

Introduction to Classes and Instances

Purpose: To gain practice with objects and classes by answering exercises in the book and using instances and methods.

Learning Objectives:

- Gain understanding of Chapter 1;
- Building a class from scratch;
- Getting familiarity with BlueJ.

Instructions:

1. Complete all the remaining Chapter 1 exercises (Starting on page 12, exercise 1.13 through exercise 1.36 on page 17) including challenge exercises. Answers to all questions should be included in a word processing document. Yes, you need to do a lab report. Yes, you need to read the parts of the book in between the exercises. Do not race through the problems!!!! Take your time, redo them as many times as it takes to feel comfortable.
2. Save a copy of the **house** project into another location. Call it house2. Modify it by adding a class **Rect**. **NOTE: Call your class Rect and not Rectangle, because the web files already contain something called Rectangle. Don't be confused by this, because the Rectangle class is a part of Java and is used in drawing methods.**
(<http://docs.oracle.com/javase/7/docs/api/java/awt/Rectangle.html>)

Hint: This class will be (very) similar to the class **Square**. Study the code for Square and base your class **Rect** on it.

In order to create this new class:

- click New Class on the left hand menu and name the new class **Rect**
- copy the code from **Square** into it, including the necessary changes to the name of the class and the code to represent a rectangle rather than a square
- test all its methods thoroughly to make sure it works the way it is supposed to



Figure 1: Here is my house with a rectangular door (for giants!)

3. Modify the **house** project so that the picture object also has a door (See Figure 1). Test ALL the **Picture** class methods to make sure they work as expected.

4. Create a class called Donut that represents an edible confectionary with a hole in the middle. The donut should be a movable one, i.e. the functionality of the class should include:

```
public void makeVisible()

public void makeInvisible()

public void moveRight() // Move the Donut a few pixels to the right.

public void moveLeft() // Move the Donut a few pixels to the left.

public void moveUp() // Move the Donut a few pixels up.

public void moveDown() // Move the Donut a few pixels down.

public void moveHorizontal(int distance) // Move the Donut horizontally by 'distance' pixels.

public void moveVertical(int distance) // Move the Donut vertically by 'distance' pixels.

public void slowMoveHorizontal(int distance) // Slowly move the Donut horizontally by 'distance' pixels.

public void slowMoveVertical(int distance) // Slowly move the Donut vertically by 'distance' pixels.

public void slowMoveAround(int size) // Slowly move Donut around clockwise in a square of 'size' pixels.

/**
 * Change the color. Valid colors are "red", "yellow", "blue", "green", "magenta" and "black".
 */
public void changeColor(String newColor)
```

Hint: Do this manually first in BlueJ. Figure out your algorithm before you make your Donut class. Your donut and your donut hole do not need to move at the same time.