6/2/24.

Lab program 8.

```
class college extends thread
{
    public void run()
    {   for(int i=0; i<6; i++){
        System.out.println("BMS college of
                    engineering");

        try
        {

            Thread.sleep(10000);

        }

        catch (InterruptedException e)
        {

            e.printStackTrace();

        }

        }

    }

}

class Dept extends Thread
{

    public void run()
    {

        for(int i=1; i<=10; i++)
        {

            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            }

            catch (InterruptedException e)
            {
```

```
        e. print StackTrace ();
      }
    }
  }
}
class Main
{
    public static void main (String args[])
    {
        college C1 = new college ();
        C1. start ();
        Dept d1 = new Dept ();
        d1. start ();
    }
}
```

output

BMS college g engineering
CSE
CSE
CSE
CSE
SSE
BMS college g engineering
CSE
CSE
CSE
CSE
CSE
BMS college g engineering
BMS college g engineering

Lab program 10

Incorrect implementation of a producer
and consumer.

```
class Q {
    int n;
    synchronized int get() {
        system.out.println("got : " +n);
        return n;
    }
    synchronized void put(int n) {
        this.n=n;
        system.out.println("Put :" +n);
    }
}

class Producer implements Runnable {
    Q q;
    Producer (Q q) {
        this. q =q;
        new thread (this, "Producer"), start();
    }
    public void run() {
        int i =0;
        while (i<15) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;
    consumer (Q q) {
```

```
this.q=q;
new thread (this, "consumer"), start();
}
public void run() {
int i=0;
while (i<15) {
int i=q.get();
i++;
}
}
}
class PC {
    public static void main(String args[])
    {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press control-C to
                            stop.");
    }
}
```

output

```
Press control-C to stop.
Put: 0
Put: 1
Put: 2
Put: 3
Put: 4
Put: 5
put: 6.
```

put : 7

put : 8

put : 9

put : 10

put : 11

put : 12

put : 13

put : 14.

got : 14

got : 14

got : 14

got : 14