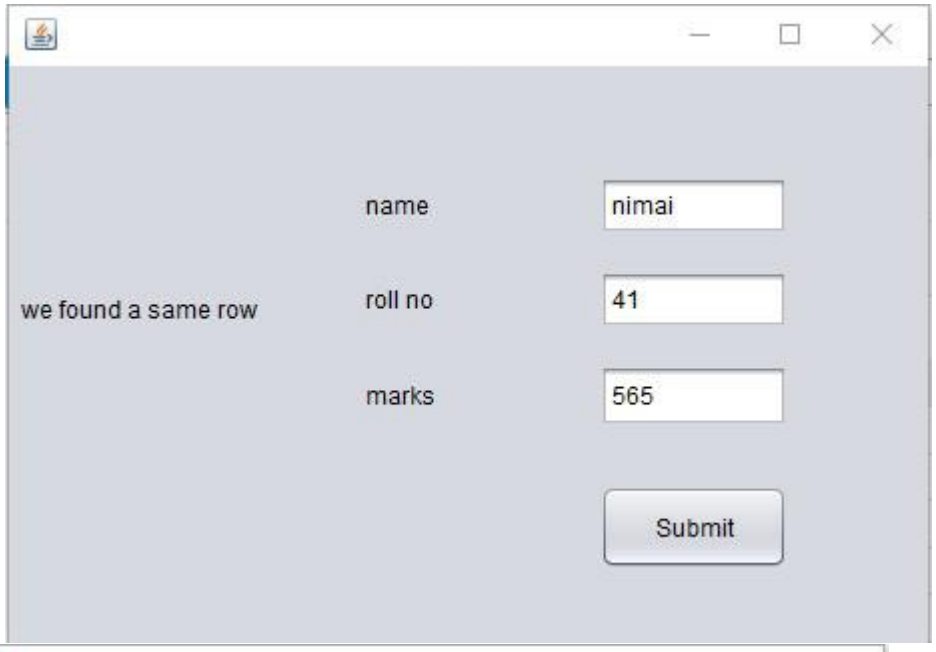
ANS:1

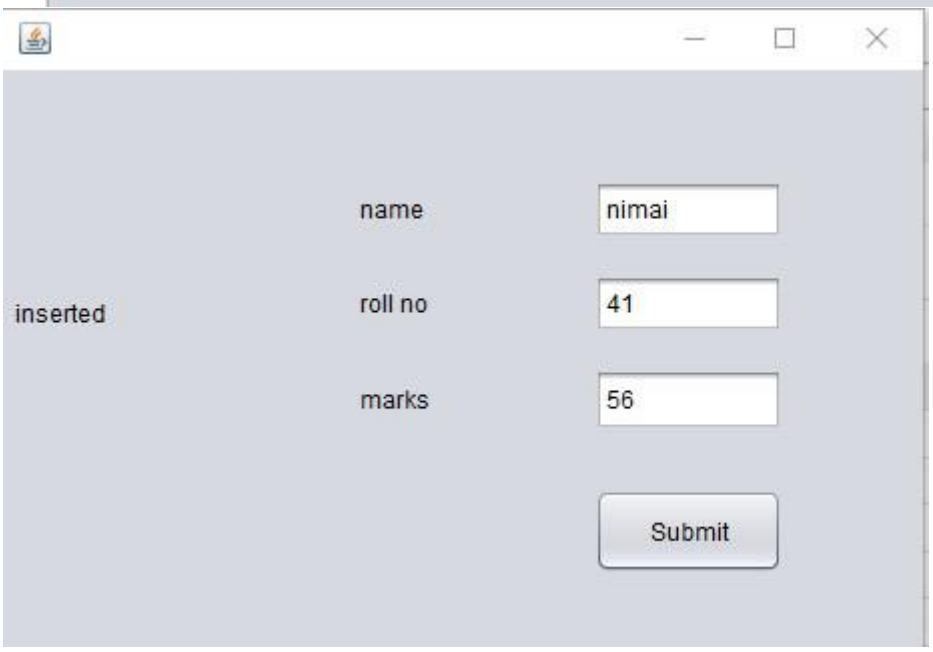
```
 * @author SAPTARSHI */
public class student extends javax.swing.JFrame {

   String name;
   int rollno;
   double marks;
   int ide;

   public student() {
      initComponents();
   }
   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
   private void initComponents() {

      nm = new javax.swing.JTextField();
      roll = new javax.swing.JTextField();
      mrks = new javax.swing.JTextField();
      jLabel1 = new javax.swing.JLabel();
      jLabel2 = new javax.swing.JLabel();
      jLabel3 = new javax.swing.JLabel();
      submit = new javax.swing.JButton();
      status = new javax.swing.JLabel();

      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

      nm.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            nmActionPerformed(evt);
         }
      });

      mrks.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            mrksActionPerformed(evt);
         }
      });

      jLabel1.setText("name");

      jLabel2.setText("roll no");

      jLabel3.setText("marks");

      submit.setText("Submit");
      submit.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            submitActionPerformed(evt);
         }
      });

      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
      getContentPane().setLayout(layout);
      layout.setHorizontalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

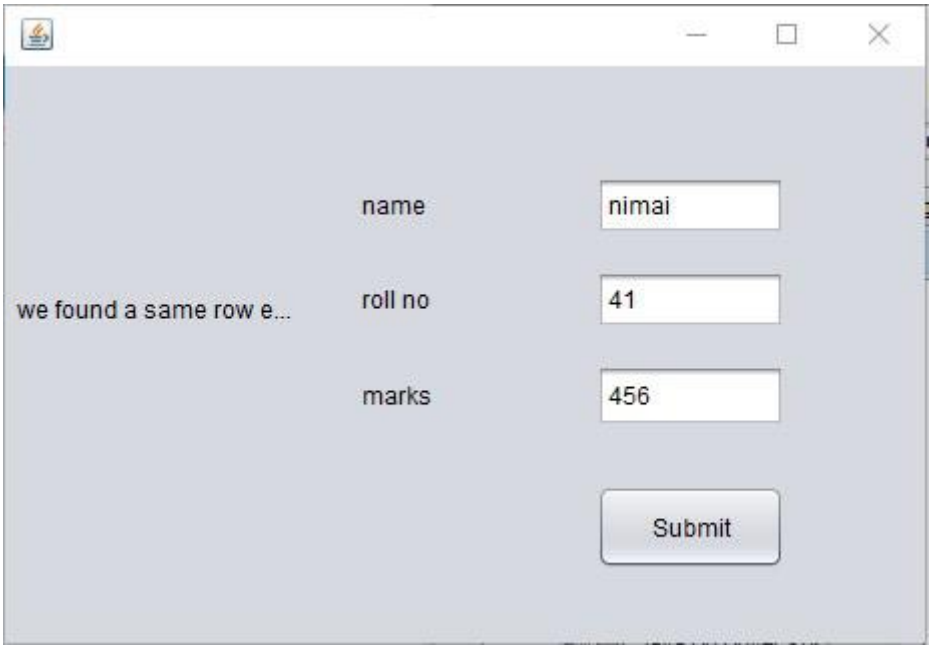
```
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
          .addContainerGap()
          .addComponent(status, javax.swing.GroupLayout.PREFERRED_SIZE, 139,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 33,
Short.MAX_VALUE)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel1)
            .addComponent(jLabel3)
            .addComponent(jLabel2))
          .addGap(82, 82, 82)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(submit, javax.swing.GroupLayout.DEFAULT_SIZE, 94, Short.MAX_VALUE)
            .addComponent(mrks, javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(roll, javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(nm, javax.swing.GroupLayout.Alignment.TRAILING))
          .addGap(70, 70, 70))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(55, 55, 55)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
          .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(nm, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel1))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(roll, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel2))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
              .addComponent(mrks, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel3)))
          .addGroup(layout.createSequentialGroup()
            .addGap(7, 7, 7)
            .addComponent(status, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        .addGap(29, 29, 29)
        .addComponent(submit, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(37, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void nmActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
  }

  private void mrksActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
```

```
    }

    private void submitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        rollno=Integer.parseInt(roll.getText().toString());
        name=nm.getText().toString();
        marks=Integer.parseInt(mrks.getText().toString());
        try{

