Date: 02-06-23

Experiment - 6.1

Aim:

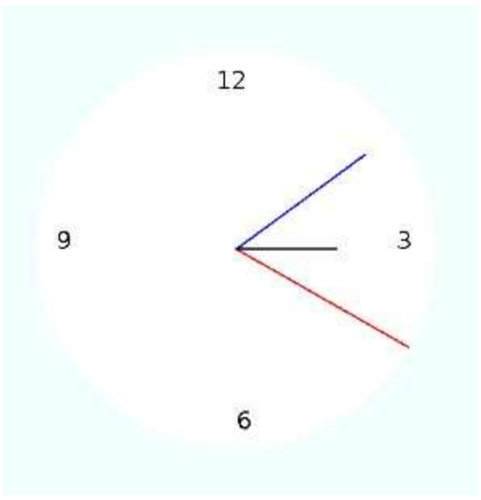
Write a program to develop an analogue clock using applet.

Code:

```
import java.applet.Applet;
import java.awt.*;
import java.util.*;

public class AnalogClock extends Applet {
	public void init(){
		this.setSize(new Dimension(800, 400));
		setBackground(new Color(240, 255, 255));
		new Thread() {
			@Override
			public void run()
			{
				while (true) {
					repaint();
					delayAnimation();
				}
			}
		}.start();
	}

	private void delayAnimation(){
		try {
			Thread.sleep(1000);
		}
		catch (InterruptedException e) {
			e.printStackTrace();
		}
	}

	public void paint(Graphics g) {
		Calendar time = Calendar.getInstance();

		int hour = time.get(Calendar.HOUR_OF_DAY);
		int minute = time.get(Calendar.MINUTE);
		int second = time.get(Calendar.SECOND);

		if (hour > 12) {
			hour -= 12;
		}

		g.setColor(Color.white);
		g.fillOval(300, 100, 200, 200);
```

```
                    g.setColor(Color.black);
                    g.drawString("12", 390, 120);
                    g.drawString("9", 310, 200);
                    g.drawString("6", 400, 290);
                    g.drawString("3", 480, 200);

                    double angle;
                    int x, y;

                    angle = Math.toRadians((15 - second) * 6);

                    x = (int)(Math.cos(angle) * 100);
                    y = (int)(Math.sin(angle) * 100);

                    g.setColor(Color.red);
                    g.drawLine(400, 200, 400 + x, 200 - y);

                    angle = Math.toRadians((15 - minute) * 6);

                    x = (int)(Math.cos(angle) * 80);
                    y = (int)(Math.sin(angle) * 80);

                    g.setColor(Color.blue);
                    g.drawLine(400, 200, 400 + x, 200 - y);

                    angle = Math.toRadians((15 - (hour * 5)) * 6);

                    x = (int)(Math.cos(angle) * 50);
                    y = (int)(Math.sin(angle) * 50);

                    g.setColor(Color.black);
                    g.drawLine(400, 200, 400 + x, 200 - y);
          }
}
```

Output:



Date: 02-06-23

Experiment - 6.2

Aim:

Write a Java program to show multithreaded producer and consumer application.

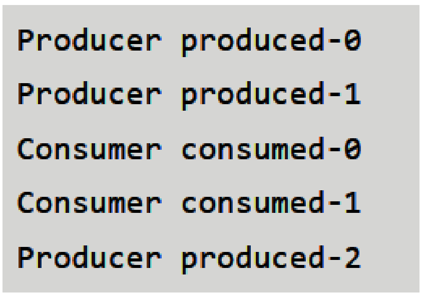
Code:

```
import java.util.LinkedList;

class ProducerConsumer {
    private LinkedList<Integer> buffer = new LinkedList<>();
    private int capacity = 5;

    public void produce() throws InterruptedException {
        int value = 0;
        while (true) {
            synchronized (this) {
                while (buffer.size() == capacity) {
                    wait();
                }

                System.out.println("Producer produced: " + value);
                buffer.add(value++);

                notify();

                Thread.sleep(1000);
            }
        }
    }

    public void consume() throws InterruptedException {
        while (true) {
            synchronized (this) {
                while (buffer.isEmpty()) {
                    wait();
                }

                int value = buffer.removeFirst();
                System.out.println("Consumer consumed: " + value);

                notify();

                Thread.sleep(1000);
            }
        }
    }
}

public class Main {
```

Date: 02-06-23

```java
  public static void main(String[] args) {
    ProducerConsumer pc = new ProducerConsumer();

    Thread producerThread = new Thread(() -> {
      try {
        pc.produce();
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    });

    Thread consumerThread = new Thread(() -> {
      try {
        pc.consume();
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    });

    producerThread.start();
    consumerThread.start();
  }
}
```

Output:



```
Producer produced-0
Producer produced-1
Consumer consumed-0
Consumer consumed-1
Producer produced-2
```