Advance Programming

BCA V Semester

Lab Report

Submitted By: Shiba Ratna Tamrakar

(Sample Lab sheet)

Kantipur City College

Content

CONTENT..2
LAB SHEET 1..3
LAB SHEET 2..7
LAB SHEET 3..13
LAB SHEET 4..16
LAB SHEET 5..21
LAB SHEET 6..25
LAB SHEET 7..29
LAB SHEET 8..33
LAB SHEET 9..38
LAB SHEET 10..41

Lab Sheet 1

1. Write a program that will display your name, rollnum, sex, and faculty.

Background

This question is to demonstrate how a java code is written

Expected output

```
C:\jdk1.3\bin\java.exe q1
Name:	Shiba Ratna Tamrakar

Roll:	4001ca03

Sex:	Male

Faculty:BCA

Application Exit...
```

Source code

```
import java.io.*;
class q1
{
	public static void main(String args[])
	{
		System.out.println("Name:\t Shiba Ratna Tamrakar");
		System.out.println("Roll:\t 4001ca03");
		System.out.println("Sex:\tMale");
		System.out.println("Faculty:\t BCA");
	}
}
```

---

2. Write a program that will generate fibonacci series

Background

This Question is to demonstrate the use of for loop in Java. It also gives the logic to solve the Fibonacci series

Algorithm

```
• initialize three veriables say p=0, q=1, i=0 and sum=0
• print values of p and q
• loop until the falue of i=10
      sum=p+1
      print the value of sum
      assign p=q and q=sum
• end loop
```

Expected output

```
C:\jdk1.3\bin\java.exe  q2
011 2 3 5 8 13 21 34 55 Application Exit...
```

Source code

```
import java.io.*;
class q2
{
	public static void main(String args[])
	{
```

```
                int sum=0,p=0,q=1;
                for(int i=0;i<10;i++)
                {
                        sum=p+q;
                        System.out.println("Faculty:\t BCA");
                        p=q;
                        q=sum;
                }
        }
}
```

3. a) Run the following program and observe the output

Background

This is to demonstrate the use of for loop

Algorithm

1. set i=0
2. if i<1000 then print “Hello we are having lab” else loop to step 1
3. end

Expected output

```
        1000 times
                Hello we are having lab
```

Source code

```
import java.io.*;
class q3a
{
        public static void main(String args[])
        {
                for(int i=0;i<1000;++i)
                {
                        System.out.println("Hello we are having lab");
                }
        }
}
```

3. b. Run the following program and observes the output

Background

This is to demonstrate the use of for loop, if semi column is written at the back of the for loop then the loop does not include any line next to it.

Algorithm

4. set i=0
5. if i<1000 then else loop to step 1
6. print “ Hello we are having lab”
7. end

Expected output

```
C:\jdk1.3\bin\java.exe  q3b
Hello we are having lab

Application Exit...
```

Source code

```
import java.io.*;
class q3b
{
   public static void main(String args[])
   {
          for(int i=0;i<1000;++i);
          {
             System.out.println("Hello we are having lab");
          }
       }
    }
```

4. Create two dimensional integer arrays and which can store 16 different integer numbers and display the output by increasing the value of each element by your roll number.

Background

- Array is secondary data type which is capable to handle large numbers of variable.

Expected output

```
C:\jdk1.3\bin\java.exe  q4
32 32 32 32 33 33 33 33 34 34 34 34 35 35 35 35  Application Exit...
```

Source code

```
import java.io.*;
class q4
{
   public static void main(String args[ ])
   {
          int a[ ][ ];
          a=new int[4][4];
          for(int i=0;i<4;i++)
     {
               for(int j=0;j<4;j++)
               {
                    a[i][j]=i;
               }
     }
          for(int i=0;i<4;i++)
     {
               for(int j=0;j<4;j++)
               {
                    a[i][j]=a[i][j]+32;
               }
     }
          for(int i=0;i<4;i++)
     {
               for(int j=0;j<4;j++)
               {
                    System.out.print(a[i][j]+"  ");
               }
             }
     }
}
```

5. a

Background

These examples demonstrate that character also has equivalent integer value (i. e. ASCII code). So we can add or subtract any number to work with it.

Expected output

```
C:\jdk1.3\bin\java.exe  q5a
The value of y is ={
Application Exit...
```

Source code

```
import java.io.*;
class q5a
{
        public static void main(String args[])
        {
          char x,y;
          x='z';
          y=++x;
          System.out.println("The value of y is ="+y);

        }
}
```

5. b

Background

These examples demonstrate that character also has equivalent integer value (i. e. ASCII code). So we can add or subtract any number to work with it.

Expected output

```
C:\jdk1.3\bin\java.exe q5b
The value of a,b,c,d is=4 4 16 17
Application Exit......
```

Source code

```
import java.io.*;
class q5b
{
   public static void main(String args[])
   {
          int a,b,c,d;
          a=4;
          b=4;
          c=a*b;
          d=c+1;
          System.out.println("The value of a,b,c,d is="+a+" "+b+" "+c+" "+d);


   }
}
```

Lab Sheet 2

1. **Declare any two methods and some property in super class (You can name as per your convention) and add any new method in sub class and finally use the objects of subclass to access the property and methods of super class.**

Background

This Question is to demonstrate the use of Inheritance. **Inheritance** is the feature of object oriented programming which enhances the reusability of the code. Properties of One class can be used by its sub classes. The base class which has been inherited is called super class in Java and the inheriting class is called subclass

Expected output

```
C:\jdk1.3\bin\java.exe  q1
The output of SupObj.iProp is 1
The output of SupObj.fProp is 2.2
The output of SubObj.iProp is 10
The output of SubObj.fProp is 22.2
1st Function of a superclass
2nd Function of a superclass
1st Function of a superclass
2nd Function of a superclass
Function of a Sub-class
Application Exit...
```

Source Code

```
import java.io.*;
class Super
{
        int iProp;
        double fProp;
        public void SuperFx1()
        {
                System.out.println("1st Function of a superclass");
        }
        public void SuperFx2()

        {
                System.out.println("2nd Function of a superclass");
        }

}
class SubClass extends Super
{
        SubClass()
        {
                super();
        }
        public void SubFx()
        {
                System.out.println("Function of a Sub-class");
        }
}


class q1
{
        public static void main(String args[])
        {
```

```
                        //Creating an object of Super class
                        Super SupObj=new Super();

                        //Creating an object of Sub class
                        SubClass SubObj=new SubClass();

                        SupObj.iProp=1;
                        SupObj.fProp=2.2;

                        //Output of the properties of Superclass by object of Super class
                        System.out.println("The output of SupObj.iProp is "+SupObj.iProp);
                        System.out.println("The output of SupObj.fProp is "+SupObj.fProp);

                        SubObj.iProp=10;
                        SubObj.fProp=22.2;
                        //Output of the properties of Superclass by object of Sub class
                        System.out.println("The output of SubObj.iProp is "+SubObj.iProp);
                        System.out.println("The output of SubObj.fProp is "+SubObj.fProp);

                        //Calling the SuperFx1() method of Super class using object of super class
                        SupObj.SuperFx1();
                        //Calling the SuperFx2() method of Super class  using object of super class
                        SupObj.SuperFx2();
                        //Calling the SuperFx1() method of Super class  using object of sub class
                        SubObj.SuperFx1();
                        //Calling the SuperFx2() method of Super class  using object of sub class

                        SubObj.SuperFx2();
                        //Calling the SubFx() using object of sub class
                        SubObj.SubFx();
                }
        }
```

