Assignment No. 5

Multithreading

Question 1: Write a Java program to create and run a thread by extending the Thread class. The thread should print "Hello from Thread" five times.

```java
class MyThread1 extends Thread {
    MyThread1(String tname) {
        super(tname);
    }

    public void run() {
        System.out.println("Hello from Thread");
    }
}

public class Ques1 {
    public static void main(String[] args) {
        int i = 1;
        while (i <= 5) {
            MyThread1 t1 = new MyThread1("Thread 1");
            t1.start();
            i++;
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques1.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques1
Hello from Thread
Hello from Thread
Hello from Thread
Hello from Thread
Hello from Thread
```

Question 2: Write a Java program to create and run a thread by implementing the Runnable interface. The thread should print numbers from 1 to 5.

```java
class MyThread2 implements Runnable {
    Thread t;

    MyThread2(String tname) {
        t = new Thread(this, tname);
    }
```

```
    public void run() {
        for (int i = 1; i < 6; i++) {
            System.out.println(i);
        }
    }
}

public class Ques2 {
    public static void main(String[] args) {
        MyThread2 t2 = new MyThread2("Thread");
        t2.t.start();
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques2.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques2
1
2
3
4
5
```

Question 3: Write a Java program where the main thread prints "Main Thread Running" and a child thread prints "Child Thread Running". Run them simultaneously.

```
class MyThread3 extends Thread {
    public void run() {
        System.out.println("Child Thread Running");
    }
}

public class Ques3 {
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyThread3());
        t1.start();

        System.out.println("Main Thread Running");
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques3.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques3
Main Thread Running
Child Thread Running
```

Question 4: Write a Java program to demonstrate the use of setName() and getName() methods for threads.

```java
class MyThread4 extends Thread {
    public void run() {
        System.out.println("Thread is running");
    }
}

public class Ques4 {
    public static void main(String[] args) {
        MyThread4 t1 = new MyThread4();
        MyThread4 t2 = new MyThread4();

        System.out.println("Thread 1: " + t1.getName());
        System.out.println("Thread 2: " + t2.getName());

        t1.start();
        t2.start();

        t1.setName("MyThread4 Thread1");
        t2.setName("MyThread4 Thread2");

        System.out.println("\nAfter changing thread names");
        System.out.println("Thread 1: " + t1.getName());
        System.out.println("Thread 2: " + t2.getName());
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques4.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques4
Thread 1: Thread-0
Thread 2: Thread-1

After changing thread names
Thread 1: MyThread4 Thread1
Thread 2: MyThread4 Thread2
Thread is running
Thread is running
```

Question 5: Write a Java program to demonstrate the use of setPriority() and getPriority() methods by creating two threads with different priorities.

```java
class MyThread5 extends Thread {
    public void run() {
        System.out.println("\n" + Thread.currentThread().getName() + " is
running with priority "
                + Thread.currentThread().getPriority());
    }
}

public class Ques5 {
    public static void main(String[] args) {
        MyThread5 t0 = new MyThread5();
        MyThread5 t1 = new MyThread5();

        System.out.println("Before setting priority");
        System.out.println("t0 priority: " + t0.getPriority());
        System.out.println("t1 priority: " + t1.getPriority());

        t0.setPriority(6);
        t1.setPriority(1);

        System.out.println("\nAfter setting priority");
        System.out.println("t0 priority: " + t0.getPriority());
        System.out.println("t1 priority: " + t1.getPriority());

        t0.start();
        t1.start();
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques5.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques5
Before setting priority
t0 priority: 5
t1 priority: 5

After setting priority
t0 priority: 6
t1 priority: 1

Thread-0 is running with priority 6

Thread-1 is running with priority 1
```

