**Name:**

**Advanced Programming in Java**

**Lab Exercise 12.18.2023**

**Building Graphics Programs**

In this example we will create a window that draws a variety of geometric shapes

import javax.swing.\*; //for JFrame

import java.awt.\*; //for Graphics and Container

public class GraphicsDemo {

public static void main(String[] args) {

JFrame win;

Container contentPane;

Graphics g;

win = new JFrame("My Graphics");

win.setSize(300, 300);

win.setLocation(100, 100);

win.setVisible(true);

contentPane = win.getContentPane();

g = contentPane.getGraphics();

//Add this line if you do not see the graphics

try {Thread.sleep(200);} catch (Exception e) {}

g.drawRect(50, 50, 100, 30); //x, y, width, height

g.drawLine(0,0,300,300);

g.drawRoundRect(100, 100, 100, 50, 30, 30);

g.drawOval(150, 170, 100, 50);

}

}

In this example, we will create more geometric shapes that will employ color as well as solid fill.

import javax.swing.\*; //For JFrame

import java.awt.\*; //For Container and Graphics

public class GraphicsDemo2 {

public static void main(String[] args) {

JFrame win;

Container contentPane;

Graphics g;

win = new JFrame("Rectangles");

win.setSize(300, 200);

win.setLocation(100,100);

win.setVisible(true);

contentPane = win.getContentPane();

contentPane.setBackground(Color.LIGHT\_GRAY);

//add this line if you do not see rectangles

try {Thread.sleep(200);} catch (Exception e) {}

g = contentPane.getGraphics();

g.setColor(Color.blue);

g.drawRect(50,50,100,30);

