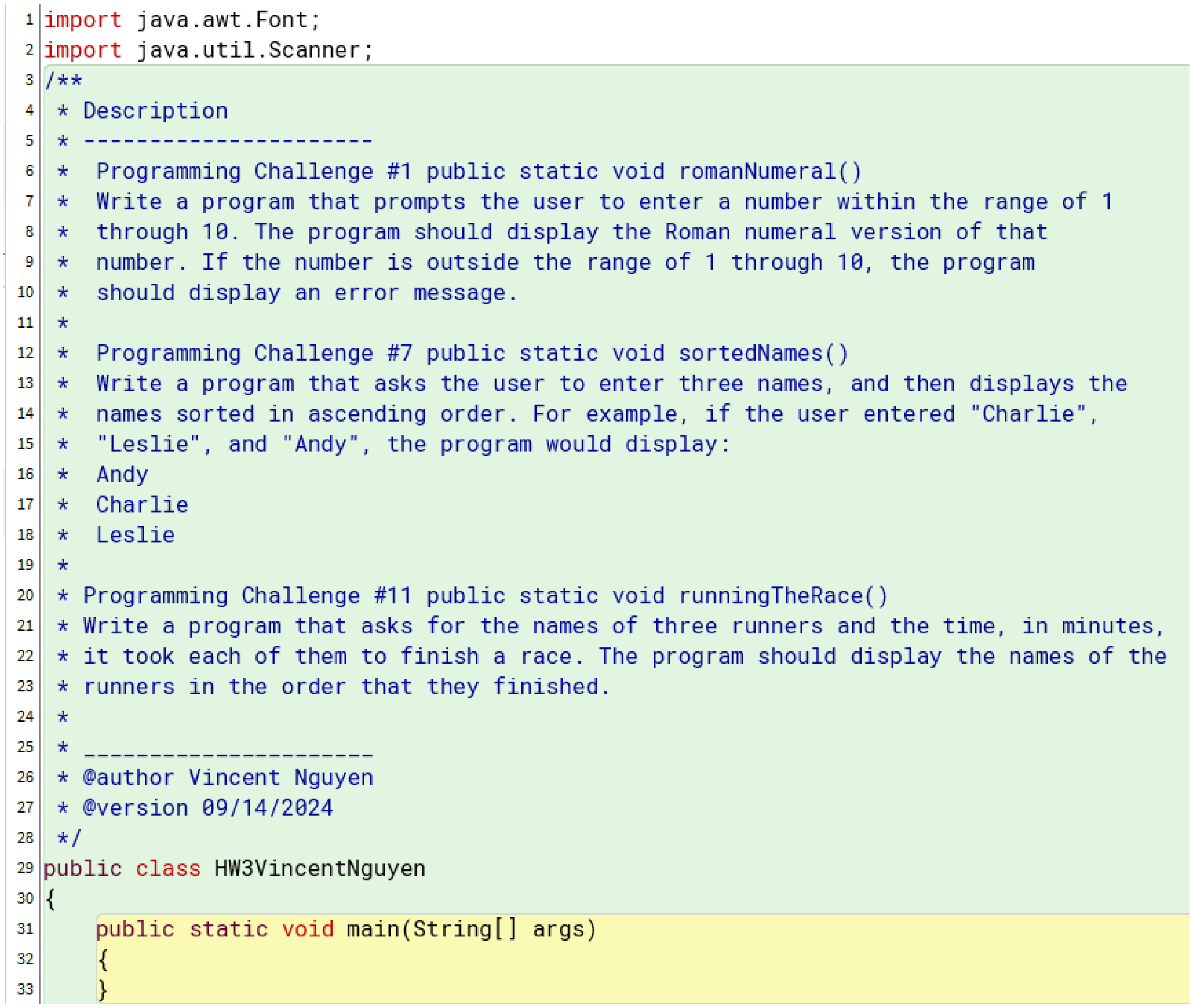
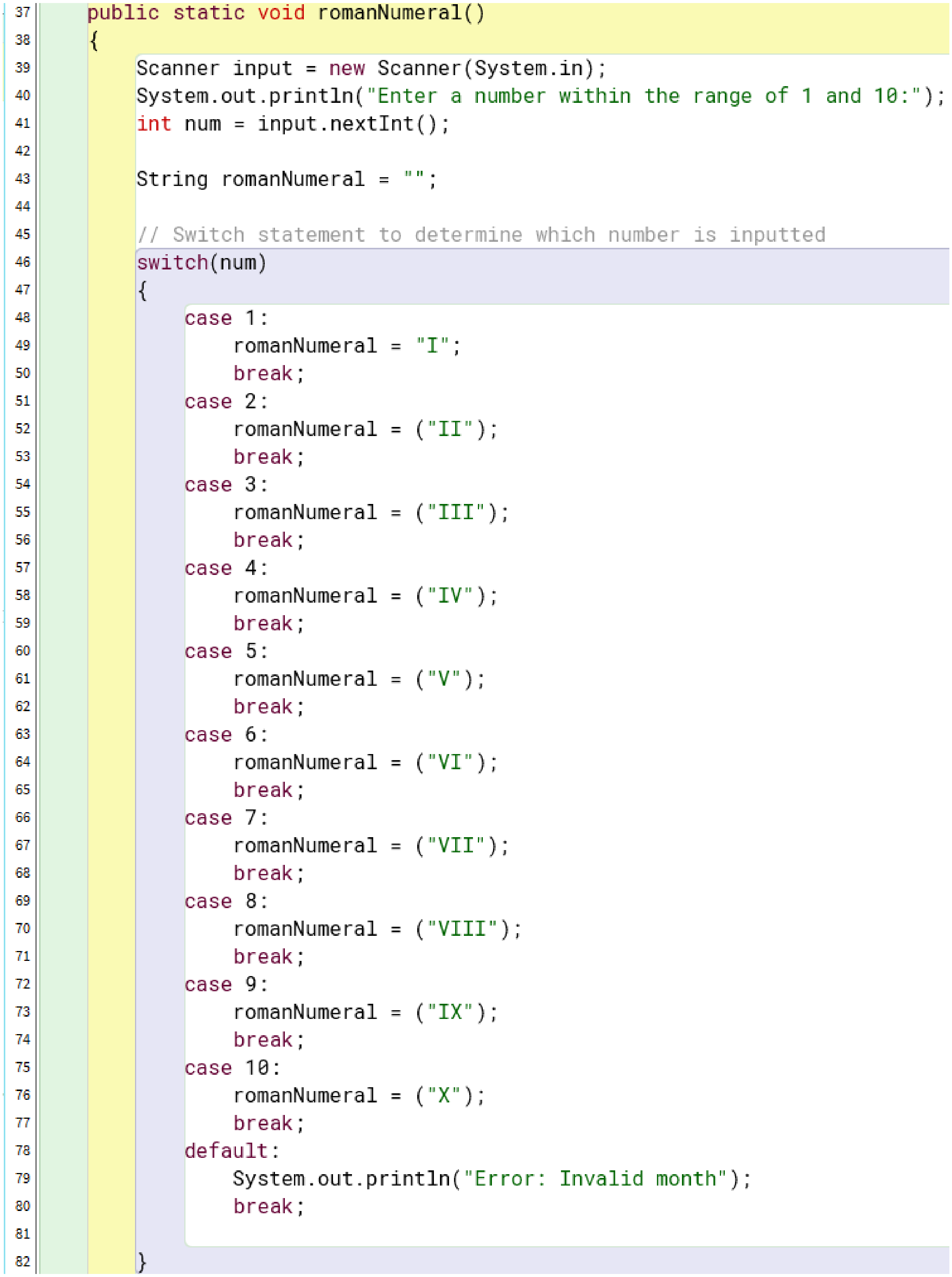
This is what I have so far for two programs

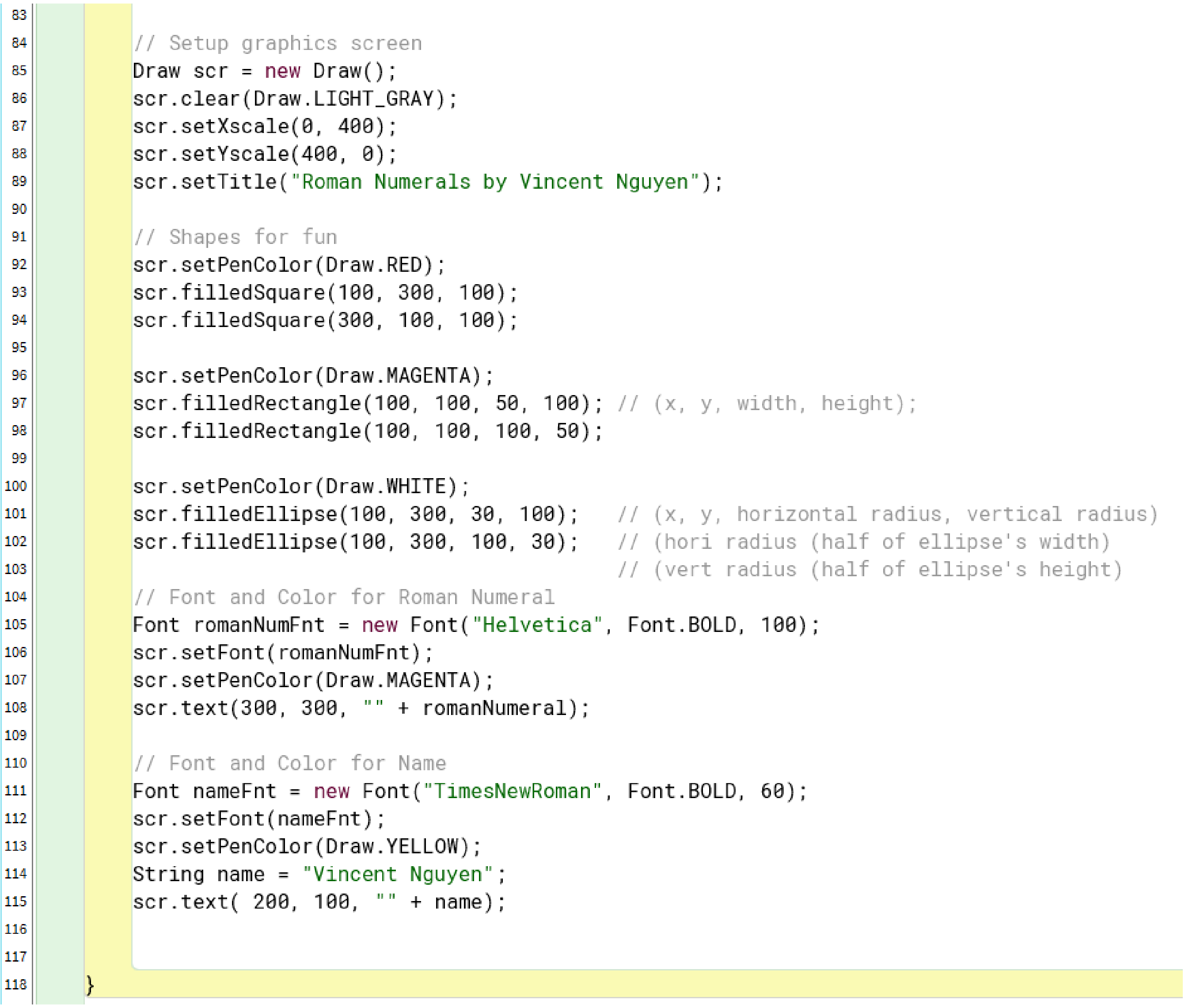


```
import java.awt.Font;
import java.util.Scanner;
/**
 * Description
 * ----------------------
 *  Programming Challenge #1 public static void romanNumeral()
 *  Write a program that prompts the user to enter a number within the range of 1
 *  through 10. The program should display the Roman numeral version of that
 *  number. If the number is outside the range of 1 through 10, the program
 *  should display an error message.
 *
 *  Programming Challenge #7 public static void sortedNames()
 *  Write a program that asks the user to enter three names, and then displays the
 *  names sorted in ascending order. For example, if the user entered "Charlie",
 *  "Leslie", and "Andy", the program would display:
 *  Andy
 *  Charlie
 *  Leslie
 *
 * Programming Challenge #11 public static void runningTheRace()
 * Write a program that asks for the names of three runners and the time, in minutes,
 * it took each of them to finish a race. The program should display the names of the
 * runners in the order that they finished.
 *
 * ______________________
 * @author Vincent Nguyen
 * @version 09/14/2024
 */
public class HW3VincentNguyen
{
    public static void main(String[] args)
    {
    }
```

I initially thought of doing if else statements but since roman numerals were going to go off of numbers I thought it would be easier and cleaner with a case switch method from the book.

