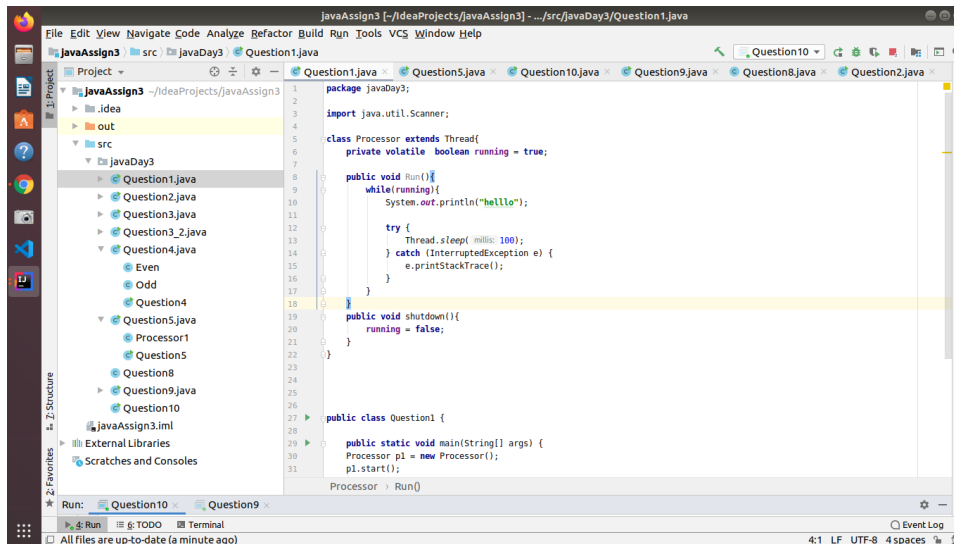


Question: 1:

Write a program to demonstrate the use of volatile keyword.



```
package javaDay3;
import java.util.Scanner;

class Processor extends Thread{
    private volatile boolean running = true;

    public void Run(){
        while(running){
            System.out.println("hello");

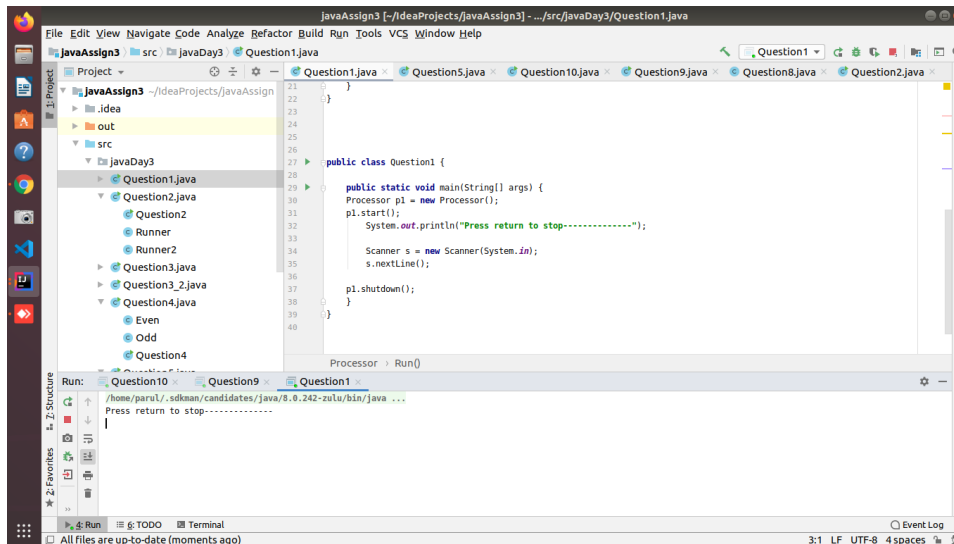
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    public void shutdown(){
        running = false;
    }
}

public class Question1 {

    public static void main(String[] args) {
        Processor p1 = new Processor();
        p1.start();

        Processor p2 = new Processor();
        p2.start();
    }
}
```



```
public class Question1 {

    public static void main(String[] args) {
        Processor p1 = new Processor();
        p1.start();

        System.out.println("Press return to stop.....");

        Scanner s = new Scanner(System.in);
        s.nextLine();

        p1.shutdown();
    }
}

Processor p1 = new Processor();
p1.start();

Processor p2 = new Processor();
p2.start();
```