2. **Create an interface class with few definitions of ant methods and create two other classes that implement these methods with different ways.**

Background

The **interface** keyword takes the abstract concept.We could think of it as a “pure” abstract class. It allows the creator to establish the form for a class: method names, argument lists and return types, but no method bodies

E.g.

```
        interface Name
        {
                public final double PI=3.1416;
                void back( );
        }
```

Another class which inherits the interface class uses the method of interface class in their own way.

Expected output

```
C:\jdk1.3\bin\java.exe  q2
The valu is2
The value is4
Application Exit...
```

Source Code

```
import java.io.*;
```

```
interface callback
{
        void callback(int p);
}

class client implements callback
{
        public void callback(int p)
        {
                System.out.println("The valu is" +p);
        }
}

class client2 implements callback
{
        public void callback(int p)
        {
                p*=2;
                System.out.println("The value is"+p);
        }
}
class q2
{
        public static void main(String args[])
        {
                client c1obj=new client();
                client2 c2obj=new client2();
                c1obj.callback(2);
                c2obj.callback(2);
        }
}
```

3. Declare method methodOfSuperClass( ) in super class which print your name and using the same method declaration in subclass print My Roll No is:.... What this property is called?

Background

If both base and inheriting class have methods of same name then it is called overriding. The output of the object created with the sub class is the output of the method of sub class.

Expected output

```
C:\jdk1.3\bin\java.exe  q3
My Name is Shiba Ratna Tamrakar
My Roll no. is 4001ca03
Application Exit...
```

Source Code

```
import java.io.*;
class Super
{
        void methodOfSuperClass()
        {
                System.out.println("My Name is Shiba Ratna Tamrakar");
        }
}
class SubClass extends Super
```

```
{
        SubClass()
        {
                super();
        }
        void methodOfSuperClass()
        {
                System.out.println("My Roll no. is 4001ca03");
        }
}


class q3
{
        public static void main(String args[])
        {
                //Creating an object of Super class
                Super SupObj=new Super();

                //Creating an object of Sub class
                SubClass SubObj=new SubClass();


                //Calling the method of Super class using object of super class
                SupObj.methodOfSuperClass();

                //Calling the method of Super class  using object of sub class
                SubObj.methodOfSuperClass();
        }
}
```

4. Create a package My Package where you can put few related class and use this package in mani class and use the some methods and property of class.

Background

Java allows us to group files into packages. To indicate that a file is part of a package, all we need to do is include the package declaration

Package packageName;

As the very first line in file. This file should be placed into the folder with same name of package.

Expected output

```
C:\jdk1.3\bin\java.exe  q4
My name is Shiba
My ID is 4001 ca03
Application Exit...
```

Source Code

```
                                //File MyDemo1 a part of package MyPackage
  package MyPackage;
  import java.io.*;
  public class MyDemo1
  {
        public void MyId()
        {
        System.out.println("My ID is 4001 ca03");
        }
  }

                                  //File MyDemo a part of package
package MyPackage;
```

```
import java.io.*;
public class MyDemo
{
   public void Name()
   {
   System.out.println("My name is Shiba");
   }
}
```

```
        //Importing my defined package
        import MyPackage.*;
        class q4
        {
                public static void main(String args[])
                {
                        //creating boject of class MyDemo1 and MyDemo
                        MyDemo1 dem1=new MyDemo1();
                        MyDemo dem=new MyDemo();
                        //calling function in file MyDemo
                        dem.Name();
                        //calling function in file MyDemo1
                        dem1.MyId();
                }
        }
```

5. **Write a program with out and with handling mathematical exception. What can you conclude from these two programs?**

Background

Exception is any error which occurs in run time. It is the error which is occurs due to wrong input of the user and the program will terminate. Suppose if you enter the filename which don't exist to open then exception is occurred.

Expected output

```
C:\jdk1.3\bin\java.exe  q5
The error type is  :::::java.lang.ArithmeticException: / by zero
Application Exit...
```

Source Code

```
        /*
        import java.io.*;

        class q5
        {
                public static void main(String args[])
                {
                        int a;
                        a=10/0;
                        System.out.println(a);
                }
        }
        */
        import java.io.*;

        class q5
        {
                public static void main(String args[])
                {
                        try
                        {
                                int a;
                a=10/0;
                System.out.println(a);
            }catch(Exception e)
            {
                System.out.println("The error type is  :::::" + e);
            }
        }
    }
```

Lab Sheet 3

Q 1. Write a program that will display the random number and print number is greater than 100 if generated number is greater than 100.

Source

```
import java.io.*;
import java.util.Random;
class Question1
{
	public static void main(String args[])
	{
		Random Rnum=new Random();
		int Number;
		Number=Rnum.nextInt(500);
		if (Number>100)
		{
			System.out.print(Number+" Is Greater then 100.");
		}
		else
			System.out.print(Number +" Is Smaller then 100.");
	}
}
```

Output

```
		125 Is Greater then 100.
```

Next time

```
		26 Is Smaller then 100.
```

Q 2. Write a program that will display name, roll no, age, gender and faculty at the interval of 1000 ms.

Source

```
import java.io.*;
class NewThread implements Runnable
{
	Thread T;
	NewThread()
	{
		T=new Thread(this,"me");
		System.out.println("Thread Started.");
		T.start();
	}
	public void run()
	{
		try
		{
		System.out.println("Shiba Ratna Tamrakar.");
		T.sleep(1000);
		System.out.println("4001 CA03.");
		T.sleep(1000);
		System.out.println("20 Years.");
		T.sleep(1000);
		System.out.println("Male.");
		T.sleep(1000);
		System.out.println("Science and Technology.");
		}catch(InterruptedException E)
		{
			System.out.println("Exception");
```

```
                    }
            }
    }
    class Question2
    {
                    public static void main(String args[])
                    {
                            new NewThread();
                    }
    }
```

Output

```
    Shiba Ratna Tamrakar.
    4031 CA03.
    20 Years.
    Male.
    Science and Technology.
```

Q 3. Write a program using Vector that can store different data objects where you can allocate the space at run time (Note that you must declare initial size 5 and at run time run must store 10 elements.)

source

```
    import java.io.*;
    import java.util.*;
    class Question3
    {
            public static void main(String args[])
            {
                    Vector V = new Vector(5,2);
                    V.addElement(new Integer(1000));
                    V.addElement(new String("BCA"));
                    V.addElement(new Double(2000.89));
                    V.addElement(new String("Shiba"));
                    V.addElement(new Integer(3000));
                    V.addElement(new String("Baskota "));
                    V.addElement(new Integer(4000));
                    V.addElement(new String("Kantipur"));
                    V.addElement(new Float(5000.99));
                    V.addElement(new String("Kirtipur"));
                    V.addElement(new Double(9999.99));
                    System.out.println("Contents of the Vector : ");
                    System.out.println(V);
            }
    }
```

Output

```
    Contents of the Vector :
    [1000, BCA, 2000.89, Shiba, 3000, Baskota, 4000, Kantipur, 5000.99, Kirtipur, 9999.99]
```

Vector is one dimensional dynamic Array. Vector provides the capabilities of array-like data structures that can dynamically resize themselves.

