LAB MANUAL

ADVANCE JAVA

(CSE-406-F)

V SEMESTER

DEPARTMENT OF ELECTRONICS & COMPUTER ENGG
DRONACHARYA COLLEGE OF ENGINEERING
KHENTAWAS, GURGAON- 123506

INTRODUCTION

Java is an object-oriented programming language developed by Sun Microsystems in the early 1990s. Java applications are, in the official implementation, compiled to bytecode, which is compiled to native machine code at runtime. Sun Microsystems provides a GNU General Public License implementation of a Java compiler and Java virtual machine, in compliance with the specifications of the Java Community Process.

The language itself borrows much syntax from C and C++ but has a simpler object model and fewer low-level facilities. JavaScript, a scripting language, shares a similar name and has similar syntax, but is not related to Java

Objects

Classes

Java has *nested* classes that are declared within the body of another class or interface. A class that is not a nested class is called a *top level* class. An *inner class* is a non-static nested class.

Classes can be declared with the following modifiers:

abstract – cannot be instantiated. Only interfaces and abstract classes may contain abstract methods. A concrete (non-abstract) subclass that extends an abstract class must override any inherited abstract methods with non-abstract methods. Cannot be final.

final – cannot be subclassed. All methods in a final class are implicity final. Cannot be abstract.

strictfp – all floating-point operations within the class and any enclosed nested classes use strict floating-point semantics. Strict floating-point semantics guarantee that floating-point operations produce the same results on all platforms.

Method overloading is a feature found in various object oriented programming languages such as C++ and Java that allows the creation of several functions with the same name which differ from each other in terms of the type of the input and the type of the output of the function.

An example of this would be a square function which takes a number and returns the square of that number. In this case, it is often necessary to create different functions for integer and floating point numbers.

Method overloading is usually associated with statically-typed programming languages which enforce type checking in function calls. When overloading a method, you are really just making a number of different methods that happen to have the same name. It is resolved at compile time which of these methods are used.

Method overloading should not be confused with ad-hoc polymorphism or virtual functions. In those, the correct method is chosen at runtime.

Method overriding, in object oriented programming, is a language feature that allows a subclass to provide a specific implementation of a method that is already provided by one of its superclasses. The implementation in the subclass overrides (replaces) the implementation in the superclass.

A subclass can give its own definition of methods which also happen to have the same signature as the method in its superclass. This means that the subclass's method has the same name and parameter list as the superclass's overridden method. Constraints on the similarity of return type vary from language to language, as some languages support covariance on return types.

Method overriding is an important feature that facilitates polymorphism in the design of object-oriented programs.

Some languages allow the programmer to prevent a method from being overridden, or disallow method overriding in certain core classes. This may or may not involve an inability to subclass from a given class.

In many cases, abstract classes are designed — i.e. classes that exist only in order to have specialized subclasses derived from them. Such abstract classes have methods that do not perform any useful operations and are meant to be overridden by specific implementations in the subclasses. Thus, the abstract superclass defines a common interface which all the subclasses inherit.

Examples

This is an example in Python. First a general class ("Person") is defined. The "self" argument refers to the instance object. The Person object can be in one of three states, and can also "talk".

```
class Person:
    def __init__(self):
        self.state = 0

    def talk(self, sentence):
        print sentence

    def lie_down(self):
        self.state = 0

    def sit_still(self):
        self.state = 1
```

```
    def stand(self):
        self.state = 2
```

Then a "Baby" class is defined (subclassed from Person). Objects of this class cannot talk or change state, so exceptions (error conditions) are raised by all methods except "lie_down". This is done by overriding the methods "talk", "sit_still" and "stand"

```
class Baby(Person):
    def talk(self, sentence):
        raise CannotSpeakError, 'This person cannot speak.'

    def sit_still(self):
        raise CannotSitError, 'This person cannot sit still.'

    def stand(self):
        raise CannotStandError, 'This person cannot stand up.'
```

For loop

```
for (initial-expr; cond-expr; incr-expr) {
   statements;
}
```

For-each loop

J2SE 5.0 added a new feature called the for-each loop, which greatly simplifies the task of iterating through every element in a collection. Without the loop, iterating over a collection would require explicitly declaring an iterator:

```
public int sumLength(Set<String> stringSet) {
  int sum = 0;
  Iterator<String> itr = stringSet.iterator();
  while (itr.hasNext())
    sum += itr.next().length();
  return sum;
}
```

The for-each loop greatly simplifies this method:

```
public int sumLength(Set<String> stringSet) {
  int sum = 0;
  for (String s : stringSet)
    sum += s.length();
  return sum
}
```

Arrays

1. Java has array types for each type, including arrays of primitive types, class andinterface types, as well as higher-dimensional arrays of array types.
2. All elements of an array must descend from the same type.
3. All array classes descend from the class java.lang.Object, and mirror the hierarchy of the types they contain.
4. Array objects have a read-only length attribute that contains the number of elements in the array.
5. Arrays are allocated at runtime, so the specified size in an array creation expression may be a variable (rather than a constant expression as in C).

Exception handling

Exception handling is a programming language construct or computer hardware mechanism designed to handle the occurrence of some condition that changes the normal flow of execution. The condition is called an **exception**. Alternative concepts are signal and event handler.

In general, current state will be saved in a predefined location and execution will switch to a predefined handler. Depending on the situation, the handler may later resume the execution at the original location, using the saved information to restore the original state. For example, an exception which will usually be resumed is a page fault, while a division by zero usually cannot be resolved transparently.

From the processing point of view, hardware interrupts are similar to resumable exceptions, except they are usually not related to the current program flow.

INDEX

List of Practical's (Advance Java)

S.No	List of Experiments
1.	WAP on Network Programming i.e. Client-Server Programming.
2.	WAP on Multithreading using runnable interface.
3.	WAP to Create a New Data Source for Ms. Access
4.	WAP to show connectivity with database using JDBC/ODBC driver.
5.	WAP to get Information about database using Database Meta Data
6.	WAP to get Information about particular table using Result Set Meta Data
7.	WAP to implement the concept of swings.
8.	WAP to develop an RMI application.
9	WAP in Servlets to get and display value from an HTML page.
10	WAP in JSP to get and display value from an HTML page.

PROGRAM - 1

Write A Program Related To Network Programming i.e. Client-Server Programming.

Client.java

```
import java.io.*;
import java.net.*;
public class client
{
	public static void main(String s[] )throws Exception
	{
		try
	{
		Socket server;
			String str="";
		DataInputStream d=new DataInputStream(System.in);
				PrintStream toserver;
			BufferedReader fromserver;
			server=new Socket("117.198.209.28",1096);
InputStreamReader isr=new InputStreamReader (server.getInputStream());
			fromserver= new BufferedReader(isr);
			toserver=new PrintStream(server.getOutputStream());
		while(true)
			{
```