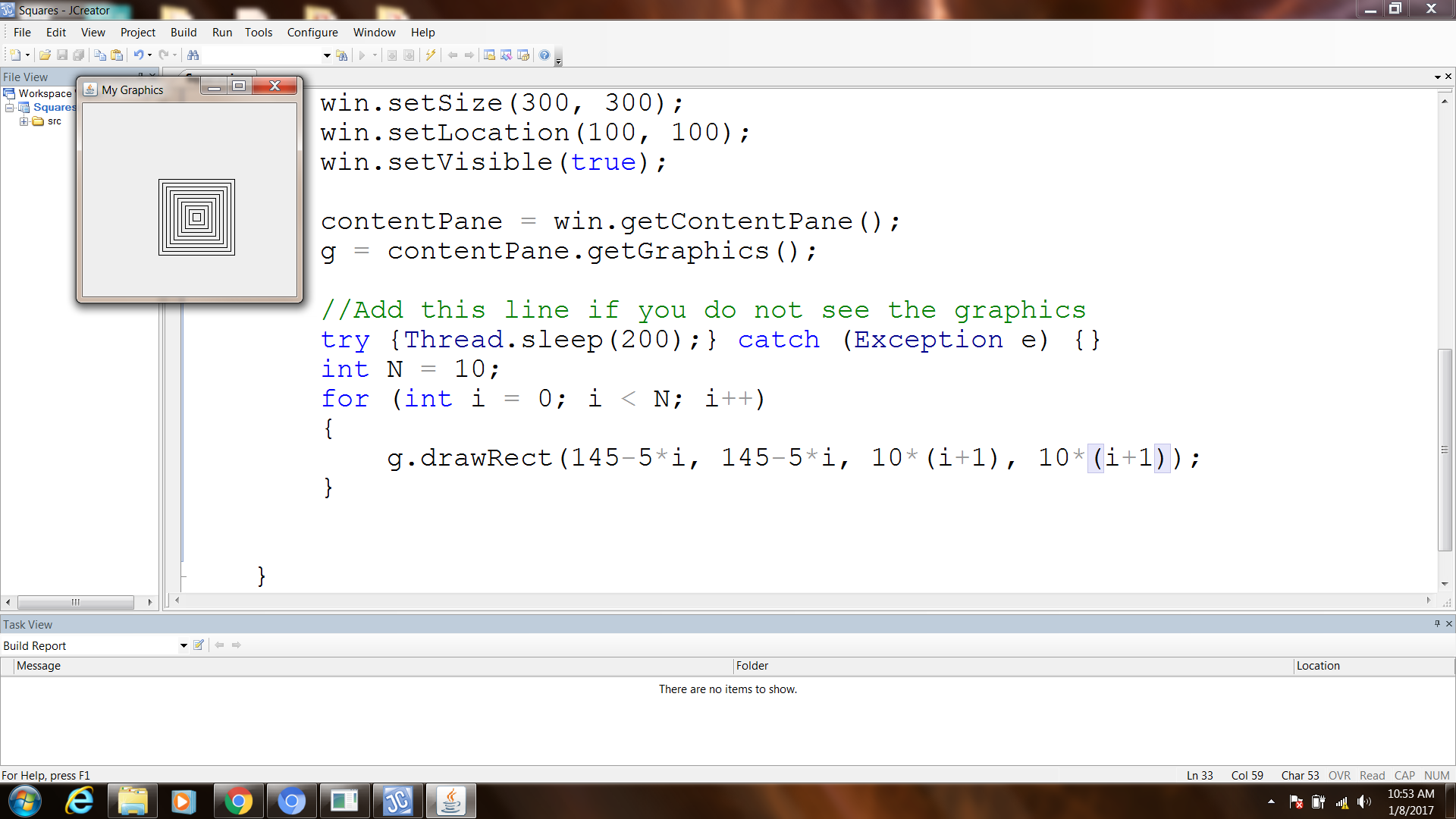
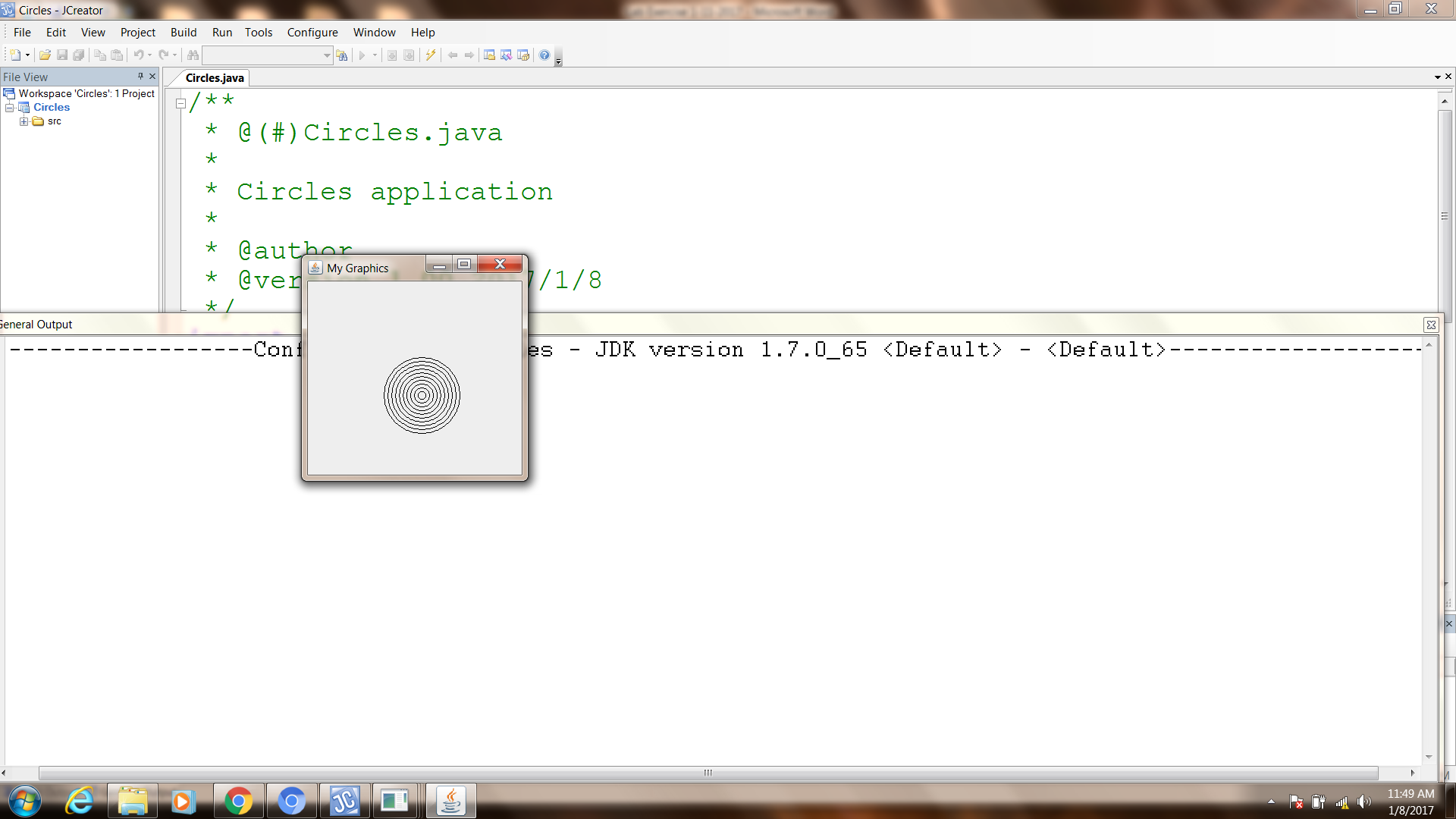
g.setColor(Color.red);

