# CPSC 1181 – Lab 9 [40 marks]

**Objectives:**

- Build a an application that uses threads to animate a collection of bouncing balls
- Use the MouseAdapter class to handle events

**Submission:**

- Zip up all of Java files and submit them to D2L prior to the due date.
- Unzipped submissions or submissions containing only .class files will be automatically given ZERO

**Overview**

In this lab you will be reproducing the BouncingBall example from class using separate Threads to animate each ball on the panel.

**Exercise 1**

Create a class called Ball. Create a constructor that gives the ball has the following attributes:

- an x position, a y position, passed as parameters
- a JComponent which represents the component the ball will be animated upon, also passed as a parameter
- a fixed diameter and random color, say 25px and red, or your choice
- a xVelocity and yVelocity, each set randomly to be between +5 or -5

**Exercise 2**

Add two methods to the Ball class. First, create a method called move(). This method should check that the ball's (x,y) position is currently within the bounds of the panel and if it is not, reverse the corresponding velocity. Then, increment the x and y coordinates of the ball by their corresponding velocities.

Then create a method called draw(Graphics2D g2). This method should draw and fill a circle that represents the ball, and do nothing else.

**Exercise 3**

Create a class called BallComponent that is a subclass of JPanel. This class is going to be responsible for creating, storing, and drawing Ball objects.

This class should contain as an attribute an ArrayList of Ball objects that is initially empty. In the constructor set the BallComponent panel to be 500 by 500 and have a black background.

Create the paintComponent(Graphics g) method for this panel. Remember to cast the Graphics object to a Graphics2D object and call the parent class' paintComponent method. Finally, this method should tell each Ball in the ArrayList to **draw itself.**

**Exercise 4**

Make the Ball class implement the runnable interface. Do this by adding a run() method that performs the following tasks *while the Thread is not interrupted*:

- move the ball
- repaint the instance variable component
- put the thread to sleep for 60ms

**Exercise 5**

# CPSC 1181 - Lab 9 [40 marks]

In the BallComponent class, use the MouseAdapter class to create an event listener that detects when we click on the panel. When a click occurs, we should perform the following tasks:

- Instantiate a new Ball object whose (x,y) position is the position where the click occurred
- Add the ball to the ArrayList
- Create a new Thread passing the new ball as a parameter
- Start the thread

**Exercise**
Finally, create a frame with the usual properties to display your BallComponent panel in.

## Marking Rubric:

**Style, Convention, Documentation [6 marks]**

**Ball [20 marks]**
+2 implement Runnable
+3 instance data
+4 constructor
+3 move
+2 draw
+6 run

**BallComponent [11 marks]**
+1 ArrayList correctly initialized
+2 constructor
+2 paintComponent
+1 mouse listener using adapter class
+5 mouse clicked and correct operations
**0 marks will be awarded if Timer is used or if Threads are not used**

**Frame [3 marks]**
+1 create frame with title and default close
+1 add correct component
+1 pack and setVisible