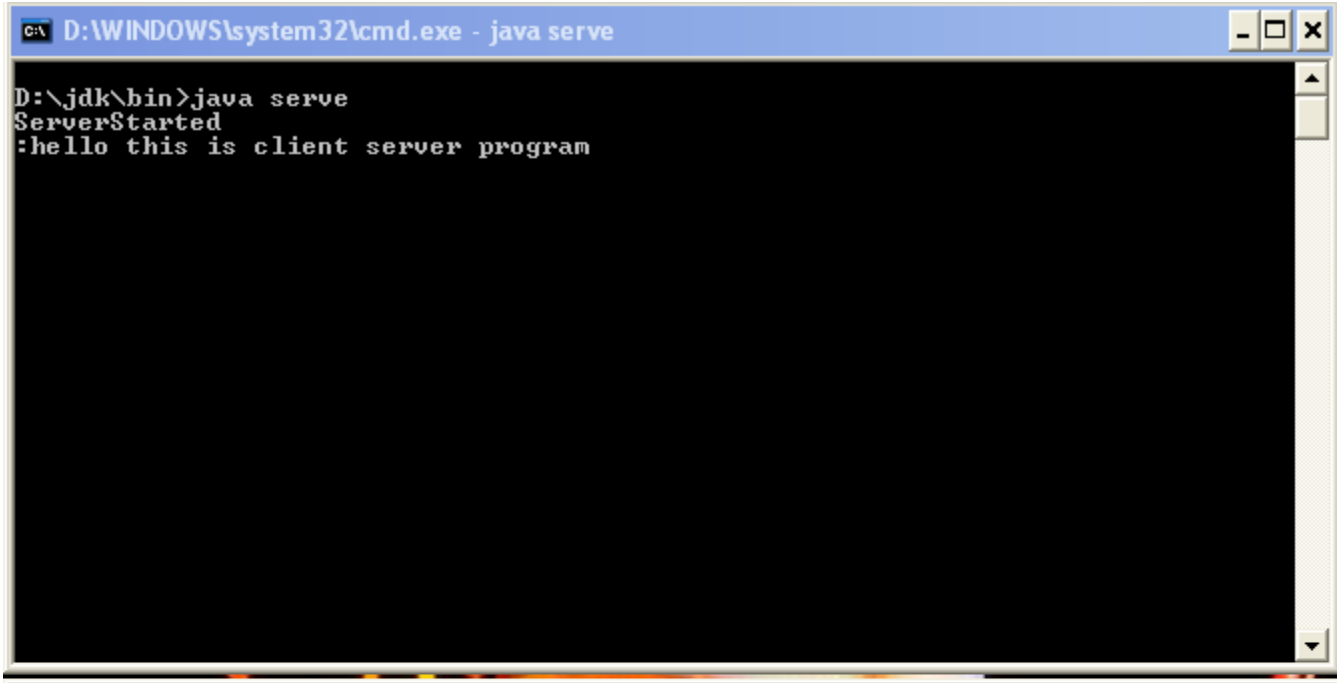
```
                        str=":"+d.readLine();
                                toserver.println(str);
                                str=fromserver.readLine();
                                System.out.println(str);
                            }
                        }
                catch(Exception e)
                        {
                          System.out.println(e);
                        }
            }
}
```

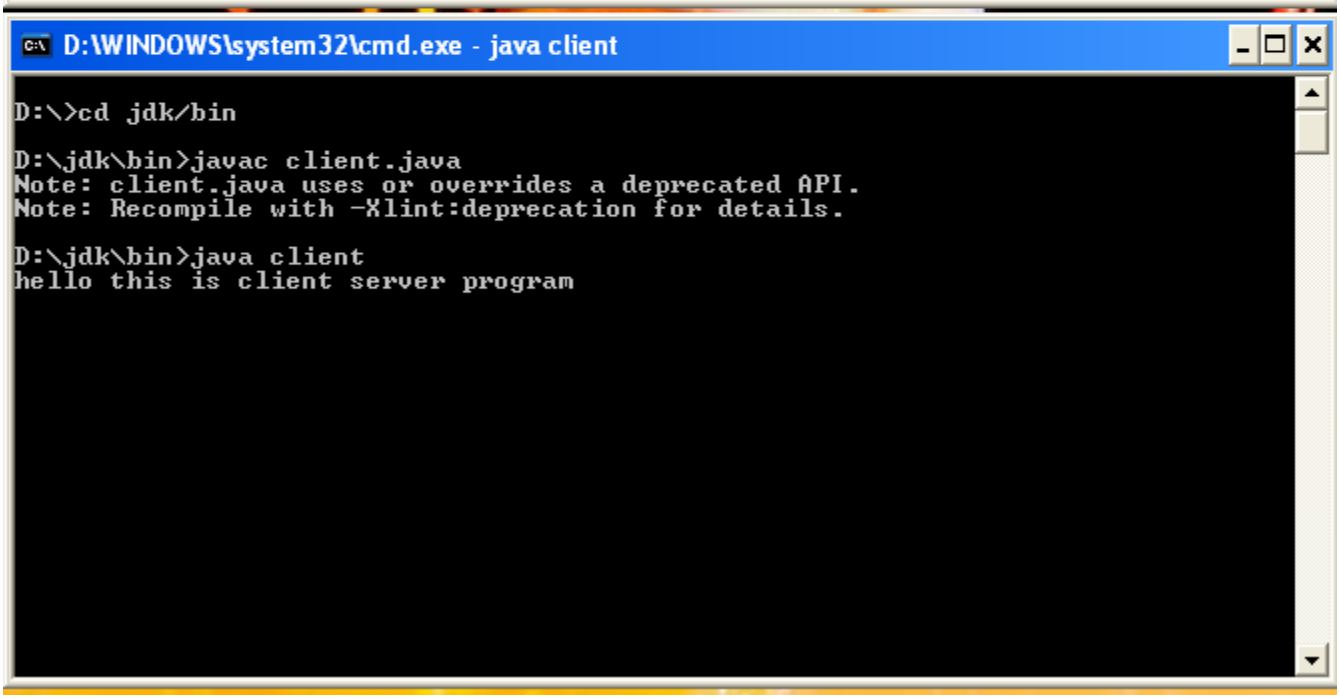
Serve.java

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


public class serve
{
      public static void main(String s[])throws Exception
      {
            ServerSocket sc;
            Socket client;
            DataInputStream d;
            PrintStream toClient;
```

```
        BufferedReader fromClient;
        String str="";
        try
        {
            d=new DataInputStream(System.in);
            sc=new ServerSocket(1096);
            System.out.println("ServerStarted");
            client=sc.accept();
            InputStreamReader isr=new InputStreamReader(client.getInputStream());
            fromClient=new BufferedReader(isr);
            toClient=new PrintStream(client.getOutputStream());
            while(true)
            {
                str=fromClient.readLine();
                System.out.println(str);
                str=":"+d.readLine();
                toClient.println(str);
             }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

OUTPUT -1


D:\WINDOWS\system32\cmd.exe - java serve
D:\jdk\bin>java serve
ServerStarted
:hello this is client server program


D:\WINDOWS\system32\cmd.exe - java client
D:\>cd jdk/bin
D:\jdk\bin>javac client.java
Note: client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
D:\jdk\bin>java client
hello this is client server program