Question: 2:

Write a program to create a thread using Thread class and Runnable interface each.

```
package javaDay3;

class Runner extends Thread {
    public void run() {
        for (int i = 0; i < 10; i++) {
            System.out.println("HelloWorld " + i);

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

class Runner2 implements Runnable {
    @Override
    public void run() {
        for (int i = 0; i < 10; i++) {
            System.out.println("HelloWorld " + i);

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

Question2 main() {
}
```

Run: Question10 Question9 Question1

Build completed successfully in 1 s 187 ms (4 minutes ago)

```
@Override
public void run() {
    for (int i = 0; i < 10; i++) {
        System.out.println("HelloWorld " + i);

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

public class Question2 {
    public static void main(String[] args) {
        Runner runner1 = new Runner();
        Thread t1 = new Thread(new Runner2());
        runner1.start();
        t1.start();
    }
}
```

Run: Question10 Question9 Question1

Build completed successfully in 1 s 187 ms (4 minutes ago)

```
Run: Question10 Question9 Question2
HelloWorld 0
HelloWorld 1
HelloWorld 1
HelloWorld 2
HelloWorld 2
HelloWorld 3
HelloWorld 3
HelloWorld 4
HelloWorld 4
HelloWorld 5
HelloWorld 5
HelloWorld 6
HelloWorld 6
HelloWorld 7
HelloWorld 7
HelloWorld 8
HelloWorld 8
HelloWorld 9
HelloWorld 9

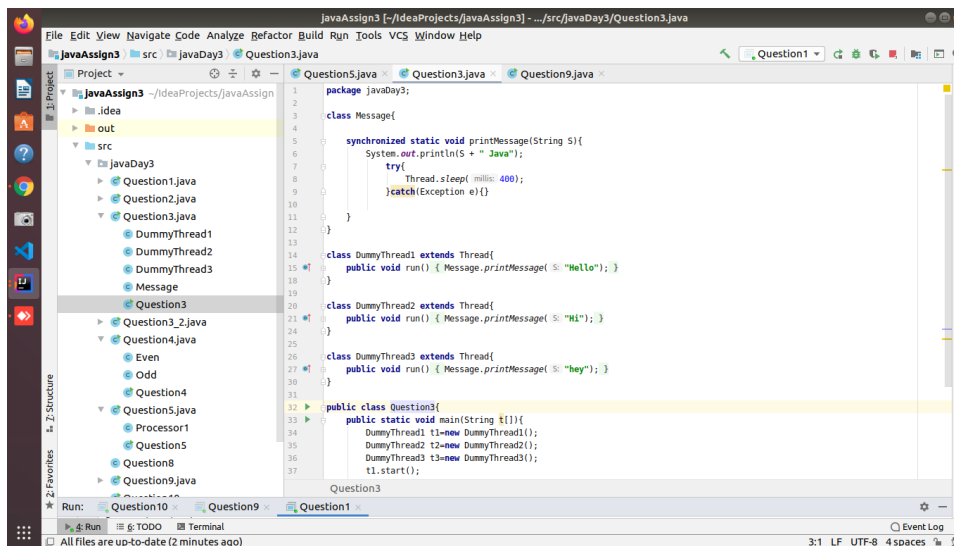
Process finished with exit code 0
```

Run: Question10 Question9 Question2

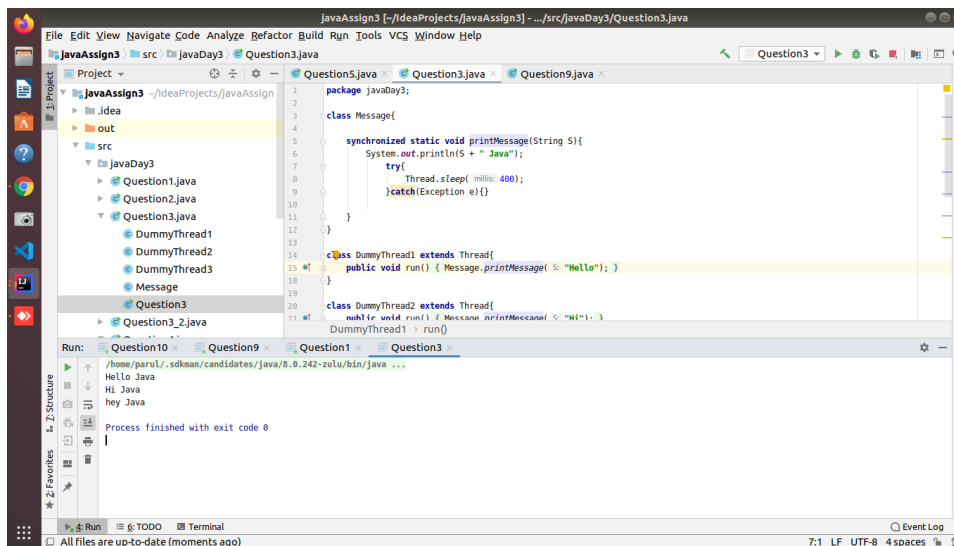
All files are up-to-date (moments ago)

