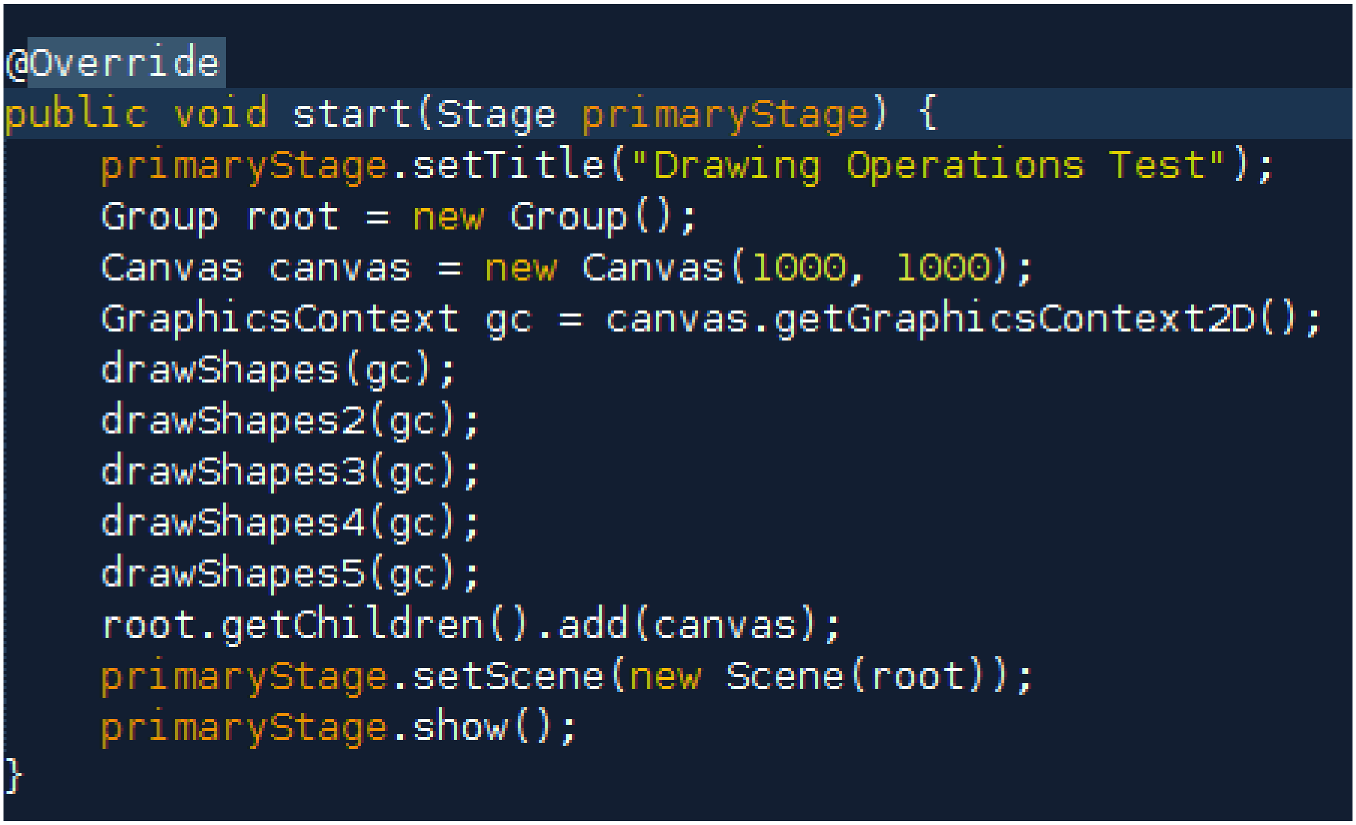
Jeremy Munch Period 4

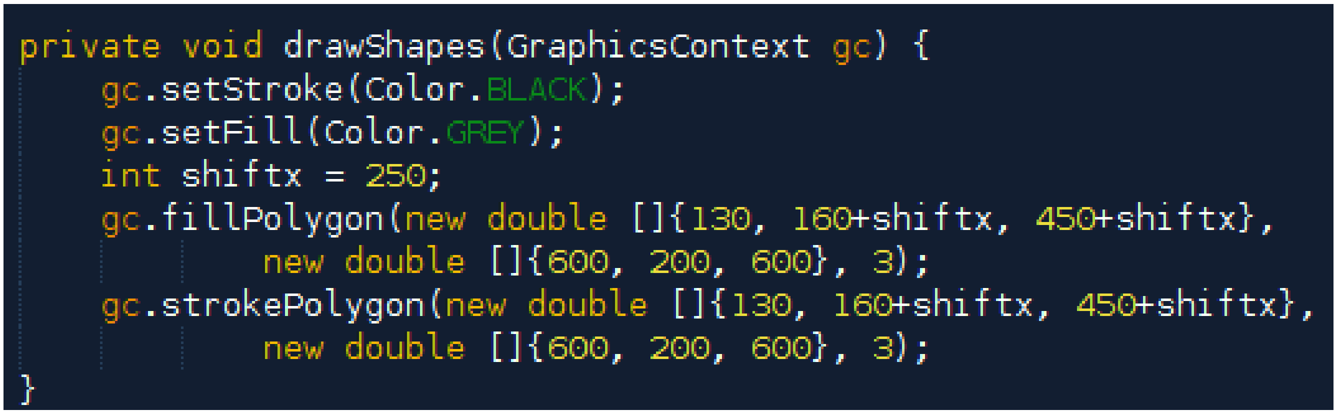
Method Madness Essay

My project, “Munch_4_MethodMadness.java”, uses several methods to create a picture. For example, my code draws a pretty little house made three bright colors, with a chimney that has a reddish brown applied to it to make it seem it was made out of bricks with one window with two lines draw inside of it to add detail. The roof of the house is gray and is shaped like a triangle like hoe kids in kindergarten used to draw houses. My drawing is probably one of the least detailed pieces of art due to the lack of understanding and the non understanding of the code. My program runs on methods known as *private void drawShapes(gc); public method is calling the several private methods   private void drawShapes(GraphicsContext gc){*.