            Connection
conn=DriverManager.getConnection("jdbc:derby://localhost:1527/Student","root","root");
            status.setText("connecting");
            Statement stmt=conn.createStatement();
            ResultSet rs=stmt.executeQuery("SELECT * from student where rollno="+rollno);

            try{
            if(rs.next()==true)
            {
               throw new Exception("we found a same row exception\n having roll="+rollno);
            }
            else
            {
               int nrow= stmt.executeUpdate("INSERT into student
values('"+name+"',"+rollno+","+marks+")");
               if(nrow!=0)
               {
               status.setText("inserted");
               }

           nm.setText("");
           roll.setText("");
           mrks.setText("");
            }
            }
            catch(Exception e)
            {
                  status.setText(""+e.getMessage());

            }
        }
        catch(Exception e)
        {

        }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
```

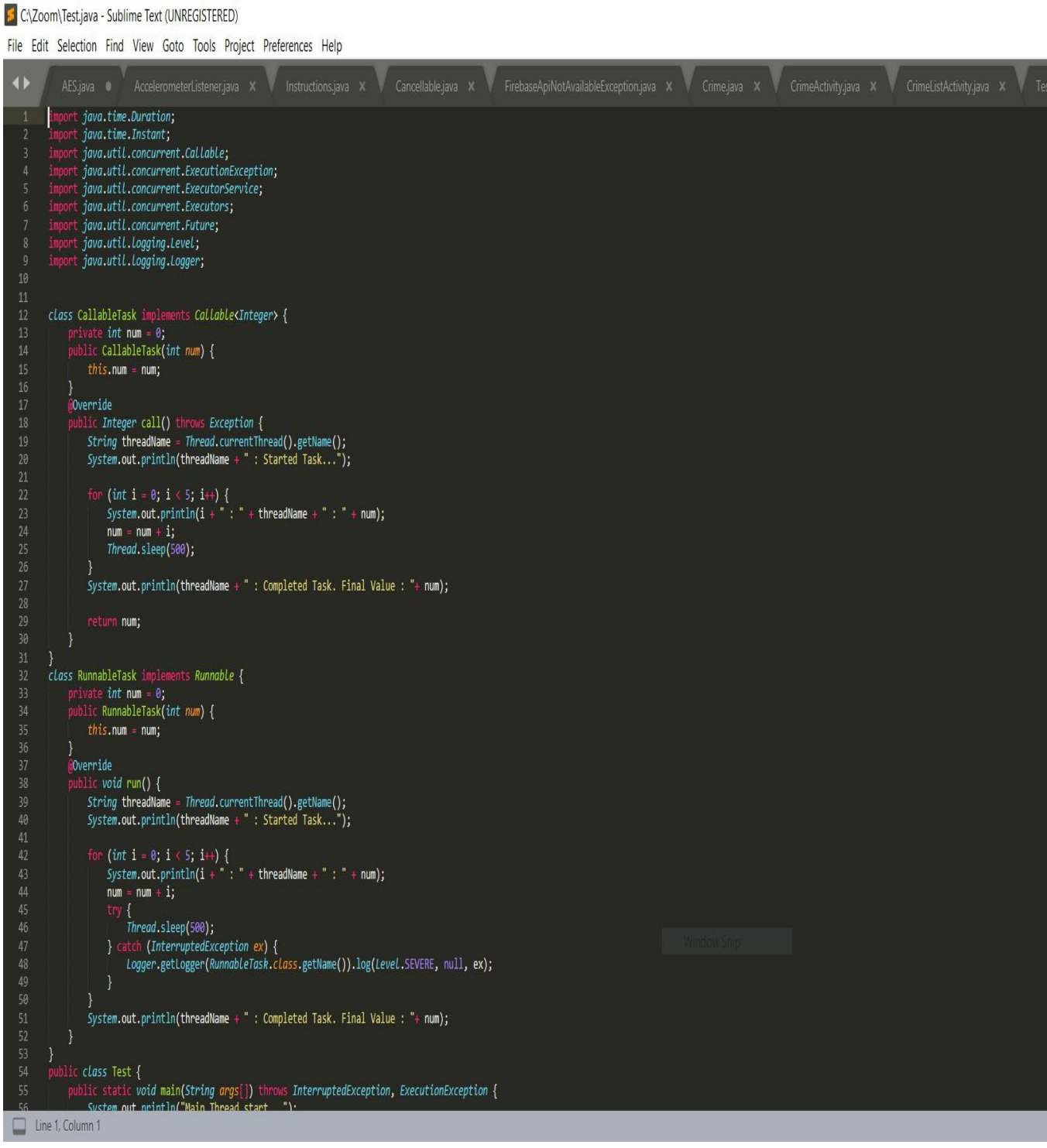
```
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(student.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(student.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(student.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(student.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new student().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JTextField mrks;
private javax.swing.JTextField nm;
private javax.swing.JTextField roll;
private javax.swing.JLabel status;
private javax.swing.JButton submit;
```


name
nimai
we found a same row
roll no
41
marks
565
Submit


name
nimai
inserted
roll no
41
marks
56
Submit




ANS2:


C:\Zoom\Test.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
AES.java
AccelerometerListener.java