g.fillRect(175,50,100,30);

}

}

1. Write an application that draws N nested squares, where N is the input to the program. The smallest square is 10 pixels wide, and the width of each successive square increases by 10 pixels.

1. Re-write the above application that draws concentric circles instead.
2. Modify #2 to have multi-colored filled circles.

**The Game of Fermi**

Write a program that plays the game of Fermi. Generate three **distinct** random digits between 0 to 9. These digits are assigned to positions 1, 2, and 2. The goal of the game is for the player to guess the digits in the three positions in the least number of tries. For each guess, the player provides three digits for position 1, 2, and 3. The program replies with a hint consisting of Fermi, Pico, and Nano. If the digit guess for a given position is correct, then the reply is Fermi. If the digit guessed for a given position is in a different position, then the reply is Pico. If the digit guessed for a given position does not match any of the three digits, then the reply is Nano. Here is an example. It the three digits are 6, 5, and 8 at positions 1, 2, and 3 respectively.

|  |  |  |
| --- | --- | --- |
| Guess | Hint | Explanation |
| 1 2 5 | Nano Nano Pico | The value 5 matches at the wrong position |
| 8 5 3 | Pico Fermi Nano | The value 5 matches at the correct position. The 8 value matches but at the wrong position |
| 5 8 6 | Pico Pico Pico | All values match but at the wrong position |

Play games repeatedly until the player wants to quit. After each game, display the number of guesses made.

**Failed/Passed**

The code given will output the same middle two lines as in the image shown above.

First parameter is the *user's* score.

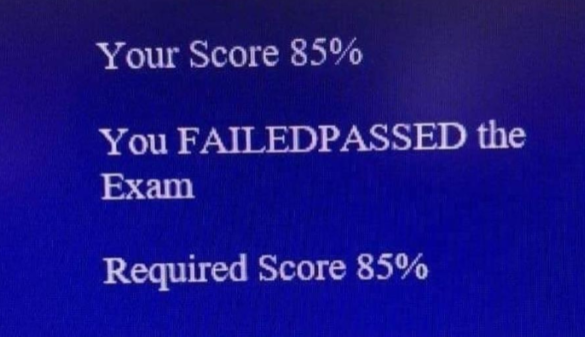
Second parameter is the *required* score.

Examples

grade\_percentage("85%", "85%") ➞ "You PASSED the Exam"

grade\_percentage("99%", "85%") ➞ "You PASSED the Exam"

grade\_percentage("65%", "90%") ➞ "You FAILED the Exam"



Notes

Note that inputs will be given as a string percentage number.