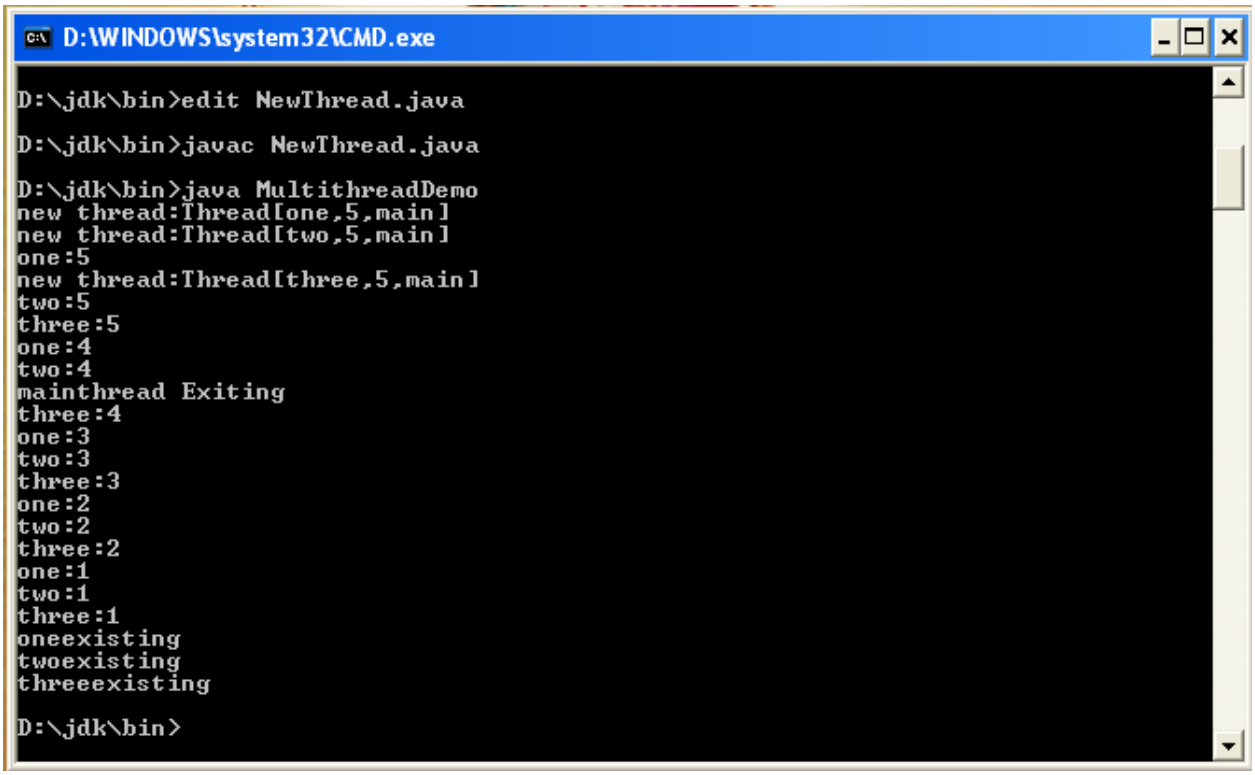
PROGRAM - 2

Write A Program To Perform Multithreading Operation

```
import java.io.*;
import java.net.*;
class NewThread implements Runnable
{
	String name;
	Thread t;
	NewThread(String threadname)
	{
		name=threadname;
		t=new Thread(this,name);
		System.out.println("new thread:"+t);
		t.start();
	}
	public void run()
	{
	try
	{
		for(int i=5;i>0;i--)
		{
			System.out.println(name +":"+i);
			Thread.sleep(1000);
		}
	}
```

```
        catch(InterruptedException e)
        {
                System.out.println(name+"Intrrupted");
        }
                System.out.println(name+"existing");
        }
}
class MultithreadDemo
{
public static void main(String args[]) throws Exception
{
new NewThread("one");
new NewThread("two");
new NewThread("three");
try
{
Thread.sleep(1000);
}
catch(InterruptedException e)
{
System.out.println("mainthread interrupted");
}
System.out.println("mainthread Exiting");
}
}
```

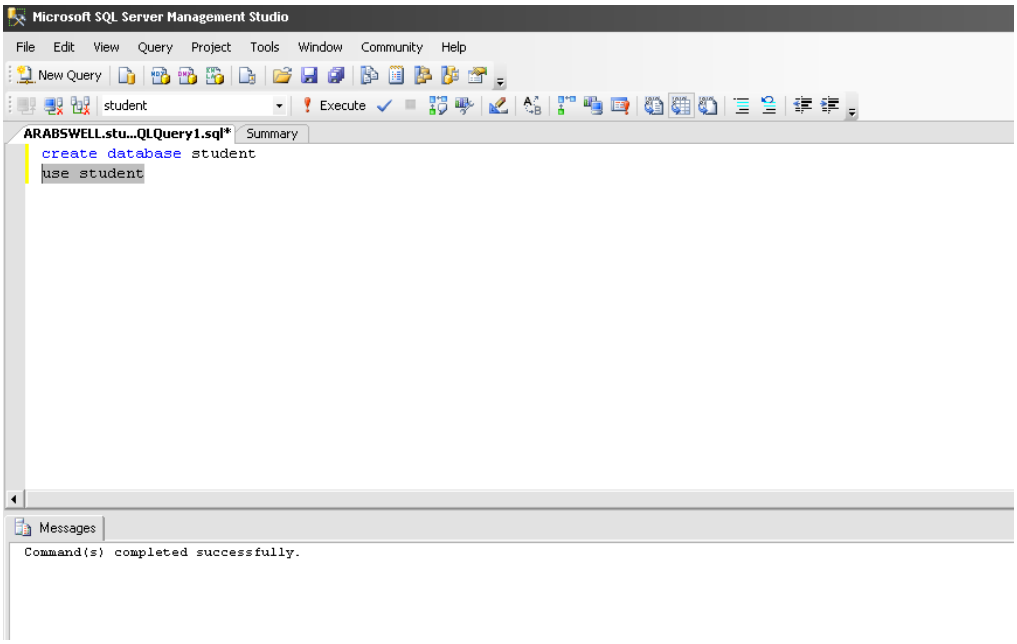
OUTPUT - 2


D:\WINDOWS\system32\CMD.exe
D:\jdk\bin>edit NewThread.java
D:\jdk\bin>javac NewThread.java
D:\jdk\bin>java MultithreadDemo
new thread:Thread[one,5,main]
new thread:Thread[two,5,main]
one:5
new thread:Thread[three,5,main]
two:5
three:5
one:4
two:4
mainthread Exiting
three:4
one:3
two:3
three:3
one:2
two:2
three:2
one:1
two:1
three:1
oneexisting
twoexisting
threeexisting
D:\jdk\bin>

