mahadev <amahadev@worcester.edu>
Date:   Sun Feb 5 13:24:06 2017 −0500

    Project2
```