Q 4. Write a program will display 4 students name, roll num, and gender for 3 times. [Hint: Multithreading]

Source

```
    import java.io.*;
    class PrintThread extends Thread
```

```
{
        public PrintThread(String Name)
        {
                System.out.println(Name);
        }
        public void run()
        {
                try
                {
                System.out.println("Shiba Ratna Tamrakar.");
                Thread.sleep(1000);
                System.out.println("4001 CA03.");
                Thread.sleep(2000);
                System.out.println("Male.");
                }catch(InterruptedException E)
                {
                        System.out.println("Exception");
                }
        }
}
class Question4
{
        public static void main(String args[])
        {
                PrintThread T,U,V,W;
                T=new PrintThread("1.");
                U=new PrintThread("2.");
                V=new PrintThread("3.");
                W=new PrintThread("4.");
                T.start();
                U.start();
                V.start();
                W.start();
                }
}
```

Output:

```
1.
2.
3.
4.
Shiba Ratna Tamrakar.
Shiba Ratna Tamrakar.
Shiba Ratna Tamrakar.
Shiba Ratna Tamrakar.
4001 CA03.
4001 CA03.
4001 CA03.
4001 CA03.
Male.
Male.
Male.
Male.
```

Lab Sheet 4

Q 1. Create an applet that print your name, roll no, age, sex, hobby, diagonally.

Source

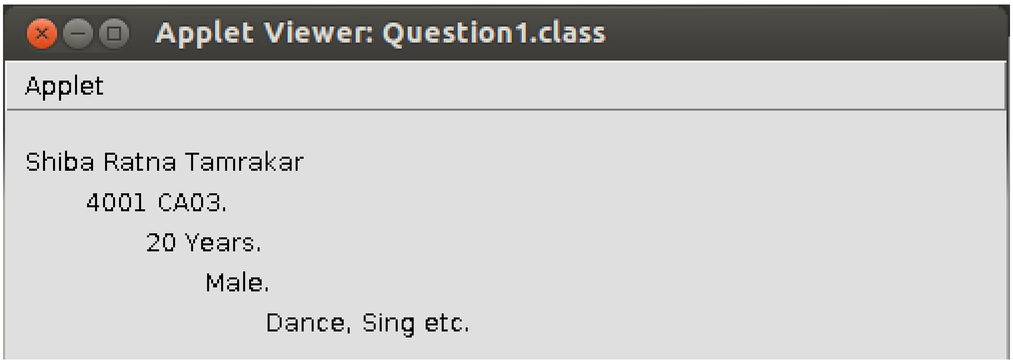
Java File

```
import java.awt.*;
import java.applet.*;
public class Question1 extends Applet
{
        public void paint(Graphics G)
        {
                G.drawString("Shiba Ratna Tamrakar ",10,30);
                G.drawString("4001 CA03.",40,50);
                G.drawString("20 Years.",70,70);
                G.drawString("Male.",100,90);
                G.drawString("Dance, Sing etc.",130,110);
        }
}
```

HTML File

```
<html>
        <head>
                <title>Question 1</title>
        </head>
<body>
        <applet code="Question1" width="220" height="120"></applet>
</body>
</head>
```

Output:



```
Shiba Ratna Tamrakar
     4001 CA03.
          20 Years.
               Male.
                    Dance, Sing etc.
```

Q 2. Write a program for creating Digital Clock.

Source

Java File

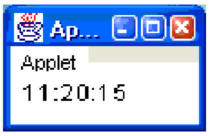
```
import java.awt.*;
import java.applet.*;
import java.util.*;
public class Question2 extends Applet implements Runnable
{
        Thread T;
        GregorianCalendar Show;
        public void init()
        {
                T=new Thread(this,"Clock");
```

```
				Show=new GregorianCalendar();
			}
			public void paint(Graphics G)
			{
				int h=Show.get(Calendar.HOUR);
				int m=Show.get(Calendar.MINUTE);
				int s=Show.get(Calendar.SECOND);
				G.drawString(""+h,2,10);
				G.drawString(" : "+m,20,10);
				G.drawString(" : "+s,45,10);
			}
			public void start()
			{
				T.start();
			}
			public void run()
			{
				for(;;)
				{
					Show.setTime(new Date());
					repaint();
					try
					{
						T.sleep(1000);
					}catch(InterruptedException e)
					{
					}
				}
			}
		}
```

HTML File

```
		<html>
			<head>
				<title>Question 2</title>
			</head>
		<body>
			<applet code="Question2" width="70" height="20"></applet>
		</body>
		</head>
```

Output:



```
11:20:15
```

Q 3. **Write a program that will create a banner that rotates linearly.**
Source

Java File

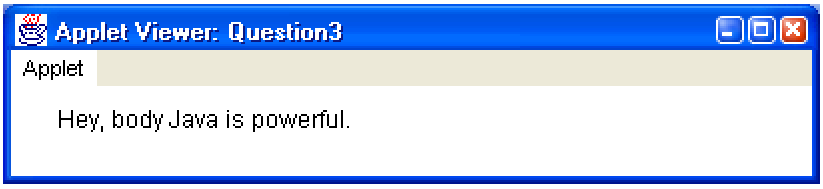
```
import java.awt.*;
import java.applet.*;
public class Question3 extends Applet implements Runnable
{
		static int i=0;
		Thread T;
		public void init()
		{
			T=new Thread(this,"me");
		}
```

```
        public void paint(Graphics G)
        {
                i++;
                if(i>=400)
                        i=0;
                G.drawString("Hey, body Java is powerful.",i,20);

        }
        public void start()
        {
                T.start();
        }
        public void run()
        {
                        for(;;)
                        repaint();
        }
}
```

HTML File

```
        <html>
                <head>
                        <title>Question 3</title>
                </head>
        <body>
                <applet code="Question3" width="400" height="40"></applet>
        </body>
        </head>
```

Output:



```
Hey, body Java is powerful.
```

Q 4. Write a program that pass your semester course information to applet and print that information on the applet.