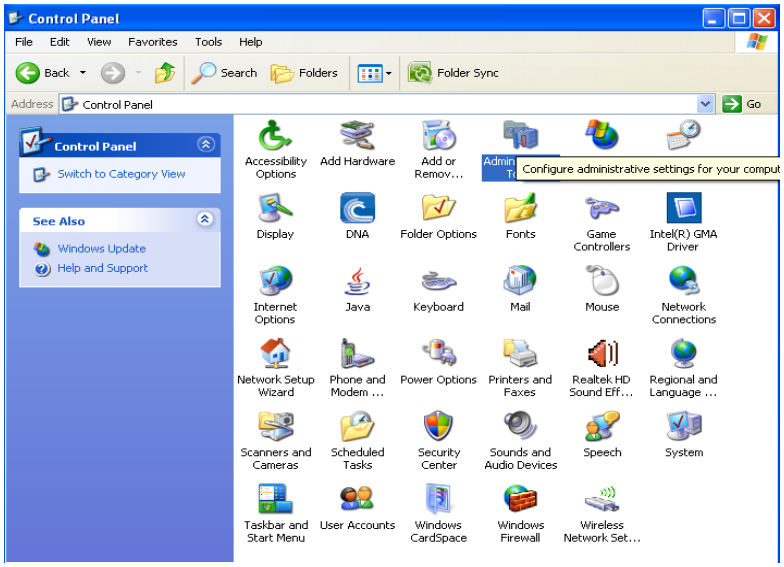
PROGRAM - 3

Write A Program To Create a New Data Source In SQL Server 2005

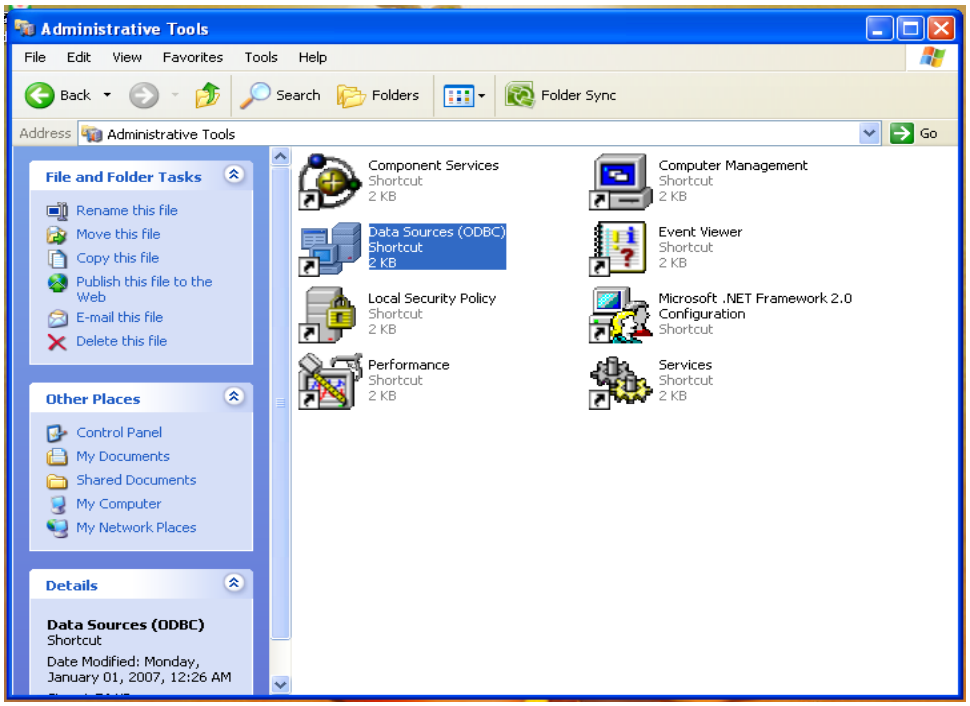
- Create A New Database in Microsoft SQL Server Management Studio 2005



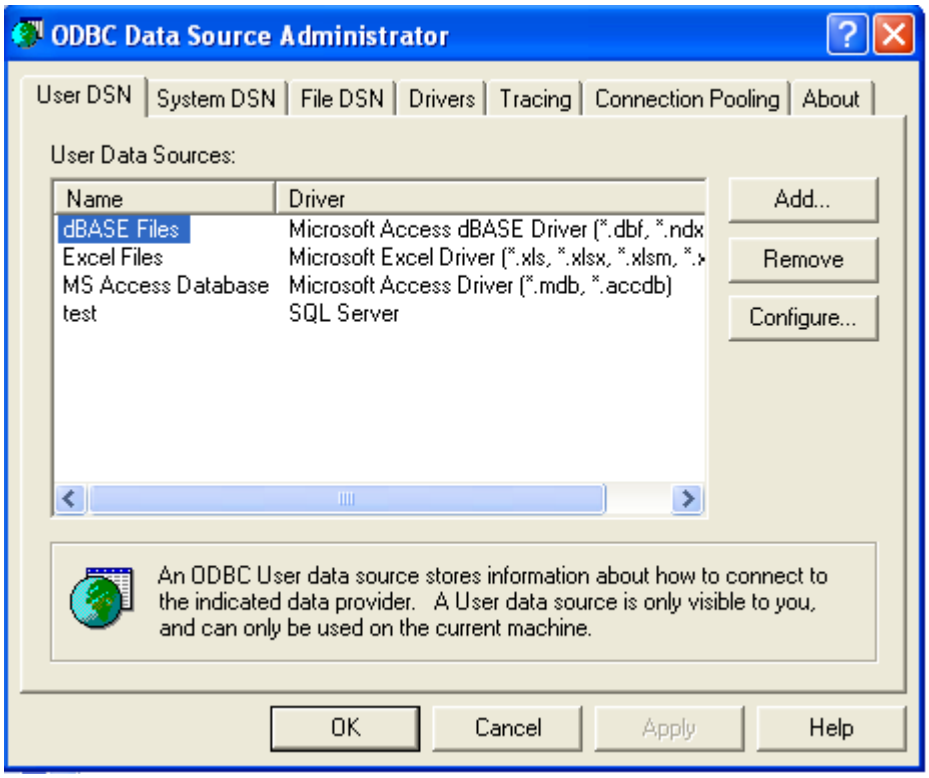
- Click On Administrative Tool In Control Panel



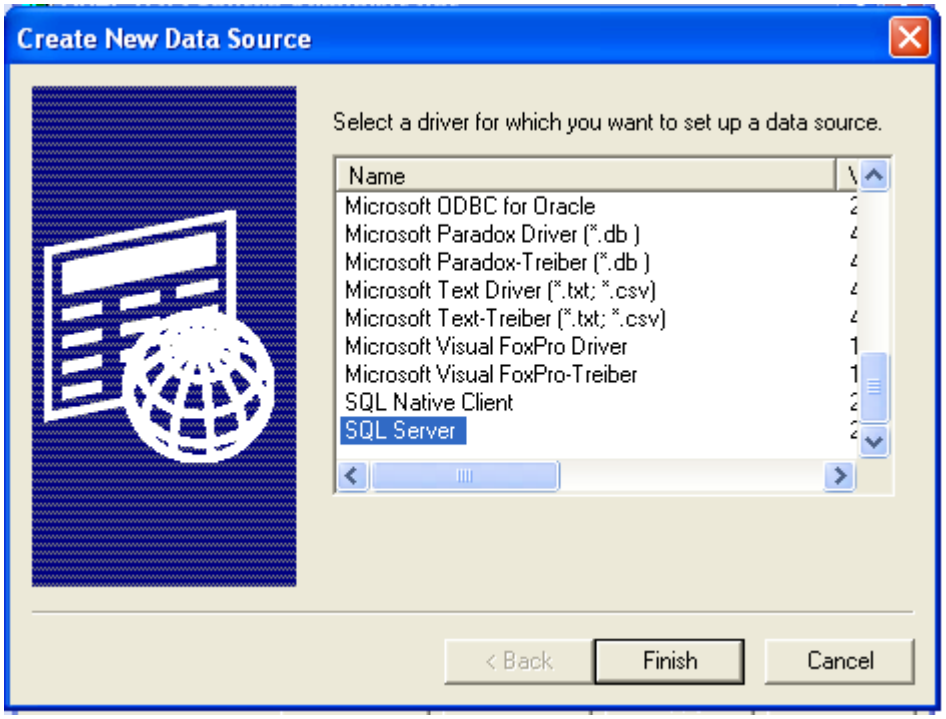
- Click On Data Source (ODBC) to Create A New Data Source



