| Student Name | CSUSM account |
|---|---|
| | |
| **Race Bays** | bays001 |
| **Janik Wolf** | wolf038 |
| **Bryce Tulle** | tulle003 |

## Part 1:

Each team works together to write a statement that should include Three parts (or sections).

1. Part one should answer the question
    - What are your team's collective goals?
        - i. We will strive to become very familiar with Software Dev process by
        - ii. Additionally, we will also strive to become familiar with Software Dev tools such as Monday.com, GitHub, Android Studio, SQLite
        - iii. We will develop our skills working as part of a team towards a common goal
        - iv. Develop individually
        - v. Work on Java skills
        - vi. Work on time management
        - vii. Working on estimating tasks
        - viii. Mobile development
        - ix. Database skills
        - x. Well working Simple application
        - xi. UX / UI design
        - xii. Figure out how to push an app to the App Store / Play Store
        - xiii. Grow as a team
2. Part two should answer the question
    - How will your team coordinate project tasks? (Team communication strategies; Get to know and acknowledge Individual strength areas; how to do task assignment)
        - i. Weekends are blocked off
        - ii. Pull tasks from general backlog of tasks
        - iii. Team assigns tasks together
        - iv. Two week sprint
        - v. Bi-weekly scrum meeting, updating tasks, addressing roadblocks
        - vi. Be open with strengths & limitations
        - vii. Task assignment based on familiarity with the task
        - viii. Task assignment based on time estimation
3. Part three should answer the question

- How will your team encourage mutual accountability? Consider the following aspects or beyond
    - i. How to make team decisions in general
        1. Majority vote
    - ii. Individual Commitment to teamwork, how many hours each plans to work on the project.
        1. Plan to spend 6 hours a week from each person
    - iii. Teamwork time: your team should have common time slots to work on projects
        1. Flexible scheduling each week decided at the beginning of the week
    - iv. Potential project risks and strategies to address risks
        1. Risk : Person sick / unable to come to class
        2. Address : They can still be on call and participate in discussion
        3. Risk : Tasks not completed in time, delays in production
        4. Address : Extend work hours to 9hrs/wk, otherwise drop req
        5. Risk : Code not commented, different coding styles
        6. Address : Adopt the Google Java Style Guide

This statement should have at least 600 words.

## Teamwork Statement

We will grow as a team who supports and encourages our members to be their best, building personal and professional connections while developing our technical software development skills using best practices. We will learn from each other's strengths and weaknesses. We will strive to become very familiar with Software Development process by intentionally including the processes and tools we are learning in this class and implementing them in our software development process as the semester continues. Additionally, we will also strive to become familiar with Software Dev tools such as Monday.com, GitHub, Android Studio, SQLite. We will sharpen our programming skills by becoming familiar with the Java programming language. We will develop our skills working as part of a team towards a common goal. We will develop our technical skills by working individually. We will broaden and sharpen our technical skills for developing mobile applications, managing databases, user experience and user interface design, pushing a finished product to the Apple App Store and Google Play Store. We will work on time management and estimating tasks by estimating epics and tasks and evaluating the accuracy of those estimations. We will respect our mental health and individuality by blocking off Saturdays and Sundays as days not to be worked.

We will work on a Two week Sprint cycle. We will meet on a biweekly interval, giving task updates and addressing any roadblocks. When a team member encounters a roadblock, they will let the team know as soon as possible. Software Design Specification (SDS) and Software Requirement Specification (SRS) meetings will be held with all team members, and the product backlog will be filled collaboratively. Sprint planning will be held every two weeks, and the sprint backlog will be filled collaboratively. A sprint retrospective will be held along with sprint planning, where we review our progress, review any roadblocks, and assign tasks on the next sprint according to how much progress was made in the last sprint. We will assign tasks as a team, pulling tasks from general backlog of tasks into a sprint backlog. Tasks will be assigned based on familiarity with the task, and the estimated complexity of the task. Task progression on the sprint backlog will be maintained by the task owners as soon as possible. We will be open with our strengths and our limitations. We will efficiently use team members strengths to avoid individual weaknesses, while at the same time allowing for individual growth by taking on challenging tasks.

We will try to come to a uniform decision on team decisions, making compromises to come to a consensus. If that is not possible, team decisions will be decided with a majority vote. We plan to spend 6 hours a week from each person working on the project. Meetings will be flexibly scheduled each week, decided at the beginning of the week based on team member availability and urgency of the meeting. In the case of a team member being unable to come to class, the group will meet virtually, so they can still be on call and participate in team. In the case of tasks not being completed in time or delays in production, the team will stretch to commit 9 hours per week in an attempt to complete the task, otherwise the feature will be not included in the delivered product. In order to reduce confusion and make the codebase as uniform as possible, we will adopt the Google Java Style Guide for all project code submitted by the team. We will also be utilizing the Pull Request feature on GitHub to give feedback and review team member code before it is approved for merging into the production branch.