Question 6: Write a Java program where one thread prints numbers from 1 to 10, and another thread prints numbers from 11 to 20.

```java
class MyThread6_1 implements Runnable {
    Thread t;

    MyThread6_1(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        for (int i = 1; i <= 10; i++) {
            System.out.println(Thread.currentThread().getName() + " : " + i);
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

class MyThread6_2 implements Runnable {
    Thread t;

    MyThread6_2(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
```

```
        for (int i = 11; i <= 20; i++) {
            System.out.println(Thread.currentThread().getName() + " : " + i);
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

public class Ques6 {
    public static void main(String[] args) {
        try {
            MyThread6_1 t1 = new MyThread6_1("Thread 1");
            MyThread6_2 t2 = new MyThread6_2("Thread 2");

            t1.t.start();
            t1.t.join();

            System.out.println();

            t2.t.start();
            t2.t.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques6.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques6
Thread 1 : 1
Thread 1 : 2
Thread 1 : 3
Thread 1 : 4
Thread 1 : 5
Thread 1 : 6
Thread 1 : 7
Thread 1 : 8
Thread 1 : 9
Thread 1 : 10
```

```
Thread 2 : 11
Thread 2 : 12
Thread 2 : 13
Thread 2 : 14
Thread 2 : 15
Thread 2 : 16
Thread 2 : 17
Thread 2 : 18
Thread 2 : 19
Thread 2 : 20
```

Question 7: Write a Java program to demonstrate the use of the sleep() method by pausing a thread for 1 second after printing each number.

```
class MyThread7 implements Runnable {
    Thread t;

    MyThread7(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        for (int i = 1; i <= 10; i++) {
            System.out.println(Thread.currentThread().getName() + " : " + i);
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

public class Ques7 {
    public static void main(String[] args) {
        try {
            MyThread6_1 t1 = new MyThread6_1("Thread 1");
            t1.t.start();
            t1.t.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques7.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques7
Thread 1 : 1
Thread 1 : 2
Thread 1 : 3
Thread 1 : 4
Thread 1 : 5
Thread 1 : 6
Thread 1 : 7
Thread 1 : 8
Thread 1 : 9
Thread 1 : 10
```

Question 8: Write a Java program where the main thread waits for a child thread to finish using the join() method.

```java
class MyThread8 extends Thread {
    public void run() {
        System.out.println("Child Thread Running");
    }
}

public class Ques8 {
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyThread8());
        t1.start();
        try {
            t1.join();
            System.out.println("Child thread finished running");
        } catch (Exception e) {
            System.out.println(e);
        }

        System.out.println("Main Thread Running");
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques8.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques8
Child Thread Running
Child thread finished running
Main Thread Running
```

Question 9: Write a Java program to check whether a thread is alive or not using the isAlive() method.

```
class MyThread9 extends Thread {
    public void run() {
        System.out.println(Thread.currentThread().getName() + " is running");
    }
}

public class Ques9 {
    public static void main(String[] args) {
        MyThread9 t = new MyThread9();
        t.start();
        System.out.println("Is thread alive? " + t.isAlive());
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques9.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques9
Thread-0 is running
Is thread alive? true
```

Question 10: Write a Java program to create two threads:

- **Thread 1 prints "Good Morning" 5 times.**
- **Thread 2 prints "Welcome" 5 times. Run both threads simultaneously.**

```
class MyThread10_1 implements Runnable {
    Thread t;

    MyThread10_1(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            System.out.println(Thread.currentThread().getName() + ": Good
Morning");
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
```

```
    }
}

class MyThread10_2 implements Runnable {
    Thread t;

    MyThread10_2(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            System.out.println(Thread.currentThread().getName() + ":
Welcome");
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

public class Ques10 {
    public static void main(String[] args) {
        MyThread10_1 t1 = new MyThread10_1("Thread 1");
        MyThread10_2 t2 = new MyThread10_2("Thread 2");
        t1.t.start();
        t2.t.start();

        try {
            t1.t.join();
            t2.t.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques10.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques10
Thread 2: Welcome
Thread 1: Good Morning
Thread 2: Welcome
Thread 1: Good Morning
Thread 2: Welcome
Thread 1: Good Morning
Thread 1: Good Morning
Thread 2: Welcome
Thread 2: Welcome
Thread 1: Good Morning
```