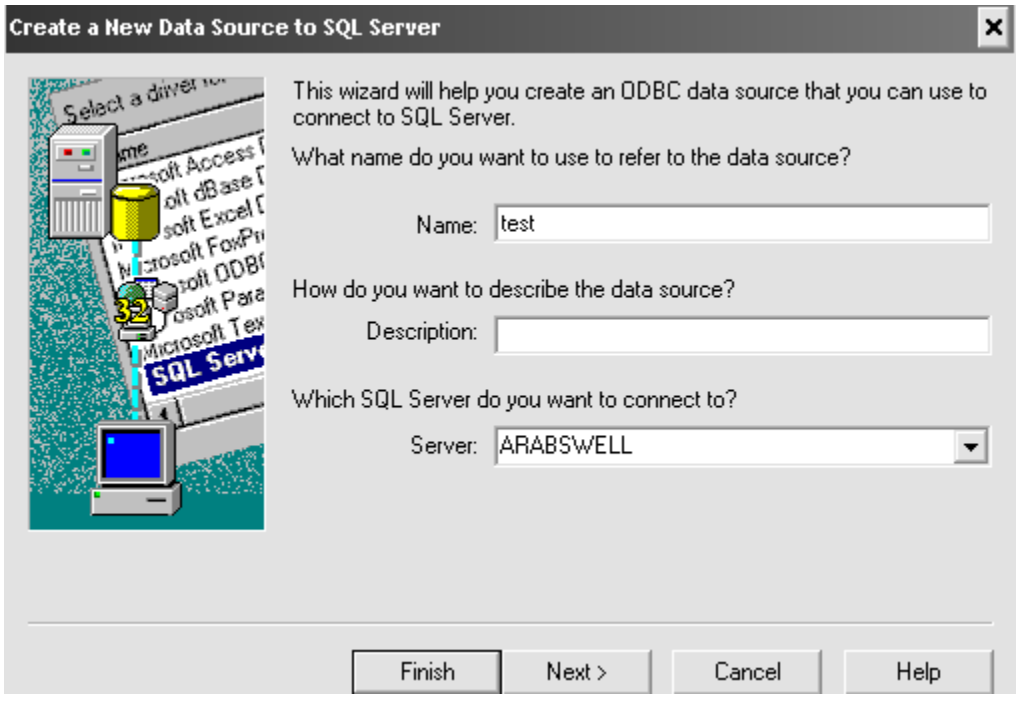
- Click On Add To Create A New User DSN



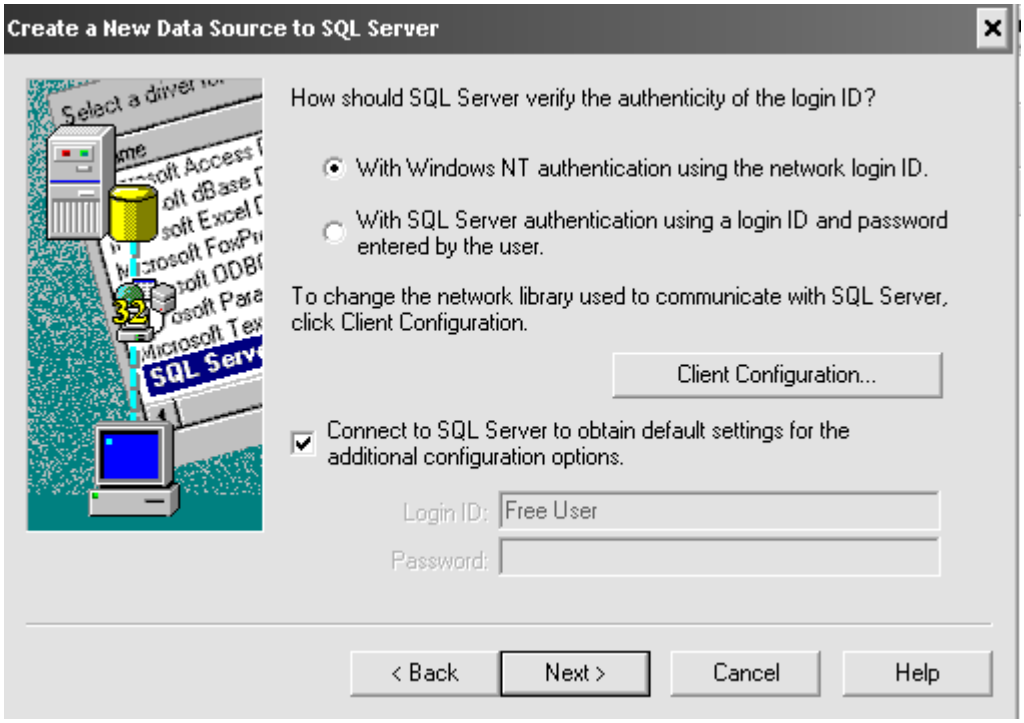
- Click On SQL Server (Database to which connect Data Source



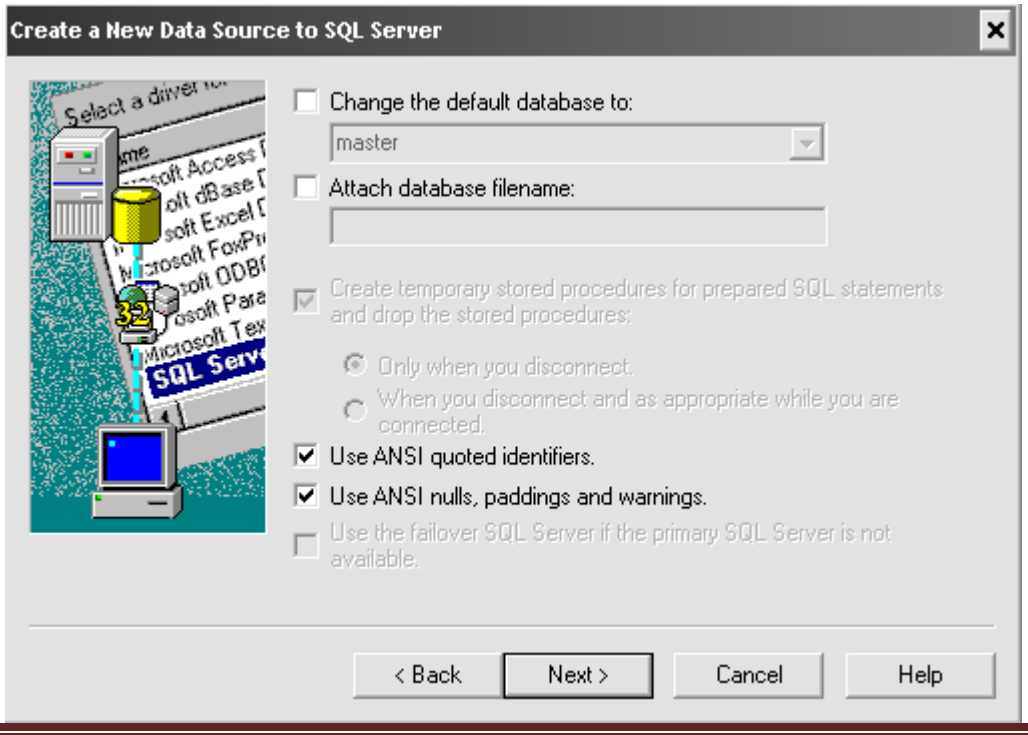
- Filled the Name of Data Source & Server Name Of PC



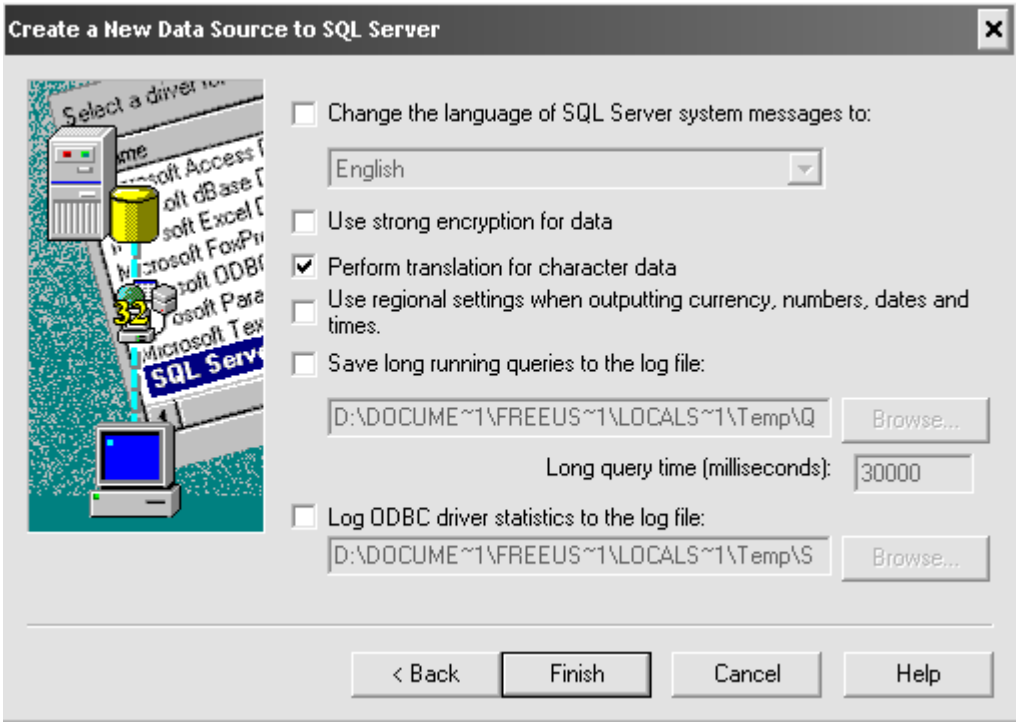
- Choose the Authentication Id for SQL server Verification



