3. Exceptions

a) Write a program to raise built-in exceptions and raise them as per the requirements.

```
public class Pr3a
{
  public static void main(String[ ] args)
  {
        try
        {
              int result = divide(10, 0);
              System.out.println("Result: " + result);
        }
        catch (ArithmeticException e)
        {
              System.err.println("Error: Division by zero.");
        }
```

```
    }
    public static int divide(int a, int b)
    {
       return a / b;
    }
}
```

Output :

```
D:\NewJavaP>javac Pr3a.java

D:\NewJavaP>java Pr3a
Error: Division by zero.
```

b) Write a program to define user defined exceptions and raise them as per the requirements.

```
// Define a custom exception class
class CustomException extends Exception
{
        public CustomException(String message)
        {
            super(message);
        }
}
public class Pr3b
{
    public static void main(String[ ] args)
    {
            try
            {
                int age = -20;
                // Check if age is negative
                if (age < 0)
                {
                    throw new CustomException("Age cannot be negative.");
                }
                System.out.println("Age: " + age);
            }
            catch (CustomException e)
            {
                System.err.println("Error: " + e.getMessage());
            }
    }
}
```

Output :

```
D:\NewJavaP>javac Pr3b.java

D:\NewJavaP>java Pr3b
Error: Age cannot be negative.
```

4. **Multithreading :** Write a java application to demonstrate multiple bouncing balls of different colors using threads.

```
import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class BouncingBalls extends JPanel implements Runnable
{
  public static final int WIDTH = 800;
  public static final int HEIGHT = 600;
  private static final int NUM_BALLS = 5;

  private List<Ball> balls;

  public BouncingBalls( )
  {
    balls = new ArrayList<>( );

    // Create and add random balls
    Random random = new Random( );
    for (int i = 0; i < NUM_BALLS; i++)
      {
        int x = random.nextInt(WIDTH);
        int y = random.nextInt(HEIGHT);
        int xSpeed = random.nextInt(5) + 1;
        int ySpeed = random.nextInt(5) + 1;
        Color color = new Color(random.nextInt(256), random.nextInt(256),
random.nextInt(256));
        balls.add(new Ball(x, y, xSpeed, ySpeed, color));
      }
  }
  @Override
  public void run( )
```

```
    {
      while (true)
      {
        for (Ball ball : balls)
        {
          ball.move( );
        }
        repaint( );
        try
        {
          Thread.sleep(10);
        }
        catch (InterruptedException e)
        {
          e.printStackTrace();
        }
      }
    }
    @Override
    protected void paintComponent(Graphics g)
    {
      super.paintComponent(g);
      for (Ball ball : balls)
      {
        ball.draw(g);
      }
    }
    public static void main(String[ ] args)
    {
      JFrame frame = new JFrame("5 Colours Bouncing Balls");
      BouncingBalls bouncingBalls = new BouncingBalls( );
      frame.add(bouncingBalls);
      frame.setSize(WIDTH, HEIGHT);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);

      Thread thread = new Thread(bouncingBalls);
      thread.start();
    }
}
```

Output :