Question 11: Write a Java program where one thread prints even numbers from 2 to 20, and another thread prints odd numbers from 1 to 19.

```
class MyThread11_1 implements Runnable {
    Thread t;

    MyThread11_1(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        for (int i = 1; i <= 20; i++) {
            if (i % 2 == 0) {
                System.out.println(Thread.currentThread().getName() + " : " +
i);
            }
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

class MyThread11_2 implements Runnable {
    Thread t;

    MyThread11_2(String tname) {
        t = new Thread(this, tname);
    }
```

```
    public void run() {
        for (int i = 1; i <= 20; i++) {
            if (i % 2 != 0) {
                System.out.println(Thread.currentThread().getName() + " : " +
i);
            }
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

public class Ques11 {
    public static void main(String[] args) {
        MyThread11_1 t1 = new MyThread11_1("Even numbers from 2 to 20");
        MyThread11_2 t2 = new MyThread11_2("Odd numbers from 1 to 19");
        t1.t.start();
        t2.t.start();

        try {
            t1.t.join();
            t2.t.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques11.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques11
Odd numbers from 1 to 19 : 1
Even numbers from 2 to 20 : 2
Odd numbers from 1 to 19 : 3
Even numbers from 2 to 20 : 4
Odd numbers from 1 to 19 : 5
Even numbers from 2 to 20 : 6
Odd numbers from 1 to 19 : 7
Even numbers from 2 to 20 : 8
Odd numbers from 1 to 19 : 9
Even numbers from 2 to 20 : 10
Odd numbers from 1 to 19 : 11
Even numbers from 2 to 20 : 12
Odd numbers from 1 to 19 : 13
Even numbers from 2 to 20 : 14
Odd numbers from 1 to 19 : 15
Even numbers from 2 to 20 : 16
Odd numbers from 1 to 19 : 17
Even numbers from 2 to 20 : 18
Odd numbers from 1 to 19 : 19
Even numbers from 2 to 20 : 20
```

Question 12: Write a Java program to create three threads. Each thread should print its own message 3 times.

```java
class MyThread12 extends Thread {
    String msg;

    MyThread12(String tname, String msg) {
        super(tname);
        this.msg = msg;
    }

    public void run() {
        for (int i = 0; i < 3; i++) {
            System.out.println(Thread.currentThread().getName() + ": " + msg);
            try {
                sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
```

```
        }
    }
}

public class Ques12 {
    public static void main(String[] args) {
        MyThread12 t1 = new MyThread12("A", "Hello! I'm Thread 1");
        MyThread12 t2 = new MyThread12("B", "Bonjour! I'm Thread 2");
        MyThread12 t3 = new MyThread12("C", "Ni Hao! I'm Thread 3");

        t1.start();
        t2.start();
        t3.start();

        try {
            t1.join();
            t2.join();
            t3.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques12.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques12
A: Hello! I'm Thread 1
B: Bonjour! I'm Thread 2
C: Ni Hao! I'm Thread 3
B: Bonjour! I'm Thread 2
C: Ni Hao! I'm Thread 3
A: Hello! I'm Thread 1
C: Ni Hao! I'm Thread 3
A: Hello! I'm Thread 1
B: Bonjour! I'm Thread 2
```

Question 13: Write a Java program to demonstrate the difference between calling run() directly and calling start() on a thread.

```
class MyThread13 extends Thread {
    String msg;

    MyThread13(String msg) {
        this.msg = msg;
```

```
    }

    public void run() {
        System.out.println(Thread.currentThread().getName() + ": " + msg);
    }
}

public class Ques13 {
    public static void main(String[] args) {
        MyThread13 t1 = new MyThread13("Called using run()");
        t1.run();

        MyThread13 t2 = new MyThread13("Called using start()");
        t2.start();
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques13.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques13
main: Called using run()
Thread-1: Called using start()
```

Question 14: Write a Java program to create a thread that calculates the sum of numbers from 1 to 100.

```
class MyThread14 extends Thread {
    public void run() {
        int sum = 0;
        for (int i = 1; i <= 100; i++) {
            sum += i;
        }

        System.out.println(Thread.currentThread().getName() + ": " + sum);
        try {
            Thread.sleep(100);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

public class Ques14 {
    public static void main(String[] args) {
        MyThread14 t1 = new MyThread14();
        t1.start();
```

```
        try {
            t1.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques14.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques14
Thread-0: 5050
```