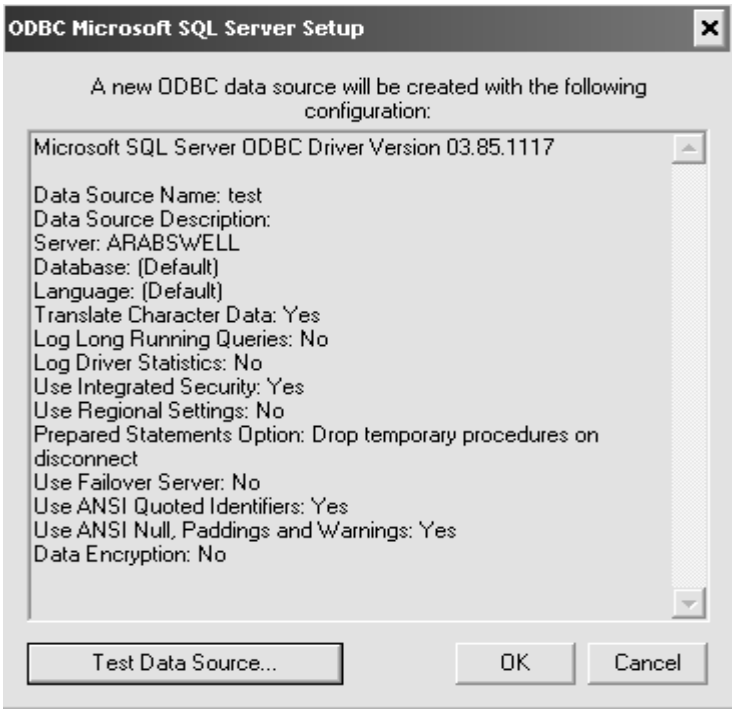
- Choose the Database to which Connect



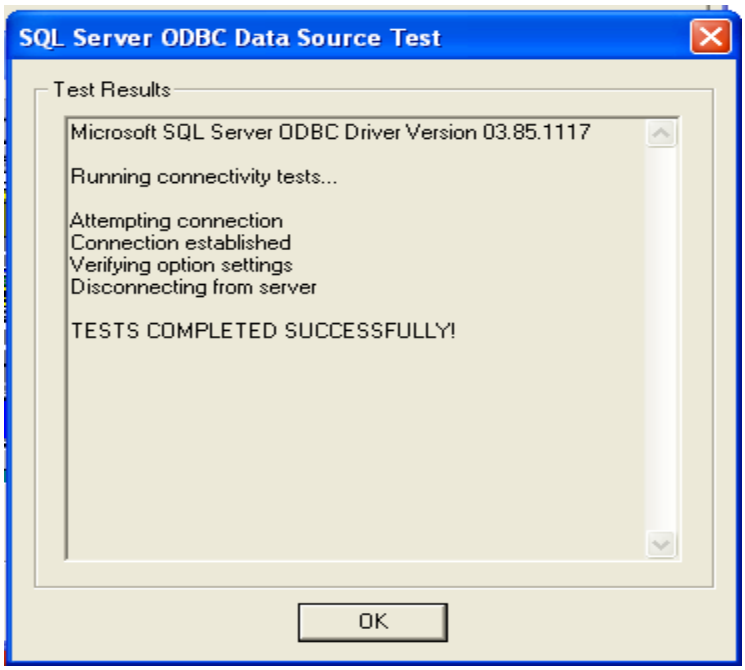
- Click On Finish Button



- Test the New ODBC Data Source



- Check if Successful Then OK otherwise Reconfigure the datasource



PROGRAM - 4

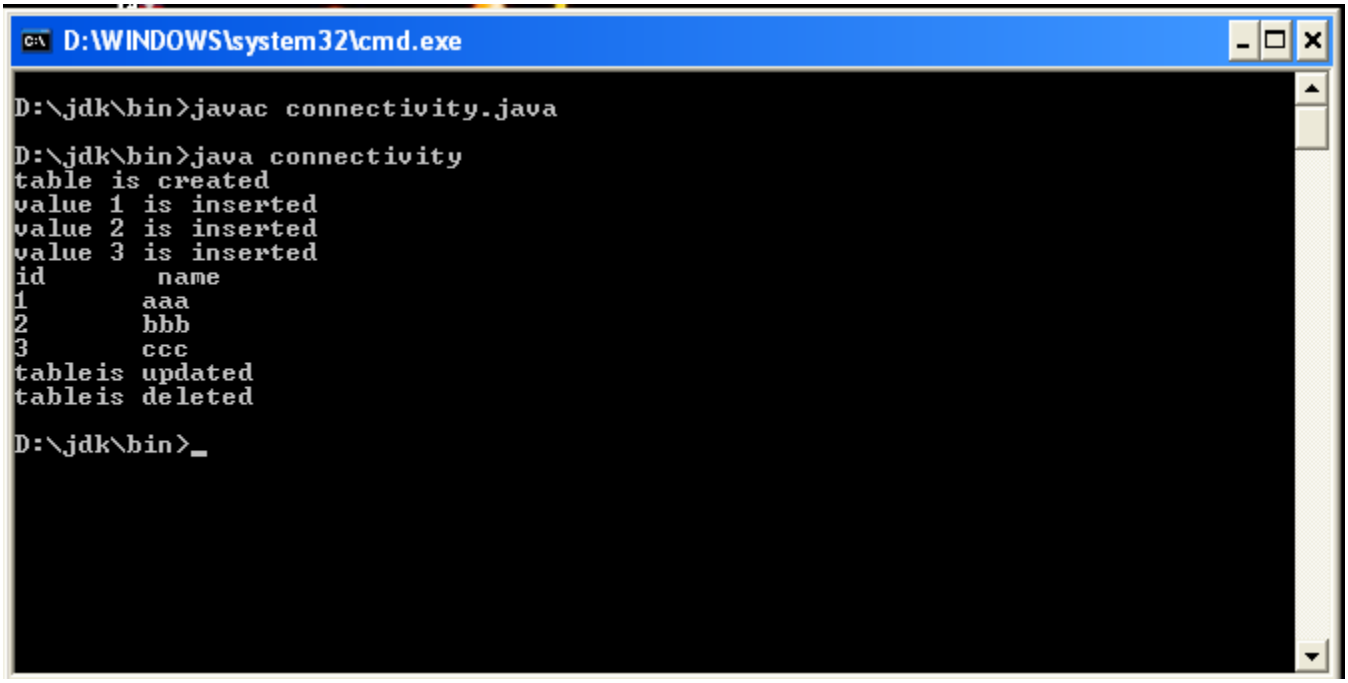
Write A Program To Perform Connectivity in JDBC (Java Database Connectivity) And ODBC (Object Database Connectivity)

```
import java.lang.*;
import java.io.*;
import java.sql.*;
public class connectivity
{
public static void main(String args[])
{
        try
        {
                String str0=”Drop table student”;
                String str1="create table student"+" (c_id integer ,"+ "c_name varchar(20))";
                String str2="insert into student(c_id,c_name)values (1,'aaa')";
                String str3="insert into student(c_id,c_name)values (2,'bbb')";
                String str4="insert into student(c_id,c_name)values (3,'ccc')";
                String str5="select * from student";
                String str6="update student set c_id=5 where c_name='bbb'";
                String str7="delete from student where c_id=5";

                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                Connection con=DriverManager.getConnection("jdbc:odbc:test","hb","");
```

```
Statement stmt=con.createStatement();


Stmt.execute(str0);
stmt.execute(str1);
System.out.println("table is created");
int count1=stmt.executeUpdate(str2);
System.out.println("value 1 is inserted");
int count2=stmt.executeUpdate(str3);
System.out.println("value 2 is inserted");
int count3=stmt.executeUpdate(str4);
System.out.println("value 3 is inserted");

ResultSet rs=stmt.executeQuery(str5);
System.out.println("id \t name");
while(rs.next())
{
        String id=rs.getString("c_id");
        String name=rs.getString("c_name");
        System.out.print(id+"\t");
        System.out.print(name+"\n");
//      System.out.print();
}

int count4=stmt.executeUpdate(str6);
System.out.println("tableis updated");
```

```
                int count5=stmt.executeUpdate(str7);
                System.out.println("tableis deleted");

                con.close();

        }
        catch(Exception ex)
        {
                System.out.println("error occured"+ ex);
        }
}
}
```