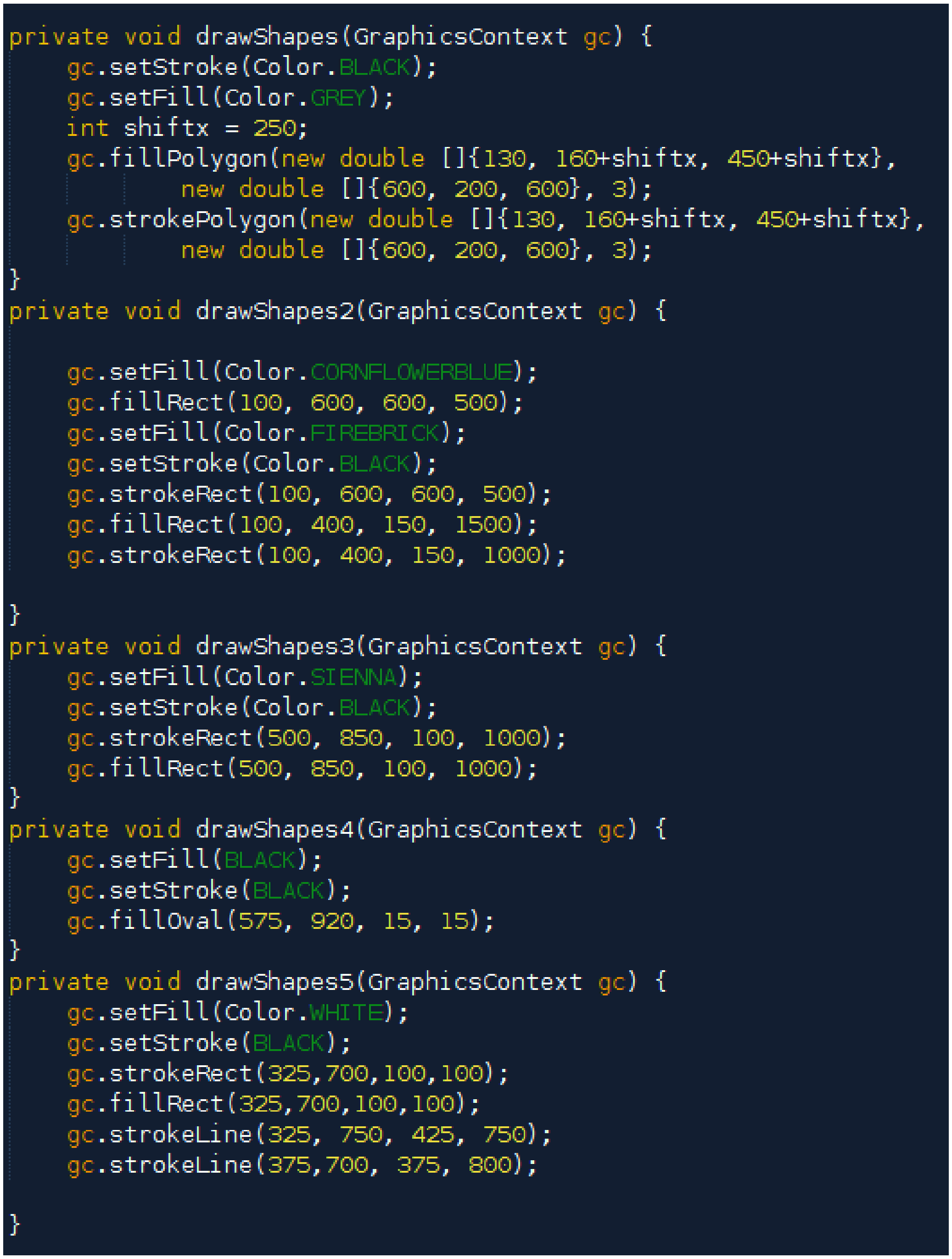
```java
@Override
public void start(Stage primaryStage) {
    primaryStage.setTitle("Drawing Operations Test");
    Group root = new Group();
    Canvas canvas = new Canvas(1000, 1000);
    GraphicsContext gc = canvas.getGraphicsContext2D();
    drawShapes(gc);
    drawShapes2(gc);
    drawShapes3(gc);
    drawShapes4(gc);
    drawShapes5(gc);
    root.getChildren().add(canvas);
    primaryStage.setScene(new Scene(root));
    primaryStage.show();
}
```

Jeremy Munch Period 4



```java
private void drawShapes(GraphicsContext gc) {
    gc.setStroke(Color.BLACK);
    gc.setFill(Color.GREY);
    int shiftx = 250;
    gc.fillPolygon(new double []{130, 160+shiftx, 450+shiftx},
            new double []{600, 200, 600}, 3);
    gc.strokePolygon(new double []{130, 160+shiftx, 450+shiftx},
            new double []{600, 200, 600}, 3);
}
```

The following methods are ones that I created to make my code art.