Question: 3:

Write a program using synchronization block and synchronization method.



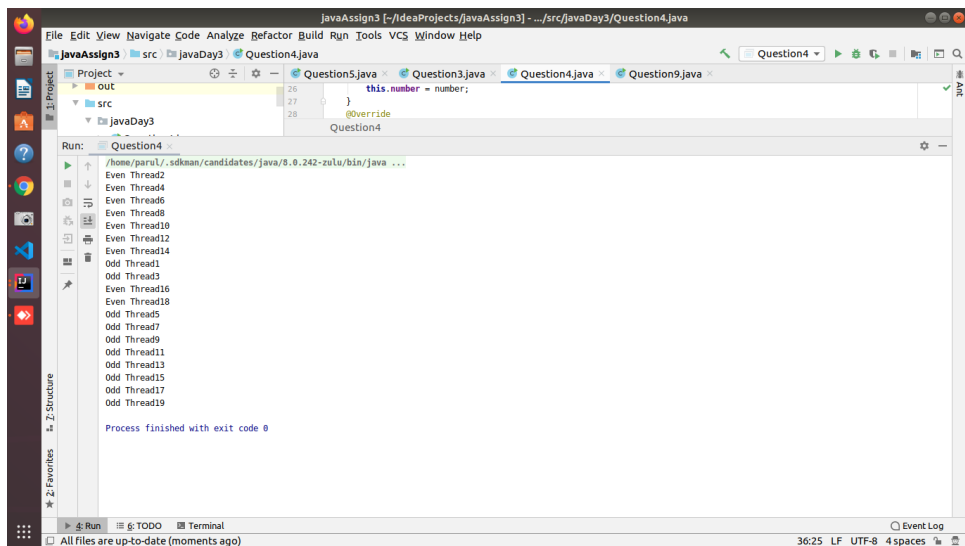
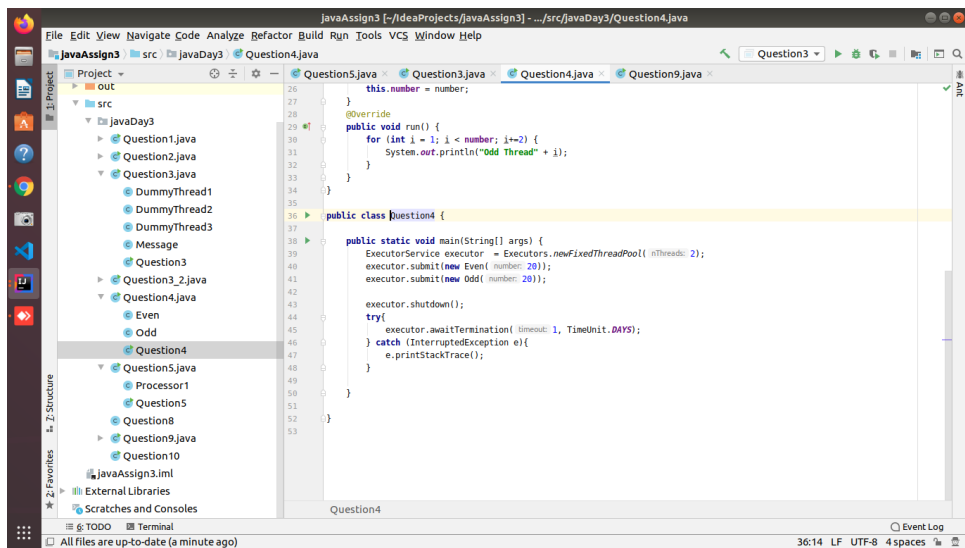
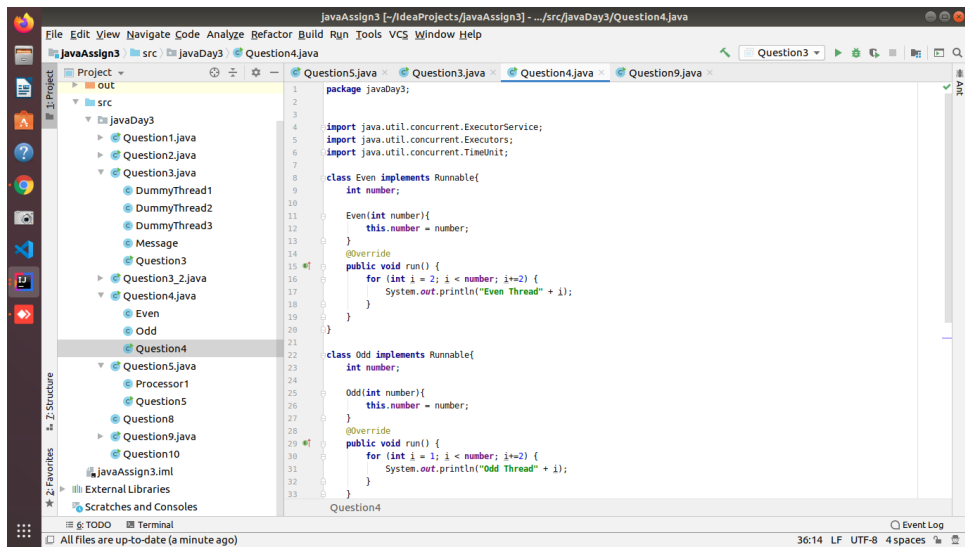
```
1 package javaDay3;
2
3 class Message{
4
5     synchronized static void printMessage(String S){
6         System.out.println(S + " Java");
7         try{
8             Thread.sleep( millis: 400);
9         }catch(Exception e){}
10    }
11
12 }
13
14 class DummyThread1 extends Thread{
15     public void run() { Message.printMessage( S: "Hello"); }
16 }
17
18 class DummyThread2 extends Thread{
19     public void run() { Message.printMessage( S: "Hi"); }
20 }
21
22 class DummyThread3 extends Thread{
23     public void run() { Message.printMessage( S: "hey"); }
24 }
25
26 public class Question3{
27     public static void main(String s[]){
28         DummyThread1 t1=new DummyThread1();
29         DummyThread2 t2=new DummyThread2();
30         DummyThread3 t3=new DummyThread3();
31         t1.start();
32     }
33 }
```



```
Run: /home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Hello Java
Hi Java
hey Java
Process finished with exit code 0
```