Source

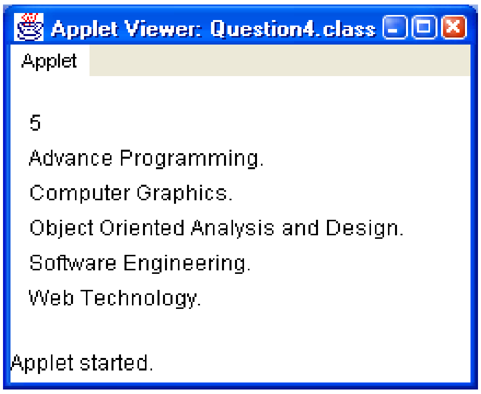
Java File

```
import java.awt.*;
import java.applet.*;
public class Question4 extends Applet
{
        String Semester,Sub1, Sub2, Sub3, Sub4, Sub5;
        public void init()
        {
                Semester=getParameter("Semester");
                Sub1=getParameter("Sub1");
                Sub2=getParameter("Sub2");
                Sub3=getParameter("Sub3");
                Sub4=getParameter("Sub4");
                Sub5=getParameter("Sub5");
        }
        public void paint(Graphics G)
        {
                G.drawString(Semester,10,30);
                G.drawString(Sub1,10,50);
                G.drawString(Sub2,10,70);
                G.drawString(Sub3,10,90);
                G.drawString(Sub4,10,110);
                G.drawString(Sub5,10,130);
        }
}
```

HTML File

```
        <html>
                <head>
                        <title>Question 4</title>
                </head>
        <body>
                <applet code="Question4" width="250" height="150">
                        <param name= Semester value=5>
                        <param name= Sub1 value="Advance Programming. ">
                        <param name= Sub2 value="Computer Graphics.">
                        <param name= Sub3 value="Object Oriented Analysis and Design.">
                        <param name= Sub4 value="Software Engineering.">
                        <param name= Sub5 value="Web Technology.">
                </applet>
        </body>
        </head>
        </html>
```

Output



```
5
Advance Programming.
Computer Graphics.
Object Oriented Analysis and Design.
Software Engineering.
Web Technology.
```

```
Applet started.
```

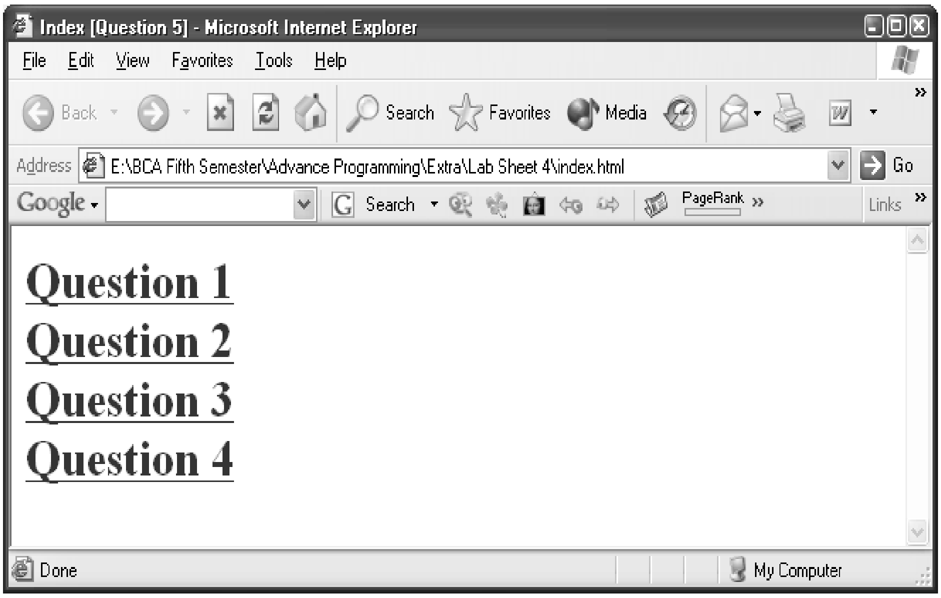
Q 5. Create index.html and provide the link to all above html files by keeping few class file in other directory.

Source

HTML File

```
        <html>
                <head>
                        <title>Index [Question 5]</title>
                </head>
        <body>
        <form></form>
        <h1>
                <a href="Question1.html">Question 1</a><br>
                <a href="Question2.html">Question 2</a><br>
                <a href="Question3.html">Question 3</a><br>
                <a href="Question4.html">Question 4</a></div>
        </h1>
        </body>
        </head>
```

Output


Index [Question 5] - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Google
Search
PageRank
Links
Go
Done
My Computer

Address E:\BCA Fifth Semester\Advance Programming\Extra\Lab Sheet 4\index.html

Question 1
Question 2
Question 3
Question 4

Lab Sheet 5

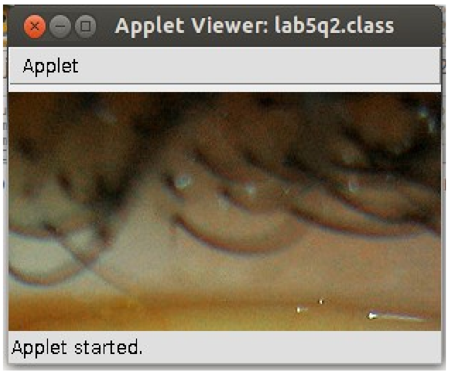
1. Write a program that will let the use to input text.

```
import javax.swing.*;
import java.awt.*;
/*
<applet code=lab5q1 width="300" height="50">
</applet>
*/
public class lab5q1 extends JApplet// implements ActionListener
{
         JTextField jtf;
        public void init()
        {
                Container Contentpane=getContentPane();
                JLabel j=new JLabel("Name:");
                Contentpane.add(j);
                jtf=new JTextField(20);
                Contentpane.setLayout(new FlowLayout());
                Contentpane.add(jtf);
        }
}
```

2.

```
import javax.swing.*;
import java.awt.*;
/*
<applet code=lab5q2 width="800" height="600">
</applet>
*/
public class lab5q2 extends JApplet
{
        public void init()
        {
                Container content=getContentPane();
                content.setLayout(new FlowLayout());

                JLabel Name=new JLabel(new ImageIcon("bik.jpg"));
                content.add(Name);
        }
}
```

Output:



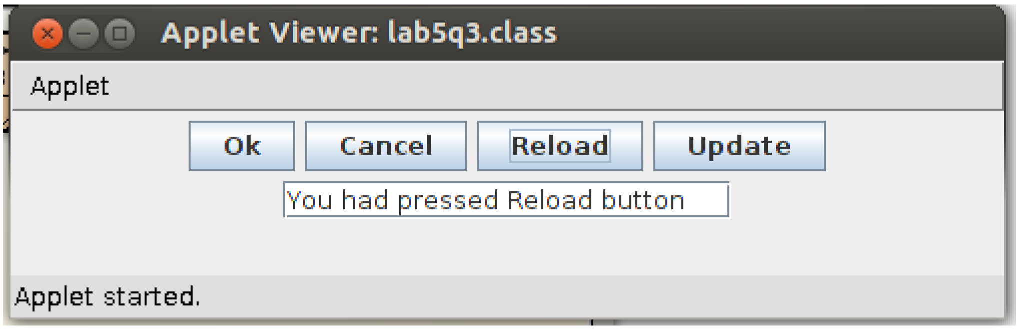
Applet started.

