

# CIS 162 Project 1

## Business Card

### Section 04 (Kurmas)

#### Due Date

- at the start of lab on Monday, 17 September (be prepared to demo in lab)

#### Before Starting the Project

- Read zyBook chapter 1 and 3
- Know how to compile and run a Java application in BlueJ
- Read the entire project description before starting
- Track the amount of time you spend on the project
- Watch the Project 1 video (<https://www.youtube.com/watch?v=JDXdr2aSJ9U>)

#### Learning Objectives

After completing this project, you should be able to:

- *write, compile and run* a simple Java application
- *declare* variables to store integers and Strings
- *assign* values to variables
- *write* code to drop simple shapes

#### Drawing Shapes

Unfortunately, our textbook does not describe drawing well. Use this document to learn everything you need to know to draw a business card.

#### X-Y Coordinate System

Computers often use a coordinate system a bit different that you may be used to. The x-axis increases from left to right like usual but the y-axis increases from top to bottom. Therefore, the origin (0, 0) is in the upper left corner.



## Draw a Rectangle

The location and size of a rectangle is determined by four numbers: 1) the x location of the upper left corner, 2) the y location of the upper left corner, 3) the width 4) and the height.

```
g.drawRect(x, y, width, height);
```

For example, the following statement will draw a 200 x 100 rectangle at location (40, 60).

```
g.drawRect(40, 60, 200, 100);
```



## Starting Template

1. Read the comments within the code to learn how to draw rectangles, ovals, lines and text.
2. Copy and paste this template into a BlueJ project.
3. Compile and run the code below before making changes.

```
import javax.swing.*;
import java.awt.*;

public class Drawing extends JPanel{

    public static void main(String[] a) {
        JFrame f = new JFrame();
        f.setContentPane(new Drawing());
        f.setSize(600, 400);
        f.setVisible(true);
    }

    public void paintComponent(Graphics g){
        // this statement required
        super.paintComponent(g);

        // Get the current width and height of the drawing.
        int drawing_width = getWidth();
        int drawing_height = getHeight();

        // optional: paint the background color (default is white)
        setBackground(Color.CYAN);

        // display words
        g.setColor(Color.black);
        g.drawString("Picture Title", 130, 20);

        // draw a solid and empty rectangle
        g.setColor(Color.RED);
        g.fillRect(100, 100, 70, 50);
        g.drawRect(100, 170, 70, 50);

        // draw a solid and empty oval
        g.setColor(Color.GREEN);
        g.fillOval(50, 10, 70, 50);
        g.drawOval(20, 50, 70, 50);

        // draw lines
        g.setColor(Color.BLUE);
        g.drawLine(190, 50, 190, 150);
        g.drawLine(210, 50, 210, 150);
        g.drawLine(230, 50, 230, 150);

        // put a box with a corner in the middle
        int box_width = 100;
        g.setColor(Color.ORANGE);
        g.fillRect(drawing_width / 2, drawing_height / 2,
            box_width, box_width);
    }
}
```

## Project 1: Design Your Own Business Card

Create an application that displays a fictional business card. All lines of code that you write will be within the `paintComponent` method.

- Start with the template provided above.
- Draw a rectangle to form the border (5 pts).
- Provide your name, a fictional business, clever tagline/motto and any other fictional contact information you choose such as phone, fax and e-mail (15 pts).
- Use at least five drawing elements such as lines, rectangles and ovals to create a logo (15 pts).
- Use at least two colors but not too many (10 pts).
- Use multiple fonts sizes and styles. See additional information below (10 pts).
- The card layout should be attractive and well balanced (10 pts).

### Program Comments (10 pts)

Elegant source code that follows the GVSU [Java Style Guide](#).

### Challenge Requirements (25 pts)

The following should only be attempted after all of the above requirements have been completed. Challenge requirements demonstrate that you have the initiative to investigate problems and identify solutions with minimal help from your instructor.