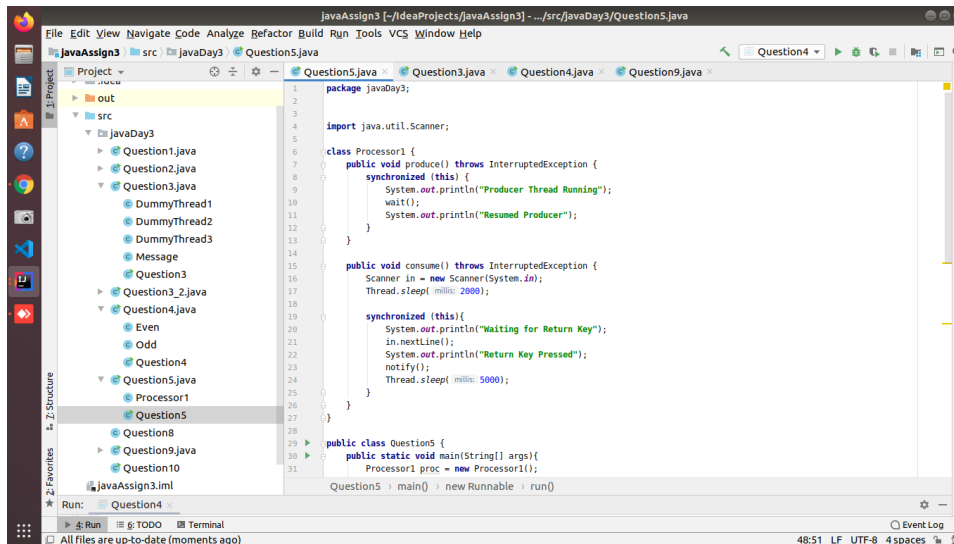
Question: 4:

Write a program to create a Thread pool of 2 threads where one Thread will print even numbers and other will print odd numbers.