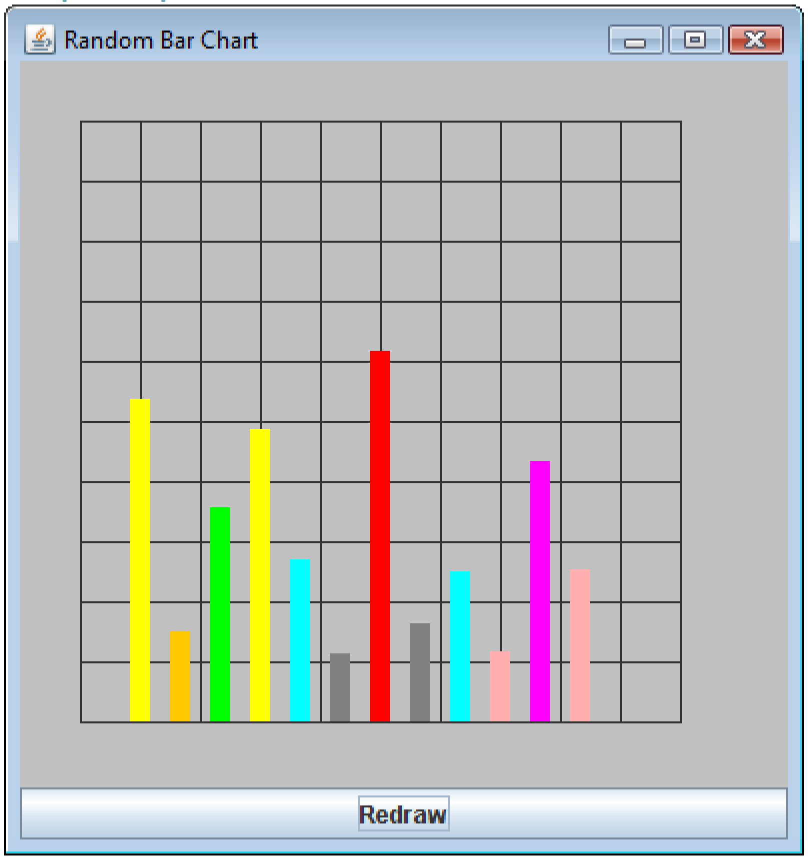
## Part 2: Java Programming

### Problem Description

Write an application that

- The application includes a user-defined JFrame object, a user-defined JPanel object, and a JButton.
- The JPanel should have a `paintComponent(Graphics g) method,` which can draw background 10X10 grids, and draw lines of random lengths and colors (thickness be 10f). Use class Line2D.Double and method draw of class Graphics2D to draw the lines;
- Every time user clicks the Redraw button, the program should randomly redraw the lines again.

### Sample Output



**Solution:**

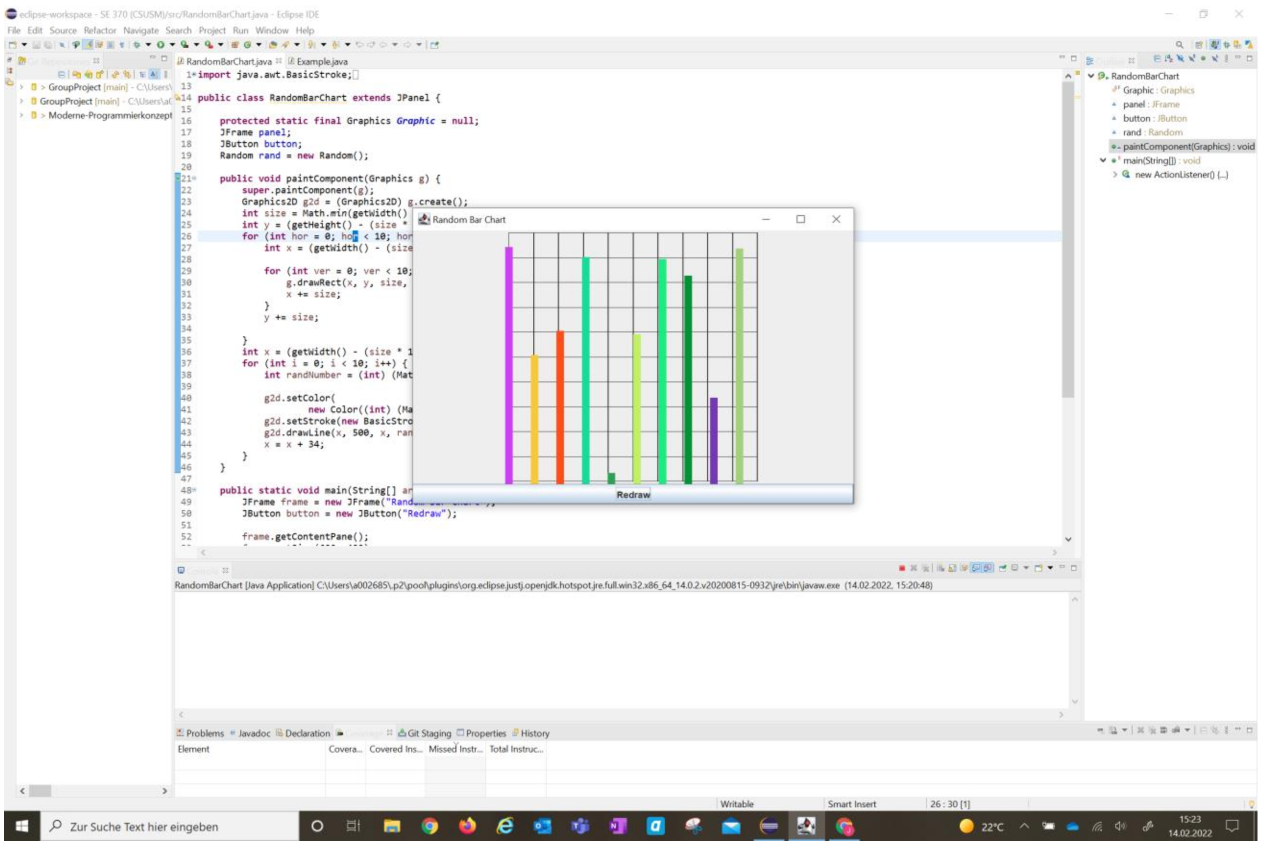
- First, remember to zip the src folder of your project and submit the zip file to the ungraded assignment named “Lab2CodeSubmission”. One submission from each team.