3. Swing program to display which Button is pressed in a TextField.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
/*
<applet code=lab5q3 width="400" height="100">
</applet>
*/
public class lab5q3 extends JApplet implements ActionListener
{
        JTextField Msg;
        public void init()
        {
                Container content=getContentPane();
                content.setLayout(new FlowLayout());

                JButton Ok=new JButton("Ok");
                Ok.setActionCommand("Ok");
                Ok.addActionListener(this);
                content.add(Ok);

                JButton Cancel=new JButton("Cancel");
                Cancel.setActionCommand("Cancel");
                Cancel.addActionListener(this);
                content.add(Cancel);

                JButton Reload=new JButton("Reload");
                Reload.setActionCommand("Reload");
                Reload.addActionListener(this);
                content.add(Reload);

                JButton Update=new JButton("Update");
                Update.setActionCommand("Update");
                Update.addActionListener(this);
                content.add(Update);

                Msg=new JTextField(20);
                content.add(Msg);


        }
        public void actionPerformed(ActionEvent a)
        {
                if (a.getActionCommand()=="Ok")
                {
                        Msg.setText("You had pressed Ok button");
                }

                else if (a.getActionCommand()=="Cancel")
                {
                        Msg.setText("You had pressed Cancel button");
                }
                else if (a.getActionCommand()=="Reload")
                {
                        Msg.setText("You had pressed Reload button");
                }
                else
                {
                        Msg.setText("You had pressed Update button");
```

```
                    }
          }
}
```

Output:



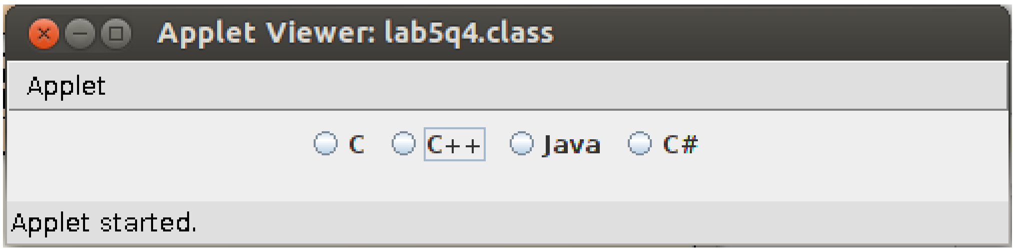
You had pressed Reload button

Applet started.

4. Program to demonstrate Option button.

```
import javax.swing.*;
import java.awt.*;
/*
<applet code=lab5q4 width="80" height="30">
</applet>
*/
public class lab5q4 extends JApplet
{
          public void init()
          {
                    Container content=getContentPane();
                    content.setLayout(new FlowLayout());
                    JRadioButton c=new JRadioButton("C");
                    content.add(c);
                    JRadioButton c1=new JRadioButton("C++");
                    content.add(c1);
                    JRadioButton java=new JRadioButton("Java");
                    content.add(java);
                    JRadioButton c4=new JRadioButton("C#");
                    content.add(c4);


          }
}
```

Output:

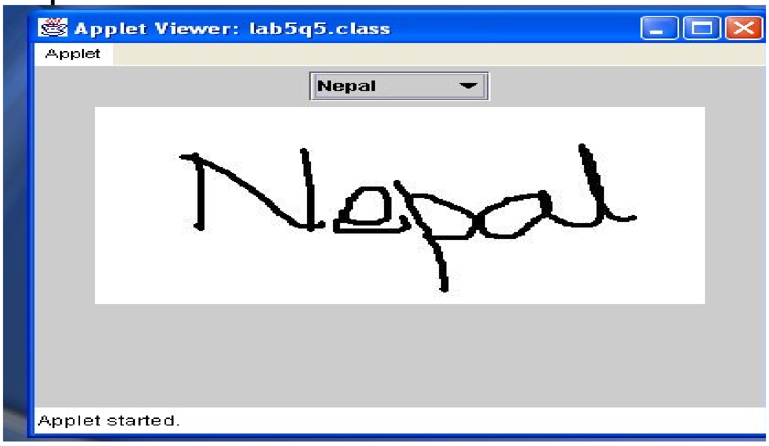


Applet started.

5

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
/*
<applet code=lab5q5 width="400" height="250">
</applet>
*/
public class lab5q5 extends JApplet implements ItemListener
{
        JLabel def;
        public void init()
        {
                Container content=getContentPane();
                content.setLayout(new FlowLayout());

                JComboBox country=new JComboBox();
                country.addItem("Nepal");
                country.addItem("India");
                country.addItem("Pakistan");
                country.addItem("Maldivs");
                country.addItem("Bangladesh");
                country.addItem("Bhutan");
                country.addItem("Srilanka");
                country.addItem("China");
                content.add(country);
                def=new JLabel(new ImageIcon("Nepal.jpg"));
                content.add(def);
        }
        public void itemStateChanged(ItemEvent Item)
        {
                String s=(String)Item.getItem();
                def.setIcon(new ImageIcon("Sunset.jpg"));
        }
}
```

Output:



Lab Sheet 6

1. **Write a program that will draw following 2D diagrams on the applet window**
   a. **Rectangle** b. **Line** c. **Filled ellipse** d. **Filled Rectangle**
   e. **With the different color for each and also for few provide the system.**

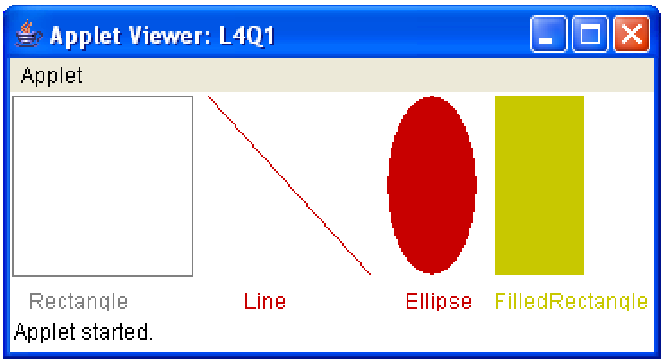
Background

The Graphics class is the member of java.awt library. It is a most important class used to do work in the applets as well as Frame. **DrawLine( )** method draws a line from the point (x1,y1) to point(x2,y2). the general

DrawRect( ) and **fillRect( )** methods respectively draw and fill rectangles.