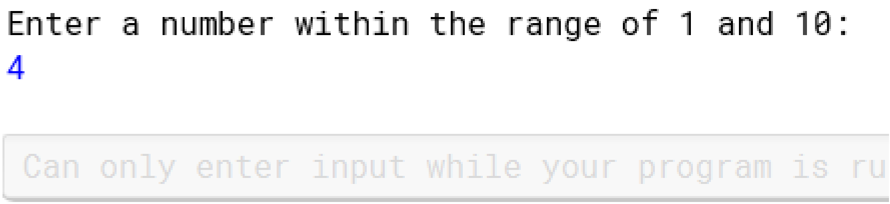


```
public static void romanNumeral()
{
    Scanner input = new Scanner(System.in);
    System.out.println("Enter a number within the range of 1 and 10:");
    int num = input.nextInt();

    String romanNumeral = "";

    // Switch statement to determine which number is inputted
    switch(num)
    {
        case 1:
            romanNumeral = "I";
            break;
        case 2:
            romanNumeral = ("II");
            break;
        case 3:
            romanNumeral = ("III");
            break;
        case 4:
            romanNumeral = ("IV");
            break;
        case 5:
            romanNumeral = ("V");
            break;
        case 6:
            romanNumeral = ("VI");
            break;
        case 7:
            romanNumeral = ("VII");
            break;
        case 8:
            romanNumeral = ("VIII");
            break;
        case 9:
            romanNumeral = ("IX");
            break;
        case 10:
            romanNumeral = ("X");
            break;
        default:
            System.out.println("Error: Invalid month");
            break;

    }
```

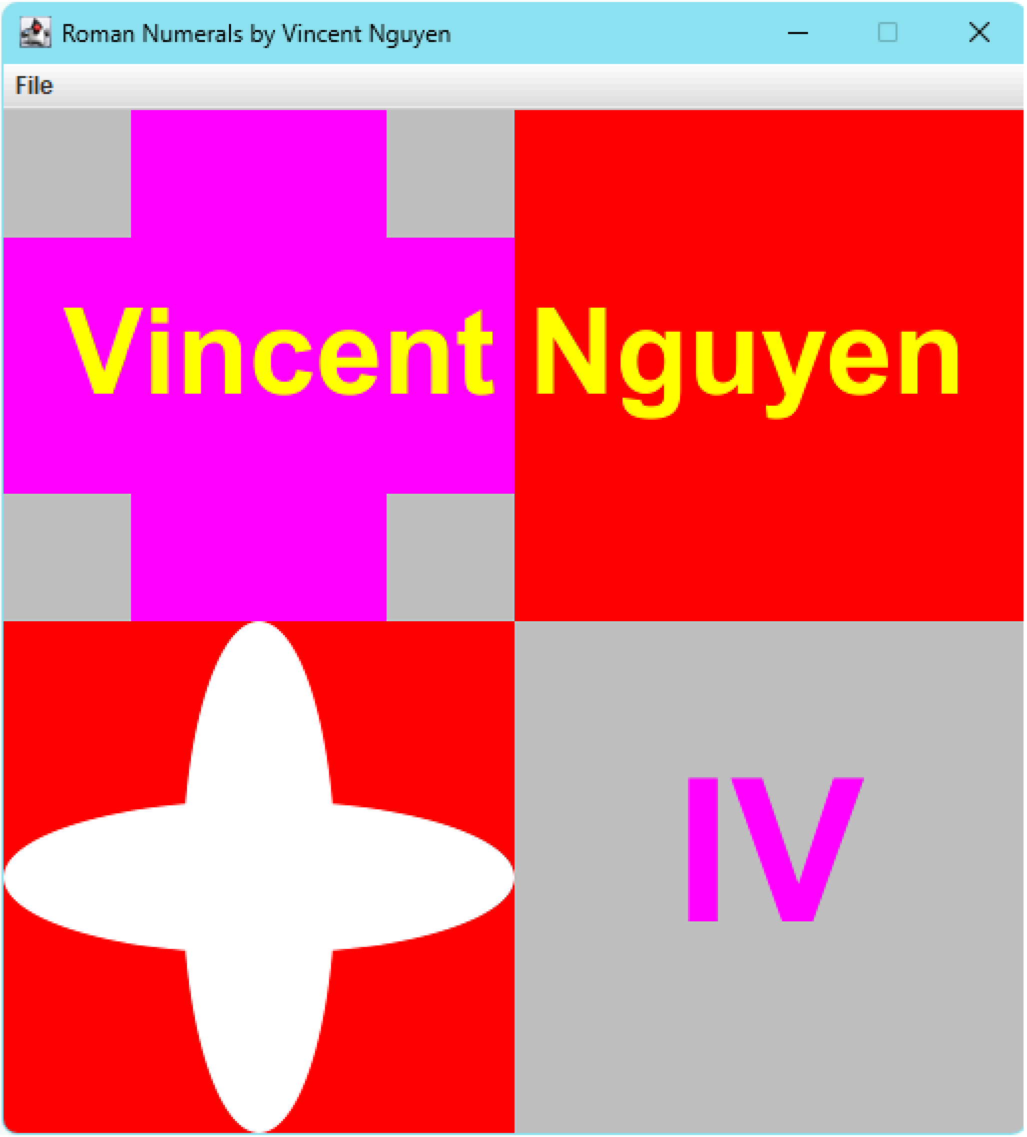


```