Question 15: Write a Java program to demonstrate how to stop a thread gracefully using a boolean flag instead of the deprecated stop() method.

```
class MyThread15 implements Runnable {
    Thread t;
    private boolean flag = true;

    MyThread15(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        int i = 0;
        while (flag) {
            System.out.println(Thread.currentThread().getName() + ": " + i);
            i++;
            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
        System.out.println(Thread.currentThread().getName() + " stopped!");
    }

    public void stopThread() {
        flag = false;
    }
}

public class Ques15 {
    public static void main(String[] args) {
        MyThread15 t1 = new MyThread15("Thread 1");
        MyThread15 t2 = new MyThread15("Thread 2");
```

```
        t1.t.start();
        t2.t.start();

        try {
            Thread.sleep(1000);
            t1.stopThread();
            t2.stopThread();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques15.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques15
Thread 2: 0
Thread 1: 0
Thread 2: 1
Thread 1: 1
Thread 2: 2
Thread 1: 2
Thread 1: 3
Thread 2: 3
Thread 2: 4
Thread 1: 4
Thread 2: 5
Thread 1: 5
Thread 2: 6
Thread 1: 6
Thread 2: 7
Thread 1: 7
Thread 2: 8
Thread 1: 8
Thread 2: 9
Thread 1: 9
Thread 1 stopped!
Thread 2 stopped!
```

Question 16: Write a Java program where one thread prints the lowercase alphabet (a to z), and another thread prints the uppercase alphabet (A to Z).

```
class MyThread16_1 implements Runnable {
    Thread t;

    MyThread16_1(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        for (int i = 65; i <= 90; i++) {
            System.out.println(Thread.currentThread().getName() + " : " +
(char) i);

            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

class MyThread16_2 implements Runnable {
    Thread t;

    MyThread16_2(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        for (int i = 97; i <= 122; i++) {
            System.out.println(Thread.currentThread().getName() + " : " +
(char) i);

            try {
                Thread.sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

public class Ques16 {
    public static void main(String[] args) {
        MyThread16_1 t1 = new MyThread16_1("Uppercase Alphabet");
        MyThread16_2 t2 = new MyThread16_2("Lowercase Alphabet");
```

```
        t1.t.start();
        t2.t.start();

        try {
            t1.t.join();
            t2.t.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques16.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques16
Uppercase Alphabet : A
Lowercase Alphabet : a
Uppercase Alphabet : B
Lowercase Alphabet : b
Lowercase Alphabet : c
Uppercase Alphabet : C
Uppercase Alphabet : D
Lowercase Alphabet : d
Uppercase Alphabet : E
Lowercase Alphabet : e
Lowercase Alphabet : f
Uppercase Alphabet : F
Uppercase Alphabet : G
Lowercase Alphabet : g
Lowercase Alphabet : h
Uppercase Alphabet : H
Uppercase Alphabet : I
Lowercase Alphabet : i
Lowercase Alphabet : j
Uppercase Alphabet : J
Uppercase Alphabet : K
Lowercase Alphabet : k
Lowercase Alphabet : l
Uppercase Alphabet : L
Uppercase Alphabet : M
Lowercase Alphabet : m
```

```
Lowercase Alphabet : n
Uppercase Alphabet : N
Uppercase Alphabet : O
Lowercase Alphabet : o
Lowercase Alphabet : p
Uppercase Alphabet : P
Lowercase Alphabet : q
Lowercase Alphabet : r
Lowercase Alphabet : s
Uppercase Alphabet : S
Uppercase Alphabet : T
Lowercase Alphabet : t
Lowercase Alphabet : u
Uppercase Alphabet : U
Lowercase Alphabet : v
Uppercase Alphabet : V
Uppercase Alphabet : W
Lowercase Alphabet : w
Uppercase Alphabet : X
Lowercase Alphabet : x
Lowercase Alphabet : y
Uppercase Alphabet : Y
Lowercase Alphabet : z
Uppercase Alphabet : Z
```