DrawOval( ) and **fillOvel( )** methods respectively draw and fill ovals. The oval is inside the imaginary rectangle.

Output



Applet started.

Source Code

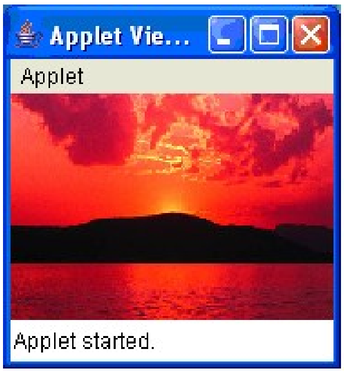
```
/* <applet code="L6Q1" height="100" width="100"></applet>*/
import java.awt.*;
import java.applet.*;
public class L6Q1 extends Applet
{
        public void paint(Graphics g)
        {

                Color dark=new Color(200,0,0); //create a color
                g.setColor(Color.gray); //sets foreground color from current session
                g.drawString("Rectangle",10,120);
                g.drawRect(1,1,100,100);                //Draw rectangle
                g.setColor(dark);
                g.drawLine(110,1,200,100);      //Draw line
                g.drawString("Line",130,120);
                g.fillOval(210,1,50,100);               //draw filled ellipse
                g.drawString("Ellipse",220,120);
                dark=new Color(200,200,0);
                g.setColor(dark);
                g.drawString("FilledRectangle",270,120);                //Draw filled
rectangle
                g.fillRect(270,1,50,100);
        }
}
```

2. Write a program that will load an image in an applet window.

Background

With **getImage( )** and **drawImage( ),** it is actually quite easy to load and display an image. The **getImage( )** method is use to load the image in the memory and **drawImage( )** method is used to draw the image in the applet.

Output



Source code

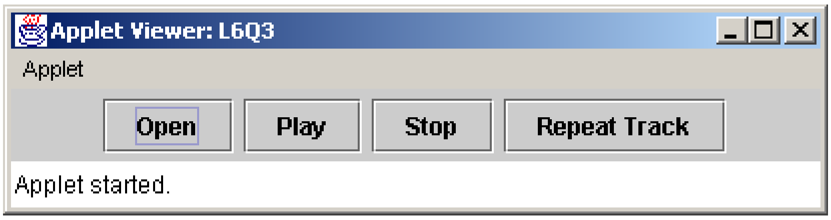
```
/* <applet code="L6Q2" height="100" width="100"></applet>*/
import java.awt.*;
import java.applet.*;

public class L6Q2 extends Applet
{
                    Image img;
                    public void init()
                    {
                     img=getImage(getDocumentBase(),"pic.jpg");
                    }
                     public void paint(Graphics g)
                    {
                              g.drawImage(img,0,0,this);
                    }
}
```

3. Write a program that will load audio files and play them randomly for each time when use presses the play button. You need to create different button for each operations.

Background

The AudioClip interface defines these methods **play()** to paly auto clip, **stop()** to stop playing clip and **loop()** to play the loop continuously. After we have loaded an audio clip using **getAudioClip( )**, we can use these methods to play it.

Output



Source Code

```
import java.awt.*;
import java.applet.*;
import javax.swing.*;
import java.awt.event.*;
/*<applet code="L6Q3" width="500" height="300"></applet>*/
public class L6Q3 extends JApplet implements ActionListener
{
                    AudioClip ac[]=new AudioClip[5];
                    double number;
                    int i,j;
                    int index;
                    public void init()
                    {
                     Container contentpane=getContentPane();
                     contentpane.setLayout(new FlowLayout());
                     JButton btnOpen=new JButton("Open");
                     btnOpen.setActionCommand("open");
                     btnOpen.addActionListener(this);
                     contentpane.add(btnOpen);

                     JButton btnPlay=new JButton("Play");
                     btnPlay.setActionCommand("play");
                     btnPlay.addActionListener(this);
                     contentpane.add(btnPlay);

                     JButton btnStop=new JButton("Stop");
                     btnStop.setActionCommand("stop");
                     btnStop.addActionListener(this);
                     contentpane.add(btnStop);

                     JButton btnRepeat=new JButton("Repeat Track");
                     btnRepeat.setActionCommand("repeat");
                     btnRepeat.addActionListener(this);
                     contentpane.add(btnRepeat);
                    }
                    public void actionPerformed(ActionEvent ae)
                    {
                     if(ae.getActionCommand()=="open")
                     {
                            for(j=1;j<=5;j++)
                            {
                                   ac[j-1]=getAudioClip(getDocumentBase(),j+".wav");
                            }
                     }
                     else if(ae.getActionCommand()=="play")
                     {
                            number=Math.random()*10;
                            index=(int)number;
                            if(index>4)
                            {
                            }
                            else
                            {
                                   ac[index].play();
                            }
                     }
                     else if(ae.getActionCommand()=="stop")
                     {
                            ac[index].stop();
                     }
                     else if(ae.getActionCommand()=="repeat")
                     {
                            ac[index].loop();
                    }
                    }
    }
```

Lab Sheet 7

1. WAP that will find the cube of age and square of your roll number. (Take input from user).

Background

This program is based on the input operation, i.e. the user interface in the program. BufferReader class is used to read characters from the consol. Java uses stream base IO operation. Here all the operations are operated via classes and objects..

Output

```
Enter the Age :25
Enter the Rollno :32
Cube of Age is: 15625
Square of Roll is: 1024
```

Source Code

```
import java.io.*;

public class L7Q1
{
	public static void main(String args[]) throws IOException
	{
		String str1, str2;
		int age;
		int roll;
		BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
		System.out.print("Enter the Age :");
		str1=br.readLine();
		System.out.print("Enter the Rollno :");
		str2=br.readLine();
		System.out.print("\n");
		age=Integer.parseInt(str1);
		roll=Integer.parseInt(str2);
		age=age*age*age;
		roll=roll*roll;
		System.out.println("Cube of Age is: "+age);
		System.out.println("Square of Roll is: "+roll);
	}
}
```

2. WAP to read the content of file and display.

Background

This program is base on file read operations. FileReader class is used to read content of the file.

Constructors for FileReader class

```
	FileReader(String filepath)
	FileReader(File Fileobject)
```

Output

```
Enter the Filename with extension L7Q2.java
```

The output will be the source

Source

```
import java.io.*;

public class L7Q2
{
	public static void main(String args[]) throws IOException
	{
		String str;
		int age;
```

```
                    FileReader fr;
                    System.out.print("Enter the Filename with extension :");
                    BufferedReader br=new BufferedReader(new
                    InputStreamReader(System.in));
                    str=br.readLine();
                    System.out.print("\n");
                    fr=new FileReader(str);
                    br=new BufferedReader(fr);
                    while ((str=br.readLine())!=null)
                    System.out.println(str);
            }
    }
```

3. WAP that will read the content of one file and write that content to new file.

Background

This program is base on file read/write operations. FileReader class is used to read content of the file. And FileWriter class is used to write content in a file.