    // Setup graphics screen
    Draw scr = new Draw();
    scr.clear(Draw.LIGHT_GRAY);
    scr.setXscale(0, 400);
    scr.setYscale(400, 0);
    scr.setTitle("Roman Numerals by Vincent Nguyen");

    // Shapes for fun
    scr.setPenColor(Draw.RED);
    scr.filledSquare(100, 300, 100);
    scr.filledSquare(300, 100, 100);

    scr.setPenColor(Draw.MAGENTA);
    scr.filledRectangle(100, 100, 50, 100); // (x, y, width, height);
    scr.filledRectangle(100, 100, 100, 50);

    scr.setPenColor(Draw.WHITE);
    scr.filledEllipse(100, 300, 30, 100);   // (x, y, horizontal radius, vertical radius)
    scr.filledEllipse(100, 300, 100, 30);   // (hori radius (half of ellipse's width)
                                            // (vert radius (half of ellipse's height)
    // Font and Color for Roman Numeral
    Font romanNumFnt = new Font("Helvetica", Font.BOLD, 100);
    scr.setFont(romanNumFnt);
    scr.setPenColor(Draw.MAGENTA);
    scr.text(300, 300, "" + romanNumeral);

    // Font and Color for Name
    Font nameFnt = new Font("TimesNewRoman", Font.BOLD, 60);
    scr.setFont(nameFnt);
    scr.setPenColor(Draw.YELLOW);
    String name = "Vincent Nguyen";
    scr.text( 200, 100, "" + name);


}
```