Instructions.java
Cancellable.java
FirebaseApiNotAvailableException.java
Crime.java
CrimeActivity.java
CrimeListActivity.java
import java.time.Duration;
import java.time.Instant;
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutionException;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;
import java.util.logging.Level;
import java.util.logging.Logger;
class CallableTask implements Callable<Integer> {
private int num = 0;
public CallableTask(int num) {
this.num = num;
}
@Override
public Integer call() throws Exception {
String threadName = Thread.currentThread().getName();
System.out.println(threadName + " : Started Task...");
for (int i = 0; i < 5; i++) {
System.out.println(i + " : " + threadName + " : " + num);
num = num + i;
Thread.sleep(500);
}
System.out.println(threadName + " : Completed Task. Final Value : "+ num);
return num;
}
}
class RunnableTask implements Runnable {
private int num = 0;
public RunnableTask(int num) {
this.num = num;
}
@Override
public void run() {
String threadName = Thread.currentThread().getName();
System.out.println(threadName + " : Started Task...");
for (int i = 0; i < 5; i++) {
System.out.println(i + " : " + threadName + " : " + num);
num = num + i;
try {
Thread.sleep(500);
} catch (InterruptedException ex) {
Logger.getLogger(RunnableTask.class.getName()).log(Level.SEVERE, null, ex);
}
}
System.out.println(threadName + " : Completed Task. Final Value : "+ num);
}
}
public class Test {
public static void main(String args[]) throws InterruptedException, ExecutionException {
Line 1, Column 1

\

C:\Zoom\Test.java - Sublime Text (UNREGISTERED)

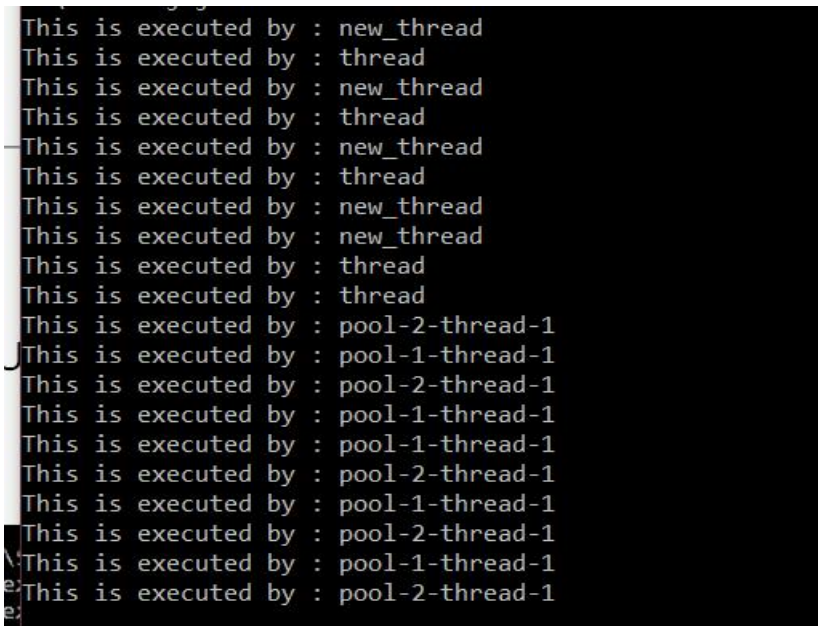
File Edit Selection Find View Goto Tools Project Preferences Help

AES.java • | AccelerometerListener.java × | Instructions.java × | Cancellable.java × | FirebaseApiNotAvailableException.java × | Crime.java ×

