**6) Write a program to illustrate Event Handling (keyboard events)?**

import *java.awt.\**;

import *java.awt.event.\**;

*class* q21 *implements* *KeyListener*

{

    Label l;

*public* void perfom()

    {

        Frame f = new Frame("Event Handling");

        f.setSize(400, 400);

        f.setVisible(true);

        f.setLayout(null);

        l = new Label("I am Label");

        l.setBounds(10, 50, 100, 20);

        TextField t1 = new TextField();

        t1.setBounds(20,80,300,300);

        f.add(l);

        f.add(t1);

        t1.addKeyListener(this);

    }

*public* void keyPressed(KeyEvent e)

    {

        l.setText("Key Pressed");

    }

*public* void keyReleased(KeyEvent e)

    {

        l.setText("Key Released");

    }

*public* void keyTyped(KeyEvent e)

    {

        l.setText("Key Typed");

    }

}

*class* Main

{

*public* *static* void main(String[] args)

    {

        q21 ob = new q21();

        ob.perfom();

    }

}

**OUTPUT:  
A screen shot of a computer screen

Description automatically generated with medium confidence**

**7)Write a program to illustrate Inter Thread Communication?**

*class* q18

{

  int amount=10000;

*synchronized* void withdraw(int amount)

  {

    System.out.println("going to withdraw...");

    if(this.amount<amount)

    {

      System.out.println("Less balance; waiting for deposit...");

      try

      {

        wait();

      }

      catch(Exception e)

      {

        System.out.println(e);

      }

    }

    this.amount-=amount;

    System.out.println("withdraw completed...");

    }

*synchronized* void deposit(int amount)

    {

      System.out.println("going to deposit...");

      this.amount+=amount;

      System.out.println("deposit completed... ");

      notify();

    }

}

*class* Main

{

*public* *static* void main(String args[])

  {

    q18 ob=new q18();

    new Thread()

    {

*public* void run()

      {

        ob.withdraw(15000);

      }

    }.start();

    new Thread(){

*public* void run()

      {

        ob.deposit(10000);

      }

    }.start();

  }

}

**OUTPUT:**

going to withdraw...

Less balance; waiting for deposit...

going to deposit...

deposit completed...

withdraw completed..

**8)Write a program to illustrate Thread Synchronization?**

*class* q17

{

*synchronized* void print(int n)

    {

        for(int i=1;i<=5;i++)

        {

            System.out.println(n\*i);

            try

            {

                Thread.sleep(400);

            }

            catch(Exception e)

            {

                System.out.println(e);

            }

        }

    }

}

*class* Main

{

*public* *static* void main(String args[])

    {

        q17 obj = new q17();

        new Thread()

        {

*public* void run()

            {

                obj.print(100);

            }

        }.start();

        new Thread()

        {

*public* void run()

            {

                obj.print(5);

            }

        }.start();

    }

}

**OUTPUT:**

100

200

300

400

500

5

10

15

20

25

**9) Write a program to illustrate Multithreading?**

*class* t1 *extends* Thread

{

*public* void run()

    {

        int t=10;

        while(t>0)

        {

            System.out.println("Thread 1");

            t--;

        }

    }

}

*class* t2 *extends* Thread

{

*public* void run()

    {

        int t=10;

        while(t>0)

        {

            System.out.println("Thread 2");

            t--;

        }

    }

}

*class* Main

{

*public* *static* void main(String[] args)

    {

        t1 ob1 = new t1();

        t2 ob2 = new t2();

        ob1.start();

        ob2.start();

    }

}

**OUTPUT:**

Thread 1

Thread 1

Thread 1

Thread 1

Thread 1

Thread 1

Thread 2

Thread 2

Thread 1

Thread 1

Thread 2

Thread 2

Thread 2

Thread 2

Thread 2

Thread 2

Thread 1

Thread 1

Thread 2

Thread 2

**10)WAP on HashSet,TreeSet,LinkedList?**

import *java.util.\**;

*class* Main

{

*public* *static* void main(String[] args)

    {

*Set* ob1 = new HashSet();

        ob1.add("subbu");

        ob1.add("09");

        ob1.add("OG");

        System.out.println(ob1);

*Set* ob2 = new TreeSet();

        ob2.add(9);

        ob2.add(13);

        ob2.add(47);

        System.out.println(ob2);

*List* ob3 = new LinkedList();

        ob3.add("dady");

        ob3.add("mummy");

        ob3.add("begum");

        System.out.println(ob3);

    }

}

**OUTPUT:**

[subbu, OG, 09]

[9, 13, 47]

[dady, mummy, begum]