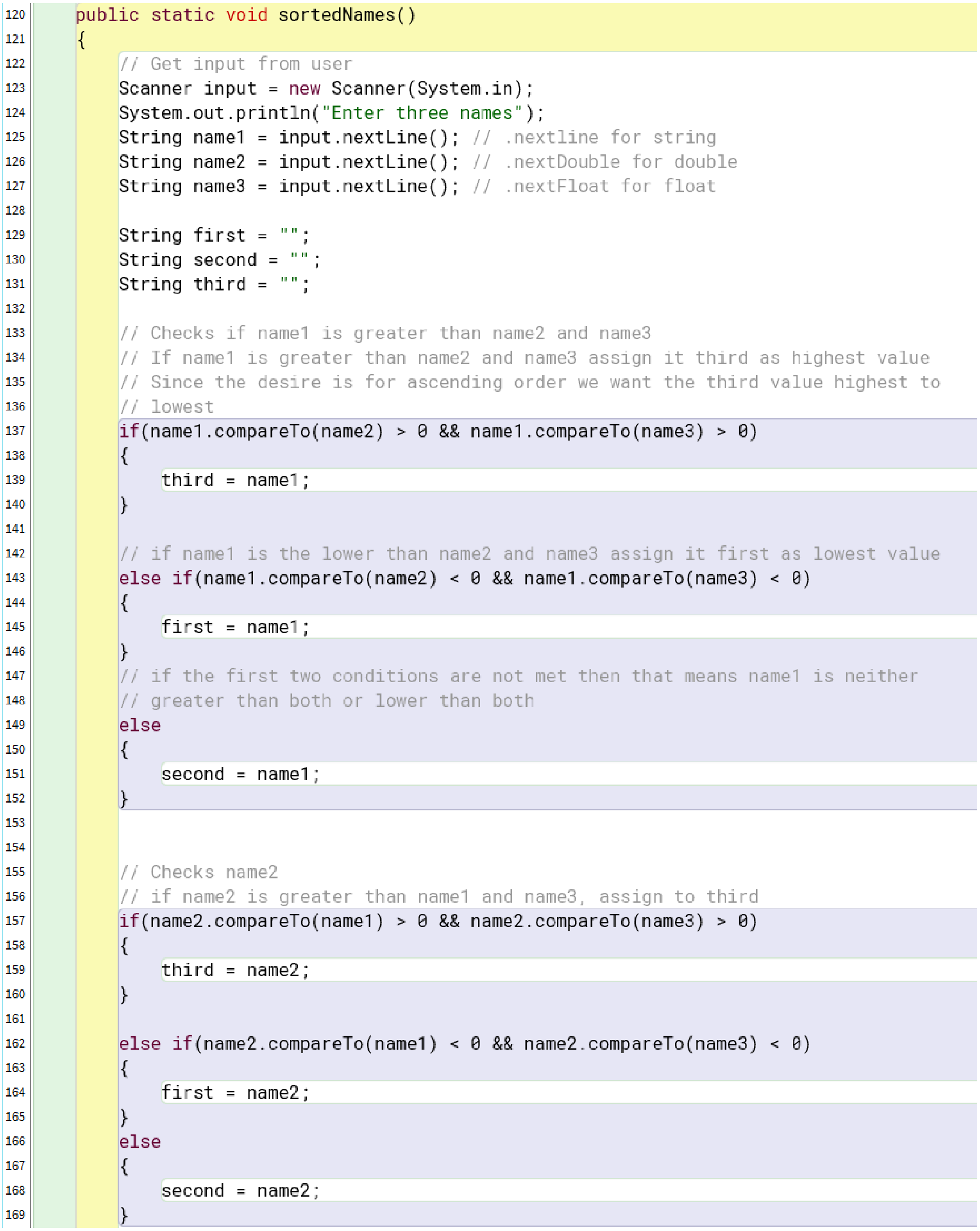
Graphics Screen for Roman numerals



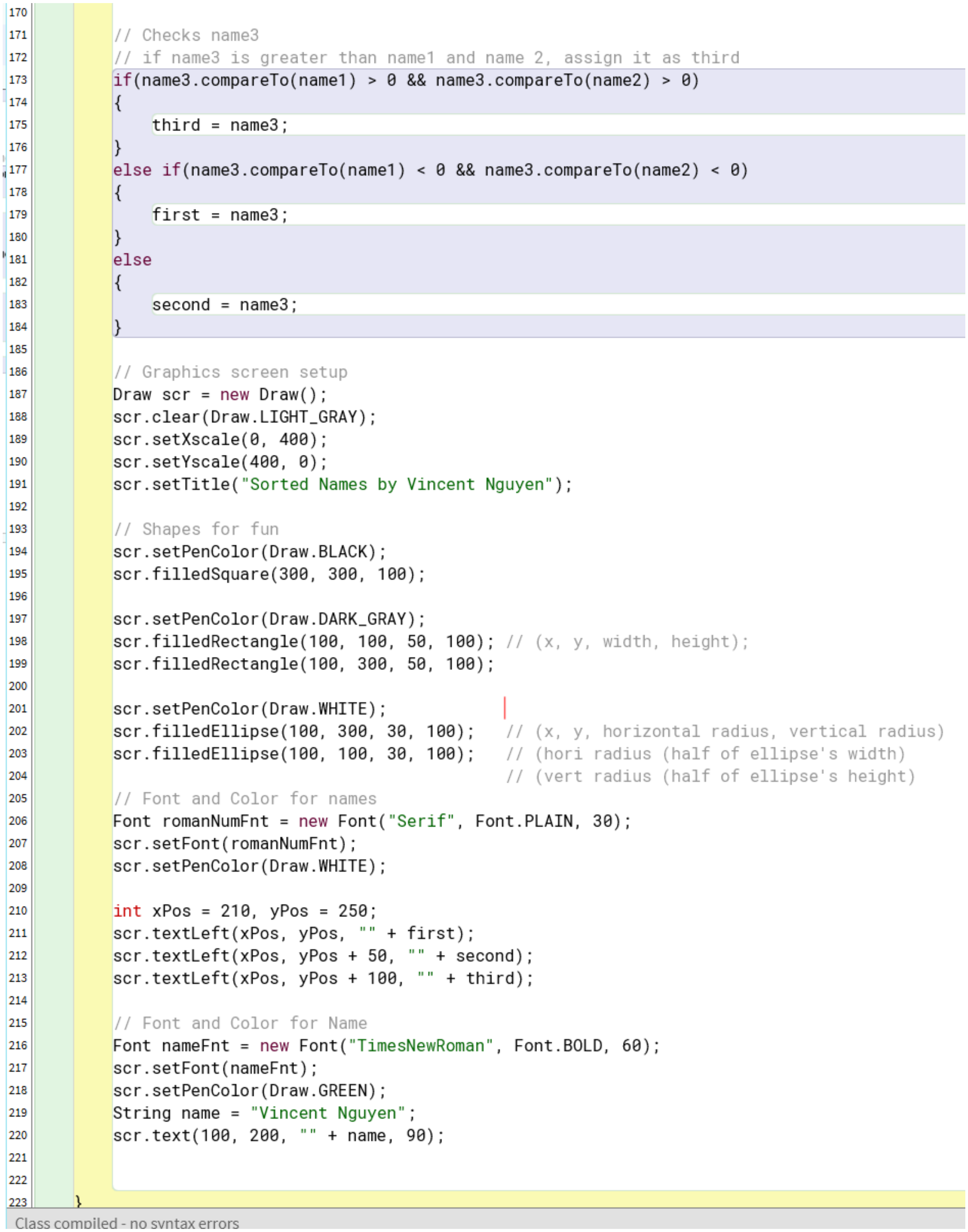
```
Enter a number within the range of 1 and 10:
4

Can only enter input while your program is ru
```



Sorted Names



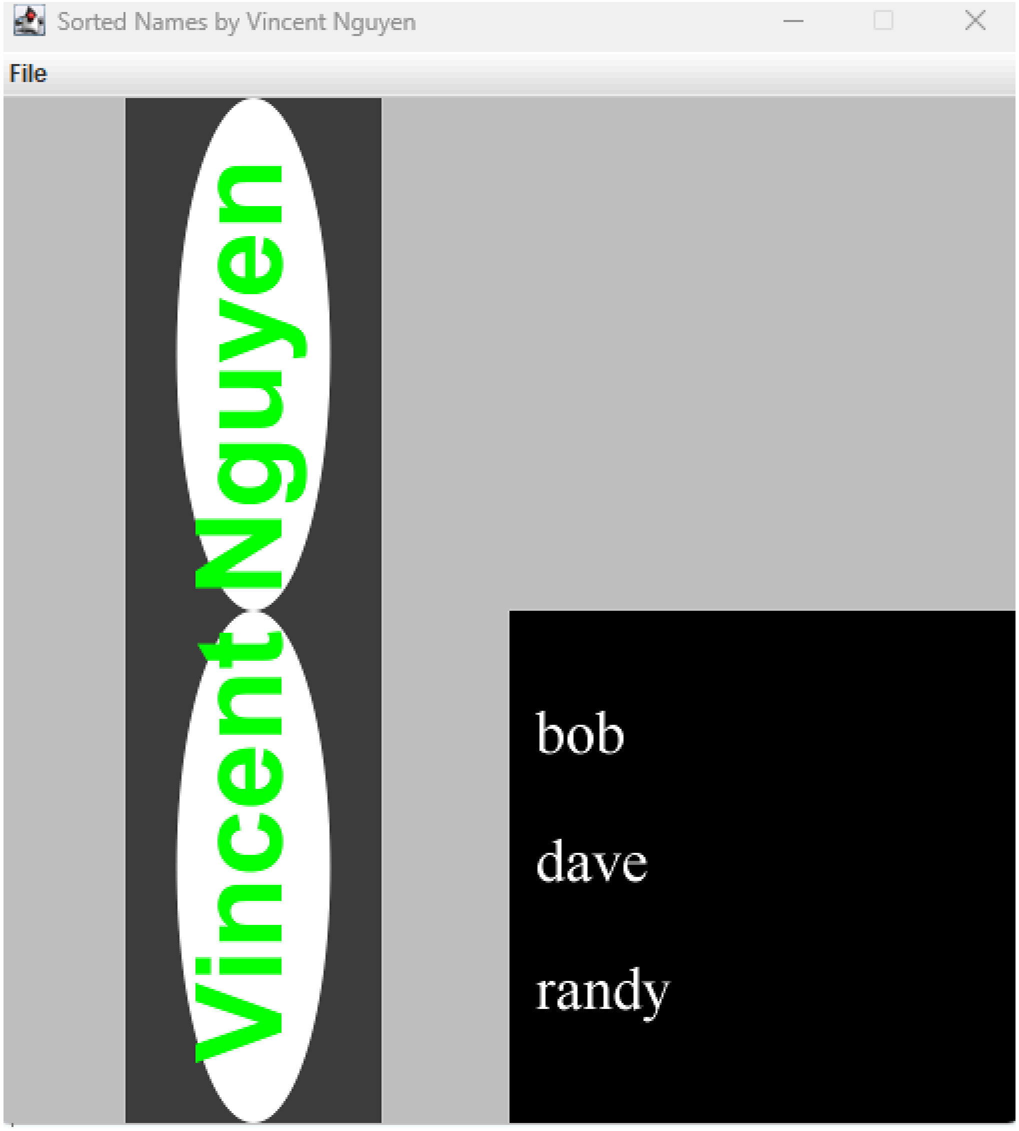
```java
public static void sortedNames()
{
    // Get input from user
    Scanner input = new Scanner(System.in);
    System.out.println("Enter three names");
    String name1 = input.nextLine(); // .nextline for string
    String name2 = input.nextLine(); // .nextDouble for double
    String name3 = input.nextLine(); // .nextFloat for float

    String first = "";
    String second = "";
    String third = "";

    // Checks if name1 is greater than name2 and name3
    // If name1 is greater than name2 and name3 assign it third as highest value
    // Since the desire is for ascending order we want the third value highest to
    // lowest
    if(name1.compareTo(name2) > 0 && name1.compareTo(name3) > 0)
    {
        third = name1;
    }

    // if name1 is the lower than name2 and name3 assign it first as lowest value
    else if(name1.compareTo(name2) < 0 && name1.compareTo(name3) < 0)
    {
        first = name1;
    }