OUTPUT - 4


D:\WINDOWS\system32\cmd.exe
D:\jdk\bin>javac connectivity.java
D:\jdk\bin>java connectivity
table is created
value 1 is inserted
value 2 is inserted
value 3 is inserted
id name
1 aaa
2 bbb
3 ccc
tableis updated
tableis deleted
D:\jdk\bin>

PROGRAM - 5

Write A program To Get Information About Database Using Database Meta Data

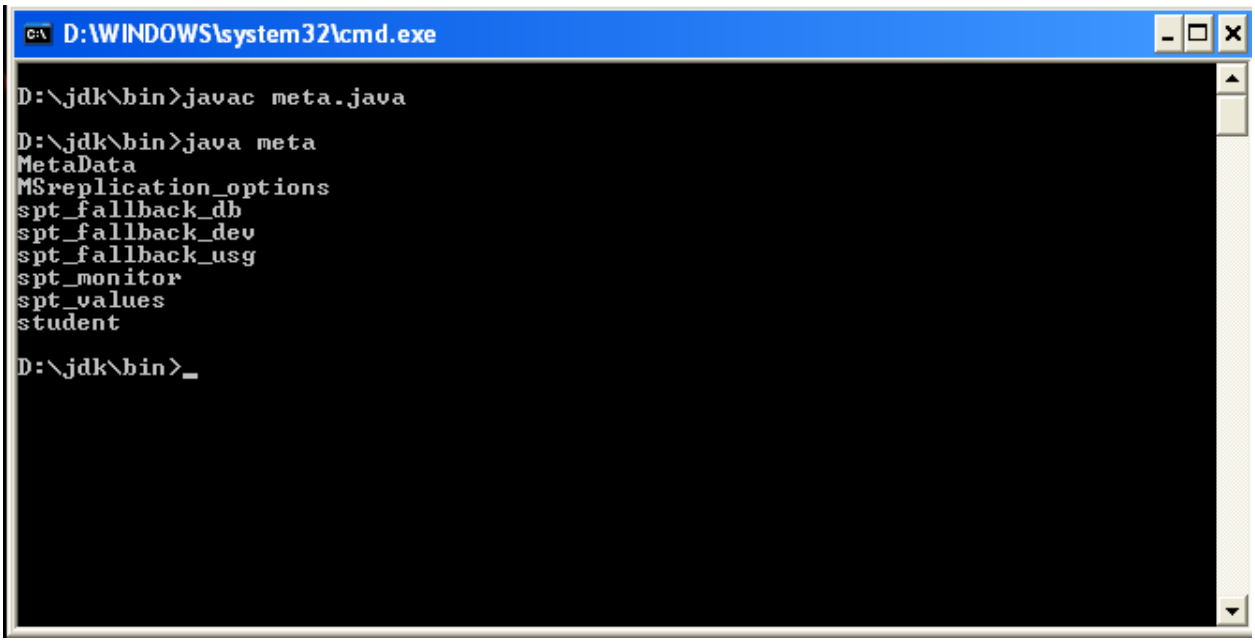
```
import java.lang.*;
import java.sql.*;
public class meta
{
	public static void main(String args[])
	{
		try
		{
			Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
			Connection con=DriverManager.getConnection("jdbc:odbc:test","JA","");
			String []bytes={"table"};
			DatabaseMetaData dbmd=con.getMetaData();
			ResultSet rs=dbmd.getTables(null,null,null,bytes);

			System.out.println("MetaData");
			while(rs.next())
			{
				System.out.println(rs.getString("Table_Name"));
			}
			con.close();
		}
```

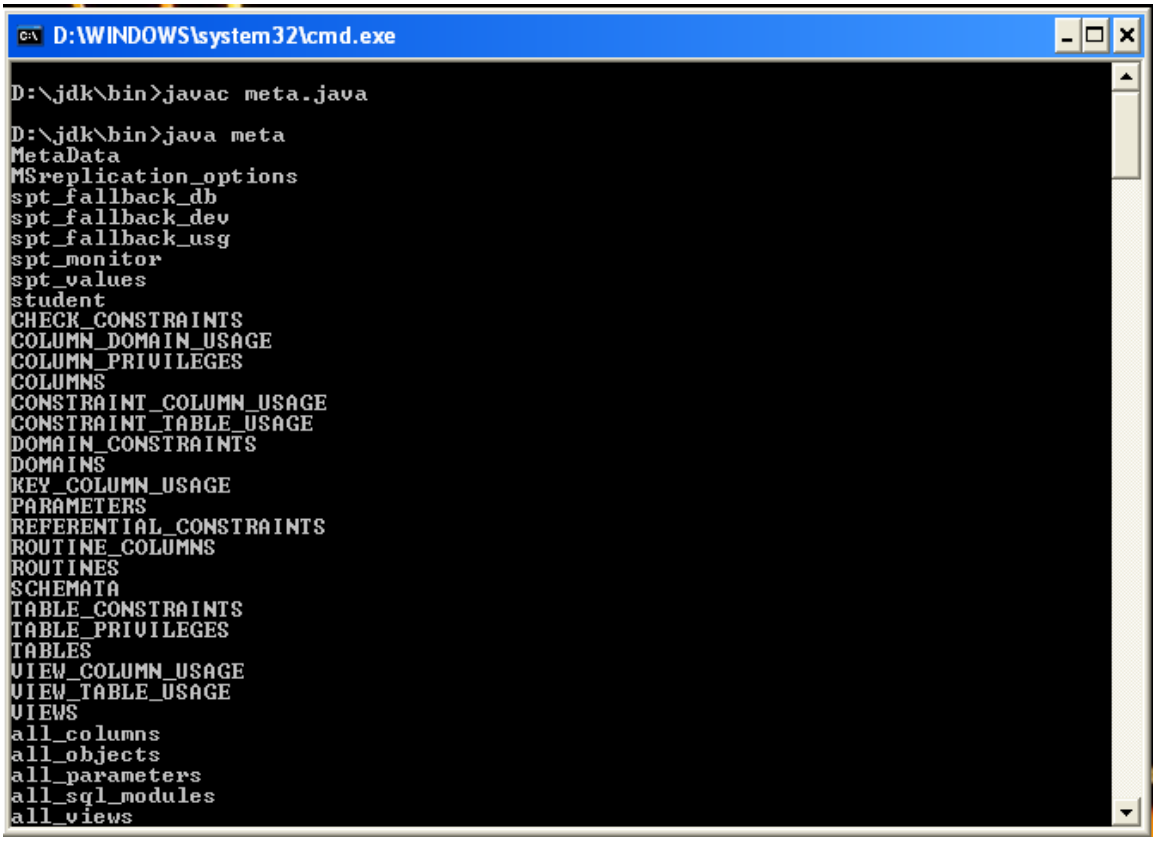
```
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
}
```

To Get All System Defined Tables

```
ResultSet rs=dbmd.getTables(null,null,null,null);
```