Question: 5:

Write a program to demonstrate wait and notify methods.



```
package javaDay3;

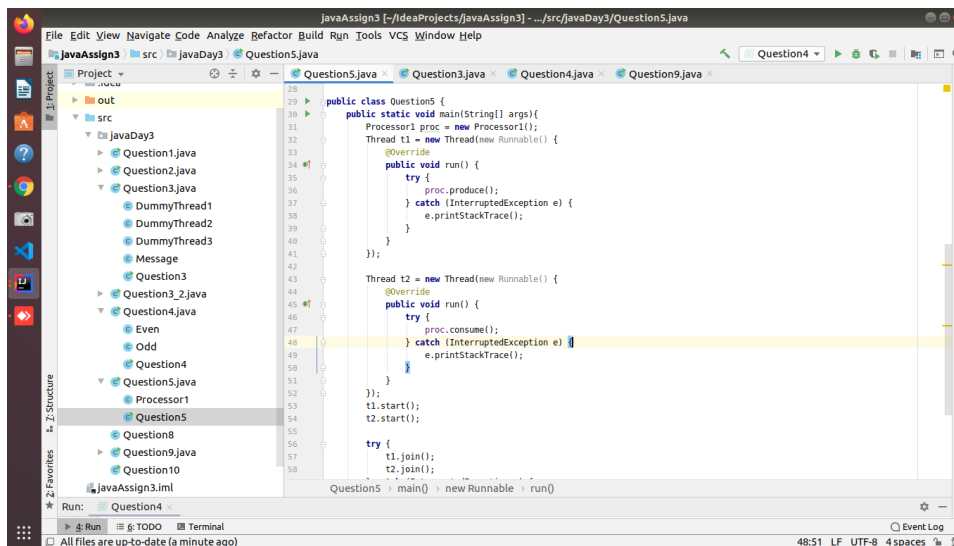
import java.util.Scanner;

class Processor1 {
    public void produce() throws InterruptedException {
        synchronized (this) {
            System.out.println("Producer Thread Running");
            wait();
            System.out.println("Resumed Producer");
        }
    }

    public void consume() throws InterruptedException {
        Scanner in = new Scanner(System.in);
        Thread.sleep(millis: 2000);

        synchronized (this) {
            System.out.println("Waiting for Return Key");
            in.nextLine();
            System.out.println("Return Key Pressed");
            notify();
            Thread.sleep(millis: 5000);
        }
    }
}

public class Question5 {
    public static void main(String[] args) {
        Processor1 proc = new Processor1();
        Question5 main() -> new Runnable -> run()
    }
}
```