    // if the first two conditions are not met then that means name1 is neither
    // greater than both or lower than both
    else
    {
        second = name1;
    }


    // Checks name2
    // if name2 is greater than name1 and name3, assign to third
    if(name2.compareTo(name1) > 0 && name2.compareTo(name3) > 0)
    {
        third = name2;
    }

    else if(name2.compareTo(name1) < 0 && name2.compareTo(name3) < 0)
    {
        first = name2;
    }
    else
    {
        second = name2;
    }
```



```

        // Checks name3
        // if name3 is greater than name1 and name 2, assign it as third
        if(name3.compareTo(name1) > 0 && name3.compareTo(name2) > 0)
        {
            third = name3;
        }
        else if(name3.compareTo(name1) < 0 && name3.compareTo(name2) < 0)
        {
            first = name3;
        }
        else
        {
            second = name3;
        }

        // Graphics screen setup
        Draw scr = new Draw();
        scr.clear(Draw.LIGHT_GRAY);
        scr.setXscale(0, 400);
        scr.setYscale(400, 0);
        scr.setTitle("Sorted Names by Vincent Nguyen");

        // Shapes for fun
        scr.setPenColor(Draw.BLACK);
        scr.filledSquare(300, 300, 100);

        scr.setPenColor(Draw.DARK_GRAY);
        scr.filledRectangle(100, 100, 50, 100); // (x, y, width, height);
        scr.filledRectangle(100, 300, 50, 100);

        scr.setPenColor(Draw.WHITE);
        scr.filledEllipse(100, 300, 30, 100);   // (x, y, horizontal radius, vertical radius)
        scr.filledEllipse(100, 100, 30, 100);   // (hori radius (half of ellipse's width)
                                                // (vert radius (half of ellipse's height)
        // Font and Color for names
        Font romanNumFnt = new Font("Serif", Font.PLAIN, 30);
        scr.setFont(romanNumFnt);
        scr.setPenColor(Draw.WHITE);

        int xPos = 210, yPos = 250;
        scr.textLeft(xPos, yPos, "" + first);
        scr.textLeft(xPos, yPos + 50, "" + second);
        scr.textLeft(xPos, yPos + 100, "" + third);

        // Font and Color for Name
        Font nameFnt = new Font("TimesNewRoman", Font.BOLD, 60);
        scr.setFont(nameFnt);
        scr.setPenColor(Draw.GREEN);
        String name = "Vincent Nguyen";
        scr.text(100, 200, "" + name, 90);


    }
```

Class compiled - no syntax errors

Graphics output for sorted names



bob

dave

randy