Constructors for FileReader class

FileReader(String filepath)
FileReader(File Fileobject)
FileWriter(String filepath)
FileWriter(File Fileobject)
FileWriter(File Fileobject,Boolean append)

Source

```
    import java.io.*;

    public class L7Q3
    {
            public static void main(String args[]) throws IOException
            {
                    String strOldFile,strNewFile,str;
                    FileReader fr;
                    FileWriter fw;

                    System.out.print("Enter the Old Filename with extension :");
                    BufferedReader br=new BufferedReader(new
    InputStreamReader(System.in));
                    strOldFile=br.readLine();
                    System.out.print("\n");

                    //To input name by user
                    System.out.print("Enter the New Filename with extension :");
                    br=new BufferedReader(new InputStreamReader(System.in));
                    strNewFile=br.readLine();
                    System.out.print("\n");

                    fr=new FileReader(strOldFile);
                    fw=new FileWriter(strNewFile);
                    br=new BufferedReader(fr);
                    String strBuf="";
                    char buffer[];
                    //read the content of line
                    while ((str=br.readLine())!=null)
                    {
                            buffer=new char[str.length()+1];
                            str=str+"\r";
                            str.getChars(0,str.length(),buffer,0);
                            fw.write(buffer);
                    }
                    fr.close();
```

```
                        fw.close();

                }
        }
```

4. Write a Program that will display the properties of a file

Output

```
        File Name: L7Q1.java
        File Path: .\a\L7Q1.java
        Absulate path: C:\Documents and Settings\Shiba\Desktop\lab7\.\a\L7Q1.java
        Parent : .\a
        Existence: true
        Is File: true
        Is Directory: false
        Read/Write: true
        Read only: true
        Is absolute: false
        Is Hidden: false
        File Size: 821 Bytes
        Is Modified on : 1136709140437
```

Source Code

```
        import java.io.*;

        public class L7Q4
        {
                static void property(String s)
                {
                        System.out.println(s);
                }
                public static void main(String arg[])
                {
                        File f=new File(".\\a\\L7Q1.java");
                        property("File Name: " +f.getName());
                        property("File Path: " +f.getPath());
                        property("Absulate path: " +f.getAbsolutePath());
                        property("Parent : " +f.getParent());
                        property("Existence: " +f.exists());
                        property("Is File: " +f.isFile());
                        property("Is Directory: " +f.isDirectory());
                        property("Read/Write: " +f.canWrite());
                        property("Read only: " +f.canRead());
                        property("Is absolute: " +f.isAbsolute());
                        property("Is Hidden: " +f.isHidden());
                        property("File Size: " +f.length() +" Bytes");
                        property("Is Modified on : " +f.lastModified());
                }
        }
```

5. WAP that will list content of directories.

Background

This program will demonstrate how the methods of File class can be use to list the content of directory.

File(Path)

Output

```
        The content of directory lab7 is :  a a.txt demo.class demo.java
```

Source

```
        import java.io.*;
```

```
public class L7Q5
{

        public static void main(String arg[])
        {
                File f=new File("..\\lab7");
                if (f.isDirectory())
                {
                        System.out.println("The content of directory lab7 is : ");
                        String s[]=f.list();
                        for(int i=0;i<s.length;i++)
                                System.out.println(s[i]);
                }
        }
}
```

Lab Sheet 8

1. Write a program that will randomly write in a file

Background

The RandomAccessFile enables us to read and write bytes, text and java data types to any location in a file. **Java.io.RandomAccessFile class** provides the ability to read and write data from or write any specified location in a file. This class implements DataInput and DataOutput interfaces.

- **void seek(long position)**, it sets the file pointer to a particular location inside the file.
- **length()** method returns the length of the file in terms of bytes.
- **Long getFilePointer()** returns the current byte location of the file pointer.
- **int skipBytes(n)** jumps the pointer by n bytes from its current location.

Source Code

```
import java.io.*;
class L8Q1
{
	public static void main(String args[ ]) throws IOException
	{
		try
		{
			RandomAccessFile r1=new RandomAccessFile("in.dat","rw");
			r1.writeShort(10);
			r1.writeInt(20000);
			r1.writeDouble(342.234);
			r1.writeChar('a');
			r1.writeBoolean(true);
			r1.writeLong(545234325);
			r1.writeFloat((float)42342.343);
			r1.seek(0);
			System.out.println(r1.readShort());
			System.out.println(r1.readInt());
			System.out.println(r1.readDouble());
			System.out.println(r1.readChar());
			System.out.println(r1.readBoolean());
			System.out.println(r1.readLong());
			System.out.println(r1.readFloat());
			r1.close();
		}
		catch(FileNotFoundException e)
		{
			System.out.println(e.getMessage());
		}
	}
}
```

2. Write a program that write five objects and read it at a later time

Background

Serialization is the process of writing the state of an object to a byte stream. This is useful when we want to save the state of out program to a persistent storage area, such as a file. At a later time, we may restore these objects by using the process of deserialization.

Output

```
obj1s=hi;i=-7;d=2.0
obj2s=hi;i=-7;d=2.0
```

Source Code

```
import java.io.*;
```

```
public class L8Q2
{
        public static void main(String arg[])
        {
                try
                {
                        X obj1=new X("hi",-7,2.0);
                        X obj2;
                        System.out.println("obj1"+obj1);
                        FileOutputStream fos=new FileOutputStream("serial");
                        ObjectOutputStream oos=new ObjectOutputStream(fos);
                        oos.writeObject(obj1);
                        oos.flush();
                        oos.close();
                        FileInputStream fis=new FileInputStream("serial");
                        ObjectInputStream ois=new ObjectInputStream(fis);
                        obj2=(X)ois.readObject();
                        ois.close();
                        System.out.println("obj2"+obj2);
                }
                catch(Exception e)
                {
                        System.out.println("the error is : "+e);
                }
        }
}

class X implements Serializable
{
        String s;
        int i;
        double d;

        public X(String s,int i,double d)
        {
                this.s=s;
                this.i=i;
                this.d=d;
        }
        public String toString()
        {
                return "s="+s+";i="+i+";d="+d;
        }
}
```

3. Write a program that will connect to the database and perform the following

  a. **Insert the new row**
  b. **Update the row**

c. **Delete the row**
d. **List the record**

Background

JDBC (Java Database Connectivity) is defined, as a set of java classes and methods to interface with database. It also provides uniform access to a wide range of relational databases