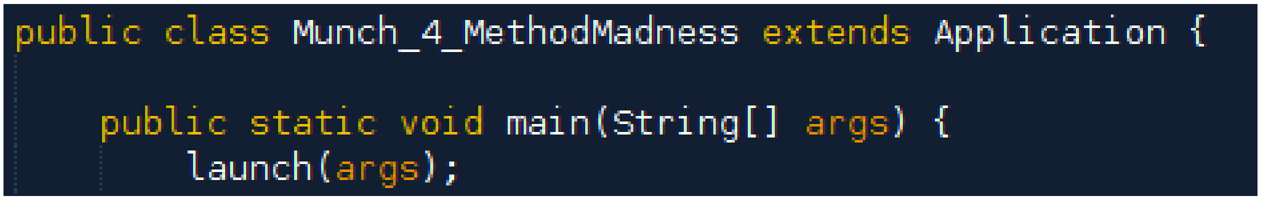


```java
private void drawShapes(GraphicsContext gc) {
    gc.setStroke(Color.BLACK);
    gc.setFill(Color.GREY);
    int shiftx = 250;
    gc.fillPolygon(new double []{130, 160+shiftx, 450+shiftx},
            new double []{600, 200, 600}, 3);
    gc.strokePolygon(new double []{130, 160+shiftx, 450+shiftx},
            new double []{600, 200, 600}, 3);
}
private void drawShapes2(GraphicsContext gc) {

    gc.setFill(Color.CORNFLOWERBLUE);
    gc.fillRect(100, 600, 600, 500);
    gc.setFill(Color.FIREBRICK);
    gc.setStroke(Color.BLACK);
    gc.strokeRect(100, 600, 600, 500);
    gc.fillRect(100, 400, 150, 1500);
    gc.strokeRect(100, 400, 150, 1000);

}
private void drawShapes3(GraphicsContext gc) {
    gc.setFill(Color.SIENNA);
    gc.setStroke(Color.BLACK);
    gc.strokeRect(500, 850, 100, 1000);
    gc.fillRect(500, 850, 100, 1000);
}
private void drawShapes4(GraphicsContext gc) {
    gc.setFill(BLACK);
    gc.setStroke(BLACK);
    gc.fillOval(575, 920, 15, 15);
}
private void drawShapes5(GraphicsContext gc) {
    gc.setFill(Color.WHITE);
    gc.setStroke(BLACK);
    gc.strokeRect(325,700,100,100);
    gc.fillRect(325,700,100,100);
    gc.strokeLine(325, 750, 425, 750);
    gc.strokeLine(375,700, 375, 800);

}
```

Jeremy Munch Period 4

The values I used for each method, like *gc.fillRect(325, 700, 100, 100);* are coordinates and the length, width, and height of the rectangle in the canvas. The template of the method would be *gc.fillRect(int x, int y, int width, int height);*, usually the method isn't just left alone like that, most likely you would find a *gc.setFill(Color);* somewhere before the actual rectangle template. The *gc.setFill(Color);* would set any templates that are after it that certain color in the canvas until a new method is created or another *setFill* template appears. The canvas is where the picture is drawn, which could be altered into any size you would like, my canvas is 1000 x 1000.

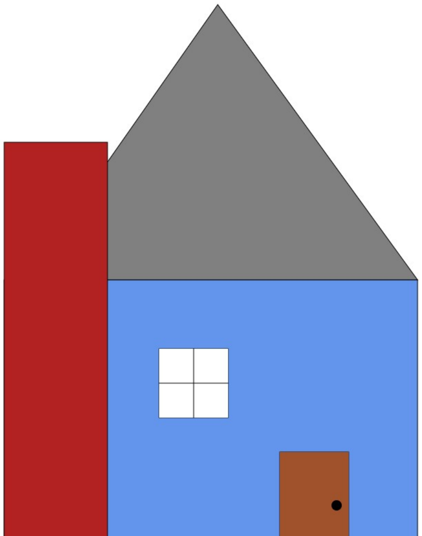
While there are many methods in my program, but only one method really matters and that is the main(): *launch(args);*.



```
public class Munch_4_MethodMadness extends Application {

    public static void main(String[] args) {
        launch(args);
```

This is why the methods are drawn on a canvas and not printed out words or numbers.

In the program, I used several access modifiers, one public and several private voids. The private method *private void drawShapes(GraphicsContext gc) {* is called in the public method *public void start(Stage primaryStage) {*. I used class constructors like *drawShapes* and *start* to label the different public and private voids. I named the voids the same as the class did and the methods I made created this picture:



Jeremy Munch Period 4

My project is probably one of the most common designs and not that creative as other classmates pieces of arts are, but I still learned several pieces of information that I will excel with throughout the year .