```
        String threadName = Thread.currentThread().getName();
        System.out.println(threadName + " : Started Task...");

        for (int i = 0; i < 5; i++) {
            System.out.println(i + " : " + threadName + " : " + num);
            num = num + i;
            try {
                Thread.sleep(500);
            } catch (InterruptedException ex) {
                Logger.getLogger(RunnableTask.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        System.out.println(threadName + " : Completed Task. Final Value : "+ num);
    }
}
public class Test {
    public static void main(String args[]) throws InterruptedException, ExecutionException {
        System.out.println("Main Thread start...");
        Instant start = java.time.Instant.now();

        runnableThreads();
        callableThreads();

        Instant end = java.time.Instant.now();
        Duration between = java.time.Duration.between(start, end);
        System.out.format("Time taken : %02d:%02d.%04d \n", between.toMinutes(), between.getSeconds(), between.toMillis());

        System.out.println("Main Thread completed...");
    }
    public static void runnableThreads() throws InterruptedException, ExecutionException {
        ExecutorService executor = Executors.newFixedThreadPool(4);
        Future<?> f1 = executor.submit( new RunnableTask(5) );
        Future<?> f2 = executor.submit( new RunnableTask(2) );
        Future<?> f3 = executor.submit( new RunnableTask(1) );

        // Waits until pool-thread complete, return null upon successful completion.
        System.out.println("F1 : "+ f1.get());
        System.out.println("F2 : "+ f2.get());
        System.out.println("F3 : "+ f3.get());

        executor.shutdown();
    }
    public static void callableThreads() throws InterruptedException, ExecutionException {
        ExecutorService executor = Executors.newFixedThreadPool(4);
        Future<Integer> f1 = executor.submit( new CallableTask(5) );
        Future<Integer> f2 = executor.submit( new CallableTask(2) );
        Future<Integer> f3 = executor.submit( new CallableTask(1) );

        // Waits until pool-thread complete, returns the result.
        System.out.println("F1 : "+ f1.get());
        System.out.println("F2 : "+ f2.get());
        System.out.println("F3 : "+ f3.get());

        executor.shutdown();
    }
}
```

Line 1, Column 1

OUTPUT:




Q3.

```
 import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import javax.swing.JOptionPane;

class Task implements Runnable
{
	private String name;

	public Task(String s)
	{
		name = s;
	}

	// Prints task name and sleeps for 1s
	// This Whole process is repeated 5 times
	public void run()
	{
		try
		{
			for (int i = 0; i<=5; i++)
			{
				if (i==0)
				{
					Date d = new Date();
```

```
                        SimpleDateFormat ft = new SimpleDateFormat("hh:mm:ss");
                        System.out.println("Initialization Time for"
                                + " task name - "+ name +" = " +ft.format(d));
                        //prints the initialization time for every task
                    }
                    else
                    {
                        Date d = new Date();
                        SimpleDateFormat ft = new SimpleDateFormat("hh:mm:ss");
                        System.out.println("Executing Time for task name - "+
                                name +" = " +ft.format(d));

                    }
                    Thread.sleep(1000);
                }
                System.out.println(name+" complete");
            }

            catch(InterruptedException e)
            {
                e.printStackTrace();
            }
        }
}
public class Java_ca
{
    static  int MAX_T = 10;

    public static void main(String args[])
    {
      Scanner sc=new Scanner(System.in);
                System.out.println("Enter Number of Task: ");
      MAX_T=sc.nextInt();
      //Runnable task[] = null;
      System.out.println("Total Number of task: "+MAX_T);

        //ExecutorService pool = Executors.newFixedThreadPool(MAX_T);
        ScheduledExecutorService pool = Executors.newScheduledThreadPool(MAX_T);
        for(int i=0;i<MAX_T;i++)
        {
          Task task=new Task("Task No: "+i);
          pool.execute(task);
        }


          pool.shutdown();
    }
}
```