Source Code

```
import java.io.*;
import java.sql.*;
public class jdbc_opr
{
    static Connection c;
    public static void main(String args[]) throws Exception
    {

        int ch;

                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        InputStreamReader i=new InputStreamReader(System.in);
        BufferedReader b=new BufferedReader(i);
        try
        {

            c=DriverManager.getConnection("jdbc:odbc:mydsn");
            do
            {
                System.out.println("\t\t\tMenu");
                System.out.println("\t\t\t1. View");
                System.out.println("\t\t\t2. Clear");
                System.out.println("\t\t\t3. Insert");
                System.out.println("\t\t\t4. Modify");
                System.out.println("\t\t\t5. Delete");
                System.out.println("\t\t\t6.exit");
                System.out.println("\t\t\tEnter Your Choice:");

                ch=Integer.parseInt(b.readLine());
                switch (ch)
                {
                case 1:
                {
                    display();
                    break;
                }
                case 2:
                {
                    System.out.flush();
                    break;
                }
                case 3:
                {
                    insert();
                    display();
                    break;
                }
                case 4:
                {
                    modify();
                    display();
                    break;
                }
```

```
                    case 5:
                    {
                            delete();
                            display();
                            break;

                    }
                    case 6:
                    {
                            System.out.println("Thank you");
                            System.exit(0);
                    }
                    default:
                        System.out.println("Invalid Choice");
                }
            }while(ch!=6);
        }
        catch(Exception e){ }
    }
    static void display()
    {
        try
        {
            ResultSet r;
            int row=0;
            Statement st=c.createStatement();
            r=st.executeQuery("select * from emp");
            System.out.println("Id\tName");

            while (r.next())
                System.out.println(r.getInt(1)+"\t"+r.getString(2));
            r.close();
        }
        catch(Exception e){ }
        }
        static void clrscr()
        {
            for(int k=1;k<25;k++)
                System.out.println();
        }

        static void insert()
        {
            try
            {
                int row;
                Statement ins_st;
                ins_st=c.createStatement();
                System.out.println("4001, Shiba");
                row=ins_st.executeUpdate("insert into emp values('4001',' Shiba')");
                c.commit();
                System.out.println("No of Rows inserted="+row);
            }
            catch(SQLException e)
            {
                    System.out.println("Error in insert"+e);
                }
        }
        static void modify()
        {
            try
            {
                int row;
                Statement modi_st;
                modi_st=c.createStatement();
                row=modi_st.executeUpdate("update emp set name='bb' where name='Shiba'");
                c.commit();
                System.out.println("No of Rows updated="+row);
            }catch(SQLException e)
            {
            }
        }
        static void delete()
        {
            try
            {
                System.out.println("4001,bb");
                int row;
                Statement del_st;
                del_st=c.createStatement();
                row=del_st.executeUpdate("delete from emp where id=4001");
                if (row>0)
                {
                    c.commit();
                }
                else
                {
                    System.out.println("Record not found");
                }
            }
            catch(SQLException e)
            {
                System.out.println("Error in delete" +e);
            }
        }
    }
```

Lab Sheet 9

1. Write a program to chat using TCP/IP

Source Code

```
//server.java
import java.net.*;
import java.io.*;
class server
{
        public static void main(String[] args)
        {
                while(true)
                {
                try
                {
                        ServerSocket listener=new ServerSocket(5000);
                        Socket to_client=listener.accept();
                        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
                        InputStream in =to_client.getInputStream();
                        OutputStream out=to_client.getOutputStream();
                        BufferedReader scanf=new BufferedReader(new
InputStreamReader(in));
                        String someString=scanf.readLine();
                        PrintWriter pout=new PrintWriter(out,true);

                        pout.println(br.readLine());
                        System.out.println(someString);
                        to_client.close();

                listener.close();
                }
                catch(Exception ie)
                {
                        System.out.println("Error");
                }
                }
        }
}

//client.java
import java.net.*;
import java.io.*;
class Client
{
        public static void main(String[] args)
        {
                Socket client;
                while(true)
                {
                        try
                        {
                                client=new Socket("localhost",5000);
                                BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
                                InputStream in=client.getInputStream();
                                OutputStream out=client.getOutputStream();
```

```
                                        BufferedReader scanf=new BufferedReader(new
InputStreamReader(in));
                                        PrintWriter pout=new PrintWriter(out,true);
                                        pout.println(br.readLine());
                                        String some=scanf.readLine();
                                        System.out.println("From Server:"+some);
                                        client.close();
                                }catch(Exception e)
                                {
                                        System.out.println(e);
                                }
                        }

                }
        }
```

2. Write a program Using UDP for chatting in Java

Source Code

```
//Demonstrate Datagrams.
import java.net.*;

class WriteServer
{
        public static int serverPort=5000;
        public static int clientPort=5001;
        public static int buffer_size=1024;
        public static DatagramSocket ds;
        public static byte buffer[]=new byte[buffer_size];

        public static void TheServer() throws Exception
        {
                int pos=0;
                while(true)
                {
                        int c=System.in.read();
                        switch (c)
                        {
                        case -1:
                                System.out.println("Server quits");
                                return;
                        case '\r':
                                break;
                        case '\n':
                                ds.send(new
DatagramPacket(buffer,pos,InetAddress.getLocalHost(),clientPort));
                                pos=0;
                                break;
                        default:
                                buffer[pos++]=(byte)c;
                        }
                }
        }

        public static void TheClient() throws Exception
        {
                while (true)
                {
                        DatagramPacket p=new DatagramPacket(buffer,buffer.length);
                        ds.receive(p);
                        System.out.println(new String(p.getData(),0,p.getLength()));
                }
        }

        public static void main(String args[]) throws Exception
        {
                if (args.length==1)
                {
                        ds=new DatagramSocket(serverPort);
                        TheServer();
                }
                else
                {
                        ds=new DatagramSocket(clientPort);
                        TheClient();
                }
        }
}
```

Lab Sheet 10

1. Write a program to implement the RMI

Background

RMI allows us to execute methods on remote server. It helps us locate and execute methods of remote objects. It's like placing a class on Machine A and calling methods of that class from Machine B as through they were from the same machine

Source code

```
//AddServer.java
import java.rmi.*;

public interface AddServer extends Remote
{
        public int AddNumbers(int fno,int sno) throws RemoteException;
}

//AddServerImpl.java
import java.rmi.*;
import java.rmi.server.*;
public class AddServerImpl extends UnicastRemoteObject implements AddServer
{

        public AddServerImpl() throws RemoteException
        {
                super();
        }

        public int AddNumbers(int fno,int sno) throws RemoteException
        {

                        return fno+sno;
        }
}

//RmiServer.java
import java.rmi.*;
import java.net.*;

public class RmiServer
{
        public static void main(String args[]) throws RemoteException,
MalformedURLException
        {
                AddServerImpl add=new AddServerImpl();
                Naming.rebind("addnumbers",add);
        }
}

//RmiClient.java
import java.rmi.*;
import java.net.*;
public class RmiClient
{
        public static void main(String args[]) throws RemoteException,
MalformedURLException, NotBoundException
        {
                String url="rmi://127.0.0.1/addnumbers";
                AddServer add;
```

```
                        add=(AddServer)Naming.lookup(url);
                        int result =add.AddNumbers(10,5);
                        System.out.println(result);
                }
        }
```