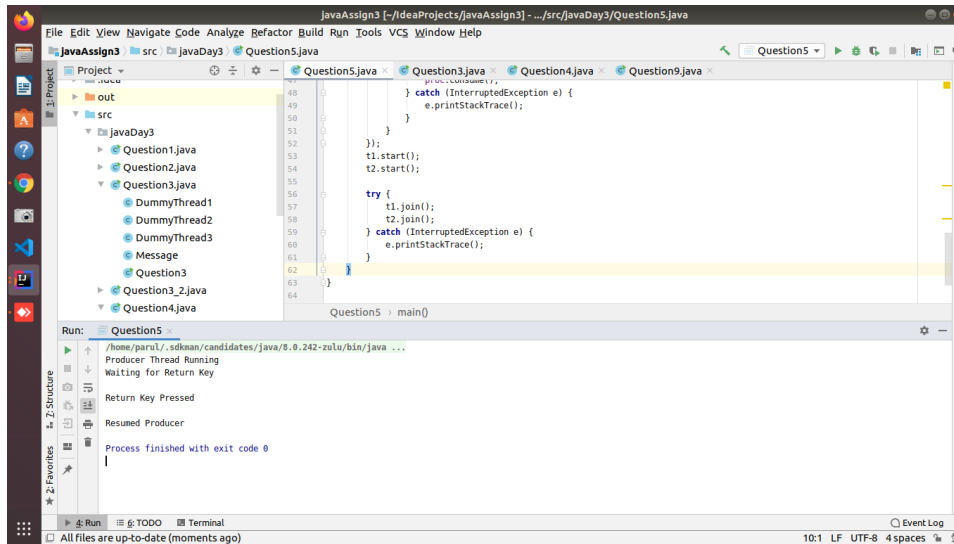


```
public class Question5 {
    public static void main(String[] args) {
        Processor1 proc = new Processor1();
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    proc.produce();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });

        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    proc.consume();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        });

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

        try {
            t1.join();
            t2.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```



Question: 6:

Write a program to demonstrate sleep and join methods.

Question: 7:

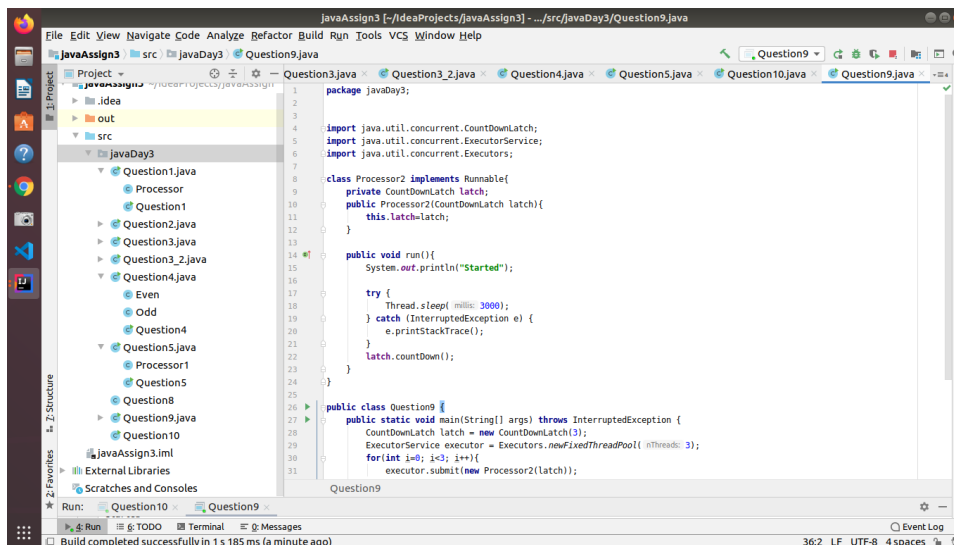
Run a task with the help of callable and store it's result in the Future.

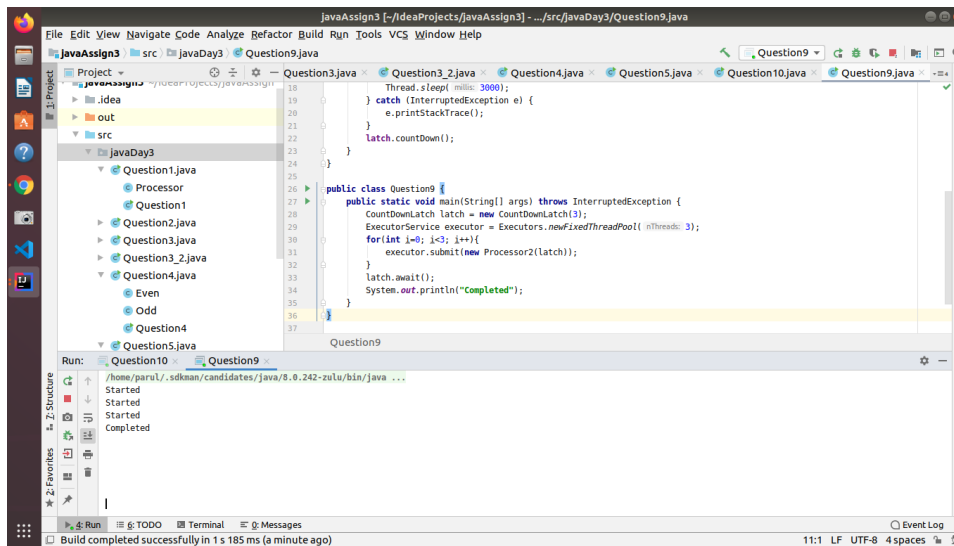
Question: 8:

Write a program to demonstrate the use of semaphore.

Question: 9:

Write a program to demonstrate the use of CountDownLatch.





Question: 10:

Write a program which creates deadlock between 2 threads.