Paste all you source code here (for grading and comments).

```
import java.awt.BasicStroke;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class RandomBarChart extends JPanel{

        protected static final Graphics Graphic = null;
        JFrame panel;
        JButton button;
        Random rand = new Random();


        public RandomBarChart(){


        }

        public void paintComponent(Graphics g) {
                 super.paintComponent(g);
                 int baseline = 200;
          Graphics2D g2d = (Graphics2D) g.create();
          int size = Math.min(getWidth() - 4, getHeight() - 4) / 10;
          int y = (getHeight() - (size * 10)) / 2;
          for (int hor = 0; hor < 10; hor++) {
              int x = (getWidth() - (size * 10)) / 2;

              for (int ver = 0; ver < 10; ver++) {
                  g.drawRect(x, y, size, size);


                  x += size;
              }
              y += size;
```

```
            }
        int x = (getWidth() - (size * 10)) / 2;
        for (int i = 0; i<10; i++) {
                int randNumber = (int) (Math.random()*400);

                g2d.setColor(new
Color((int)(Math.random()*256),(int)(Math.random()*256),(int)(Math.random()*2
56)));
                g2d.setStroke(new BasicStroke(10));
                g2d.drawLine(x,500,x,randNumber);

            x= x+34;

        }

       }


public static void main(String[] args) {
        JFrame frame = new JFrame("Random Bar Chart");
        JButton button = new JButton("Redraw");

        frame.getContentPane();
        frame.setSize(600,400);
        frame.setLayout(new BorderLayout());
        frame.add(new RandomBarChart(), BorderLayout.CENTER);
        frame.add(button, BorderLayout.SOUTH);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
        button.addActionListener( new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent arg0) {
                        frame.repaint();

                }

        });
}
}
```

- Paste a screenshot of your system running.



```
import java.awt.BasicStroke;

public class RandomBarChart extends JPanel {

    protected static final Graphics Graphic = null;
    JFrame panel;
    JButton button;
    Random rand = new Random();

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g.create();
        int size = Math.min(getWidth()
        int y = (getHeight() - (size *
        for (int hor = 0; hor < 10; hor
            int x = (getWidth() - (size

            for (int ver = 0; ver < 10;
                g.drawRect(x, y, size,
                x += size;
            }
            y += size;

        }
        int x = (getWidth() - (size * 1
        for (int i = 0; i < 10; i++) {
            int randNumber = (int) (Mat

            g2d.setColor(
                    new Color((int) (Ma
            g2d.setStroke(new BasicStro
            g2d.drawLine(x, 500, x, ran
            x = x + 34;
        }
    }

    public static void main(String[] ar
        JFrame frame = new JFrame("Rand
        JButton button = new JButton("Redraw");

        frame.getContentPane();
```

```
RandomBarChart [Java Application] C:\Users\a002685\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v20200815-0932\jre\bin\javaw.exe (14.02.2022, 15:20:48)
```

- Save this report in PDF, then **each team** submit your pdf report to the graded assignment named "Lab2ReportSubmission". One submission from each team.