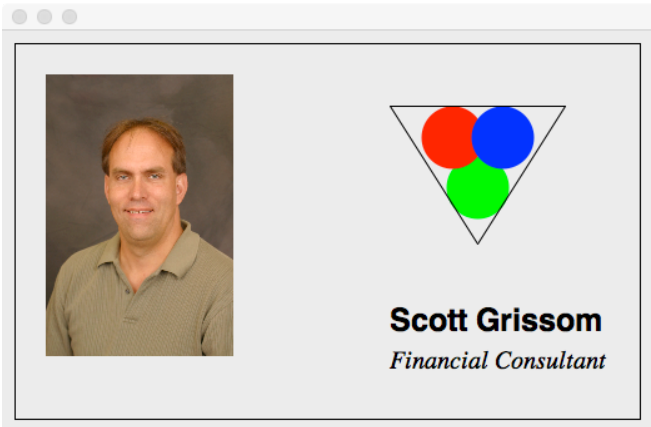
- Add a photo of yourself. See additional information below. (10 pts)
- Be able to move 1) the logo and 2) the image independently of each other by changing one or two numbers in the code. This requires proper use of variables. See additional information below (15 pts).

### Extra Credit (up to 10 pts)

Design your card so that you can re-size it. To do this, you will replace your hard-coded locations (e.g., 100, 100) with values that are relative to the current height and width of the card. (Notice how `drawing_width` and `drawing_height` are used in the example code above.) Your card need not be perfectly scalable. (Scaling things such as font sizes is quite difficult. I'm happy if the picture and logo scale and move well.)

## Poor Example

The following example is really quite bland. Your business card should be more attractive, have a more elaborate logo, and have a bit more text. Strive to impress your instructor with your graphic design skills!



## Using Variables as Method Parameters

Well-designed and maintainable code uses variables to store values. Variables are typically declared at the top of a method and then assigned values as needed. Your goal is to be able to change the location of your logo and other portions by changing only a few lines of code.

### Poor:

```
g.drawRect(10, 10, 50, 40);
```

### Better:

```
g.drawRect(x, y, width, height);  
g.drawRect(x, y, x2, y2);  
g.drawRect(top, left, bottom, right);
```

## Changing Font Sizes and Styles

The `setFont` method takes a single parameter of type `Font`. However, a `Font` object must be instantiated first.

```
Font myFont = new Font("serif", Font.ITALIC, 20);  
g.setFont(myFont);
```

- Generic font names include “serif”, “sanserif” and “monospaced”. More specific font names might work as well depending on the computer: “Times”, “Helvetica”, “Courier” and other names you might know.
- Font styles include `Font.PLAIN`, `Font.ITALIC`, `Font.BOLD` or you can combine `Font.ITALIC+Font.BOLD`.
- Font size, the third parameter, typically ranges from 10-48.

## Adding an Image

You must import additional packages at the top of your source file for the example to compile.

```
import java.awt.image.*;
import javax.imageio.*;
import java.io.*;
```

The following code uses some advanced features not covered in the book to display a JPEG or GIF image. This assumes you have an image in your BlueJ project directory called “MyPhoto.jpg”. Instead, replace “MyPhoto.jpg” with the actual name of your image.

```
BufferedImage photo = null;
try {
    File file = new File("MyPhoto.jpg");
    photo = ImageIO.read(file);
} catch (IOException e){
    g.drawString("Problem reading the file", 100, 100);
}
g.drawImage(photo, 10, 10, 150, 225, null);
```

## Late Policy

Projects should be turned in on time at the START of the class period. However, you are encouraged to complete a project even if you must turn it in late.

- The first day (-10 pts)
- Each subsequent weekday is an additional -5 pts
- Weekends and university holidays are free days.

## Submission

A professional document **that is printed and stapled**.

- Cover page - Your project must have a cover page that includes your name, a title, and screenshot of your business card. (-5 pts if missing)
- Signed Pledge – The cover page must include the following signed pledge: "I pledge that this work is entirely mine, and mine alone (except for any code provided by my instructor). " In addition, provide names of any people you helped or received help from. Under no circumstances do you exchange code electronically. You are responsible for understanding and adhering to the [School of CIS Guidelines for Academic Honesty](#).
- Time Card – The cover page must also include a brief statement of how much time you spent on the project. For example, “I spent 7 hours on this project from September 22-27 reading the book, designing a solution, writing code, fixing errors and putting together the printed document.”
- Source code - a printout of your elegant source code (with your name).
- Demo – be prepared to demo your project on a lab computer or your laptop. I will ask you to perform a variety of tasks including moving your logo that should be accomplished by changing only two lines of your code.