**Experiment – 12**

Q1) Write a program to create a thread extending Thread class and demonstrate the use of sleep() method

Ans:

**Program:**

package Exp\_12;

class Thread1 extends Thread{

    public void run(){

        for(int i=1;i<=50;i++){

            System.out.println(i+" -> "+Thread.currentThread().getName());

            try{

                Thread.sleep(1000);

            }

            catch(InterruptedException e){

                System.out.println("Error while sleepig");

            }

        }

    }

}

public class First {

    public static void main(String[] args) {

        Thread1 t1 = new Thread1();

        Thread1 t2 = new Thread1();

        t1.start();

        try{

            t1.join();

        }

        catch(InterruptedException e){

            System.out.println("Error while joining");

        }

        t2.start();

    }

}

**Output:**

A computer screen shot of a black screen

AI-generated content may be incorrect.

A black screen with a white border

AI-generated content may be incorrect.

A black screen with a white border

AI-generated content may be incorrect.

A black screen with a white dot

AI-generated content may be incorrect.

Q2) Write a program to create a thread implementing Runnable interface and demonstrate the use of join() methodAns:

**Program:**

package Exp\_12;

class MyThread implements Runnable{

    int start;

    int end;

    MyThread(int start,int end){

        this.start = start;

        this.end = end;

    }

    public void run(){

        for(int i=start;i<=end;i++){

            System.out.println(i+"->"+Thread.currentThread().getName());

        }

    }

}

public class Second {

    public static void main(String[] args) {

        Thread t1 = new Thread(new MyThread(1,100));

        Thread t2 = new Thread(new MyThread(101,200));

        t1.start();

        try{

            t1.join();

        }

        catch(InterruptedException e){

            System.out.println("Error while joining!");

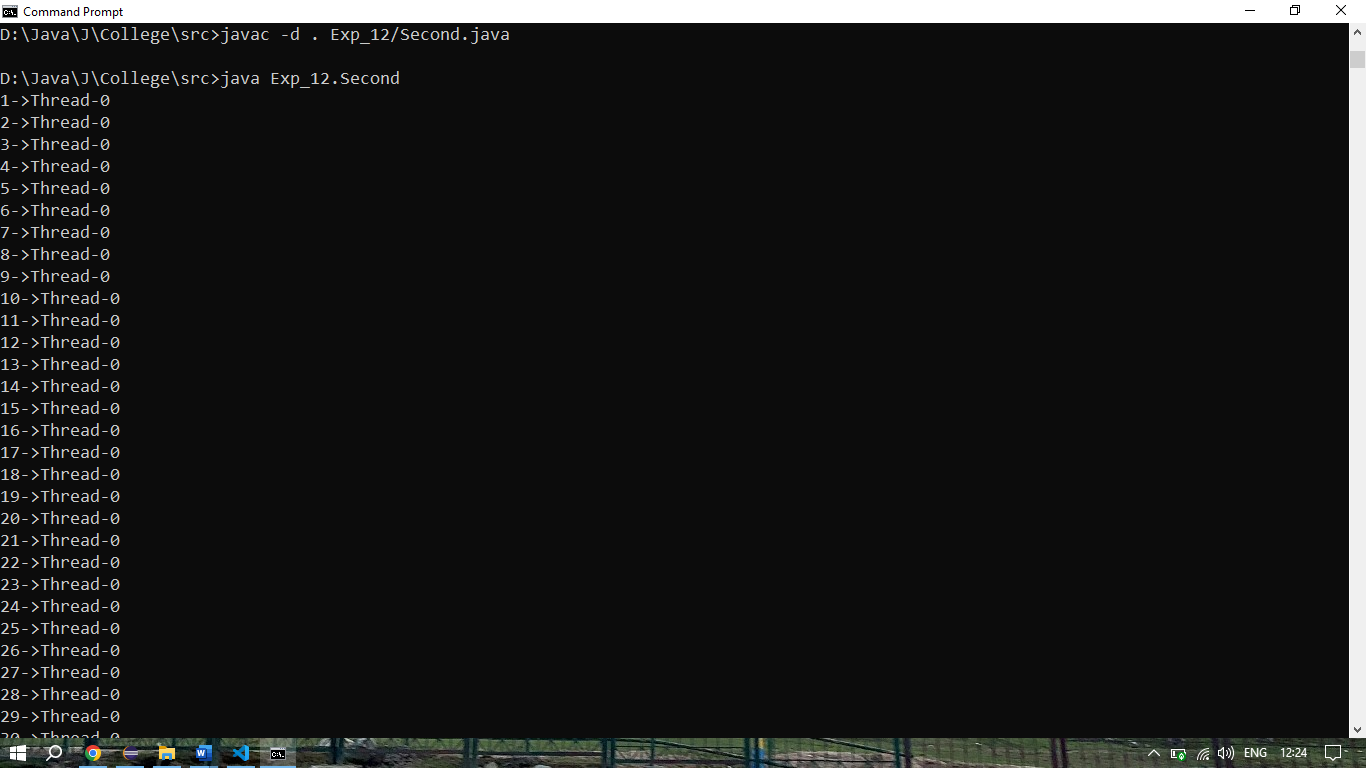
        }

        t2.start();

    }

}

**Output:**



A black screen with a black background

AI-generated content may be incorrect.

A black screen with a black background

AI-generated content may be incorrect.

A black screen with a black background

AI-generated content may be incorrect. A black screen with a black background

AI-generated content may be incorrect. A black screen with a black background

AI-generated content may be incorrect. A black screen with a black background

AI-generated content may be incorrect.

Q3) Write a program that launches 10 threads. Each thread adds 1 to a variable sum that initially is 0. Define an Integer wrapper object to hold sum. Run the program with and without synchronization to see its effect.

Ans:

**Program:**

//Without Sync

package Exp\_12;

class Count{

    private int sum = 0;

    void increment(){

        sum+=1;

    }

    int getSum(){

        return sum;

    }

}

class NewThread extends Thread{

    Count c;

    NewThread(Count c){

        this.c = c;

    }

    public void run(){

        for(int i=1;i<=1000;i++)    c.increment();

    }

}

public class ThirdWithoutSync {

    public static void main(String[] args) {

        Count c = new Count();

        Thread[] threadArray = new Thread[10];

        for(int i=0;i<10;i++){

            threadArray[i] = new NewThread(c);

        }

        for(Thread thread: threadArray){

            thread.start();

        }

        for(Thread thread: threadArray){

            try{

                thread.join();

            }

            catch(InterruptedException e){

                System.out.println("Error while joining");

            }

        }

        System.out.println(c.getSum());

    }

}

// With Sync

package Exp\_12;

class Count{

    private int sum = 0;

    synchronized void increment(){

        sum+=1;

    }

    int getSum(){

        return sum;

    }

}

class NewThread extends Thread{

    Count c;

    NewThread(Count c){

        this.c = c;

    }

    public void run(){

        for(int i=1;i<=1000;i++)    c.increment();

    }

}

public class ThirdWithSync{

    public static void main(String[] args) {

        Count c = new Count();

        Thread[] threadArray = new Thread[10];

        for(int i=0;i<10;i++){

            threadArray[i] = new NewThread(c);

        }

        for(Thread thread: threadArray){

            thread.start();

        }

        for(Thread thread: threadArray){

            try{

                thread.join();

            }

            catch(InterruptedException e){

                System.out.println("Error while joining");

            }

        }

        System.out.println(c.getSum());

    }

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.