Question 17: Write a Java program to demonstrate how multiple threads can access a shared counter variable. Show the problem of race condition (without synchronization).

```java
class MyThread17 implements Runnable {
    Thread t;
    static int counter = 0;

    MyThread17(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            counter++;
```

```
            System.out.println(Thread.currentThread().getName() + ": " +
counter);
        }
    }
}

public class Ques17 {
    public static void main(String[] args) {
        MyThread17 t1 = new MyThread17("Thread A");
        MyThread17 t2 = new MyThread17("Thread B");

        t1.t.start();
        t2.t.start();

        try {
            t1.t.join();
            t2.t.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques17.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques17
Thread B: 2
Thread B: 3
Thread A: 2
Thread B: 4
Thread B: 6
Thread B: 7
Thread A: 5
Thread A: 8
Thread A: 9
Thread A: 10
```

Question 18: Write a Java program to demonstrate synchronization by using the synchronized keyword on a method that increments a counter.

```
class MyThread18 implements Runnable {
    Thread t;
    static int counter = 0;

    MyThread18(String tname) {
```

```
        t = new Thread(this, tname);
    }

    public synchronized void incrementCounter() {
        counter++;
        System.out.println(Thread.currentThread().getName() + ": " + counter);
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            incrementCounter();
        }
    }
}

public class Ques18 {
    public static void main(String[] args) {
        MyThread18 t1 = new MyThread18("Thread A");
        MyThread18 t2 = new MyThread18("Thread B");

        t1.t.start();
        t2.t.start();

        try {
            t1.t.join();
            t2.t.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques18.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques18
Thread A: 1
Thread A: 3
Thread B: 2
Thread B: 5
Thread B: 6
Thread A: 4
Thread A: 8
Thread A: 9
Thread B: 7
Thread B: 10
```

Question 19: Write a Java program to create a thread that prints the current time every 2 seconds, five times.

```
import java.time.LocalTime;
import java.time.format.DateTimeFormatter;

class MyThread19 implements Runnable {
    Thread t;

    MyThread19(String tname) {
        t = new Thread(this, tname);
    }

    public void run() {
        DateTimeFormatter d = DateTimeFormatter.ofPattern("HH:mm:ss");
        for (int i = 0; i < 5; i++) {
            LocalTime now = LocalTime.now();
            System.out.println("Current Time: " + now.format(d));
            try {
                Thread.sleep(2000);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

public class Ques19 {
    public static void main(String[] args) {
        MyThread19 t1 = new MyThread19("Thread1");
        t1.t.start();

        try {
            t1.t.join();
            ;
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques19.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques19
Current Time: 09:21:14
Current Time: 09:21:16
Current Time: 09:21:18
Current Time: 09:21:20
Current Time: 09:21:22
```

Question 20: Write a Java program where two threads run in parallel:

- **The first thread prints "Learning Java" 5 times.**
- **The second thread prints "Multithreading in action" 5 times.**

```
class MyThread20 extends Thread {
    String msg;

    MyThread20(String tname, String msg) {
        super(tname);
        this.msg = msg;
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            System.out.println(Thread.currentThread().getName() + ": " + msg);
            try {
                sleep(100);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}

public class Ques20 {
    public static void main(String[] args) {
        MyThread20 t1 = new MyThread20("A", "Learning Java");
        MyThread20 t2 = new MyThread20("B", "Multithreading in action");

        t1.start();
        t2.start();

        try {
            t1.join();
            t2.join();
        } catch (Exception e) {
            System.out.println(e);
```

```
        }
    }
}
```

```
PS D:\CDAC\Java\MultithreadingAssignment> javac Ques20.java
PS D:\CDAC\Java\MultithreadingAssignment> java Ques20
B: Multithreading in action
A: Learning Java
B: Multithreading in action
A: Learning Java
A: Learning Java
B: Multithreading in action
A: Learning Java
B: Multithreading in action
A: Learning Java
B: Multithreading in action
```