OUTPUT - 5


D:\WINDOWS\system32\cmd.exe
D:\jdk\bin>javac meta.java
D:\jdk\bin>java meta
MetaData
MSreplication_options
spt_fallback_db
spt_fallback_dev
spt_fallback_usg
spt_monitor
spt_values
student
D:\jdk\bin>


D:\WINDOWS\system32\cmd.exe
D:\jdk\bin>javac meta.java
D:\jdk\bin>java meta
MetaData
MSreplication_options
spt_fallback_db
spt_fallback_dev
spt_fallback_usg
spt_monitor
spt_values
student
CHECK_CONSTRAINTS
COLUMN_DOMAIN_USAGE
COLUMN_PRIVILEGES
COLUMNS
CONSTRAINT_COLUMN_USAGE
CONSTRAINT_TABLE_USAGE
DOMAIN_CONSTRAINTS
DOMAINS
KEY_COLUMN_USAGE
PARAMETERS
REFERENTIAL_CONSTRAINTS
ROUTINE_COLUMNS
ROUTINES
SCHEMATA
TABLE_CONSTRAINTS
TABLE_PRIVILEGES
TABLES
VIEW_COLUMN_USAGE
VIEW_TABLE_USAGE
VIEWS
all_columns
all_objects
all_parameters
all_sql_modules
all_views

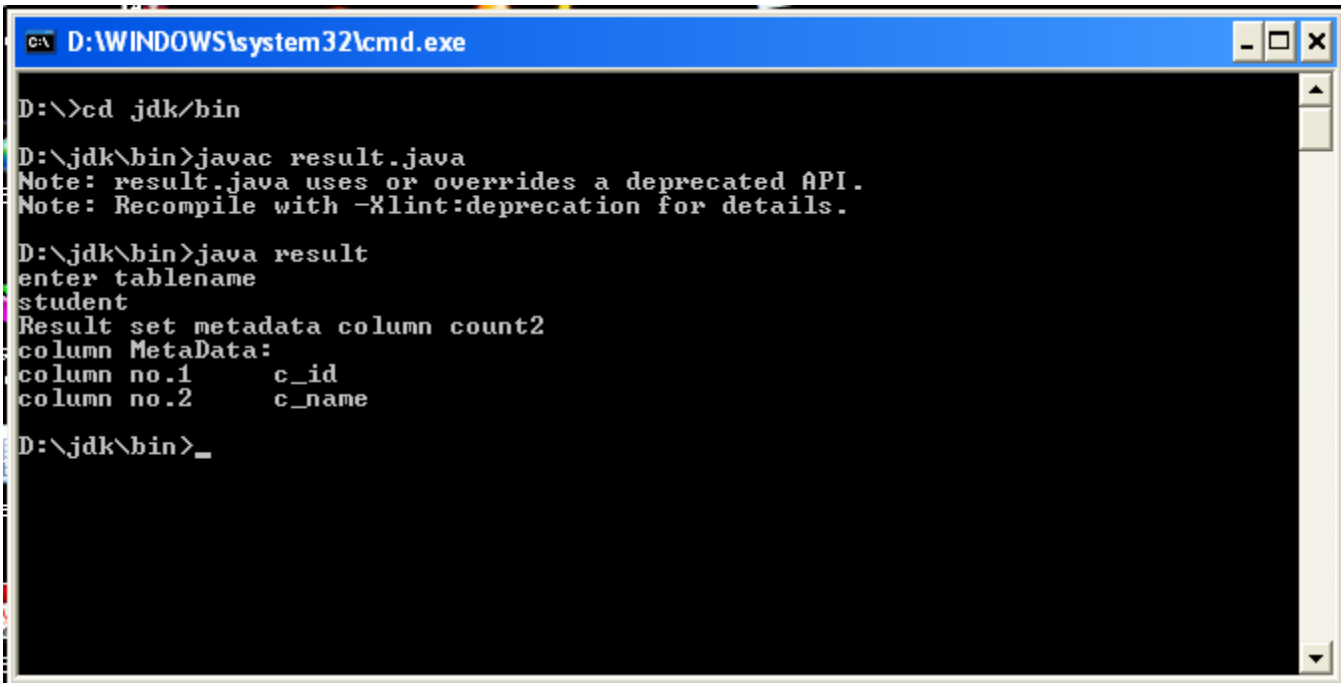
PROGRAM - 6

Write A program To Get Information About Particular Table Using Result Set Meta Data

```
import java.lang.*;
import java.sql.*;
import java.io.*;
public class result
{
	public static void main(String args[])
	{
		try
		{
			Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
			Connection con=DriverManager.getConnection("jdbc:odbc:test","JA","");
			String str=new String();
			DataInputStream d=new DataInputStream(System.in);
			System.out.println("enter tablename");
			str=d.readLine();
			PreparedStatement ps=con.prepareStatement("select * from "+str);
			ResultSet rs=ps.executeQuery();
			ResultSetMetaData rsmd=rs.getMetaData();
			int col=rsmd.getColumnCount();
			System.out.println("Result set metadata column count"+col);
			System.out.println("column MetaData:");
```

```
                for(int i=1;i<=col;i++)
                {
                        System.out.print("column no."+i+"\t");
                        System.out.print(rsmd.getColumnName(i)+"\n");
                }
                con.close();
            }
            catch(Exception e)
            {
                System.out.println(e);
            }

        }
}
```

OUTPUT - 6


D:\WINDOWS\system32\cmd.exe
D:\>cd jdk/bin
D:\jdk\bin>javac result.java
Note: result.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
D:\jdk\bin>java result
enter tablename
student
Result set metadata column count2
column MetaData:
column no.1 c_id
column no.2 c_name
D:\jdk\bin>

PROGRAM - 7

WAP TO IMPLEMENT THE CONCEPT OF SWINGS

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

/*<applet code="swing.class" height=500 width=500></applet>*/

public class swing extends JApplet implements ActionListener
{
    JTextField tf;
    JLabel l1,l2,l3,l4,l5;
    JButton b;
    JComboBox cb;
    JComboBox cb1;
    JList jl;
    JRadioButton m,f;
    ButtonGroup bg;
    JPanel p;
    public void init()
    {
        Color c=new Color(215,248,90);
        p=new JPanel();
        p.setBackground(c);
        l1=new JLabel("Enter Your Name:");
```

```
  l2=new JLabel("Gender:");
  l3=new JLabel("Qualification:");
  l4=new JLabel("Hobbies:");
l5=new JLabel("Year:");

tf=new JTextField(20);
b=new JButton("Show");
  String str[]={"BA","BBA","BCA","B.TECH","MBA","MA","MCA"};
String str1[]={"Watching TV","Reading BOOKS","Watching MOVIES",
                "Playing VIDEO GAMES","Surfing INTERNET"};
  String str2[]={"1st Year","IInd Year","IIIrd Year","IVth Year"};
  GridBagLayout gl=new GridBagLayout();
  GridBagConstraints gbc=new GridBagConstraints();

   p.setLayout(gl);

cb=new JComboBox(str);
jl=new JList(str1);
cb1=new JComboBox(str2);
m=new JRadioButton("Male");
f=new JRadioButton("Female");

        gbc.gridx=0;
        gbc.gridy=0;
        gbc.anchor=GridBagConstraints.CENTER;
```

```
gbc.fill=GridBagConstraints.HORIZONTAL;
gl.setConstraints(l1,gbc);
p.add(l1);

gbc.gridx=1;
gbc.gridy=0;
gbc.gridwidth=2;
gbc.anchor=GridBagConstraints.CENTER;
gl.setConstraints(tf,gbc);
p.add(tf);



gbc.gridx=0;
gbc.gridy=2;
gbc.gridwidth=1;
gbc.anchor=GridBagConstraints.CENTER;
gl.setConstraints(l2,gbc);
p.add(l2);

bg=new ButtonGroup();

gbc.gridx=1;
gbc.gridy=2;
gbc.anchor=GridBagConstraints.CENTER;
```

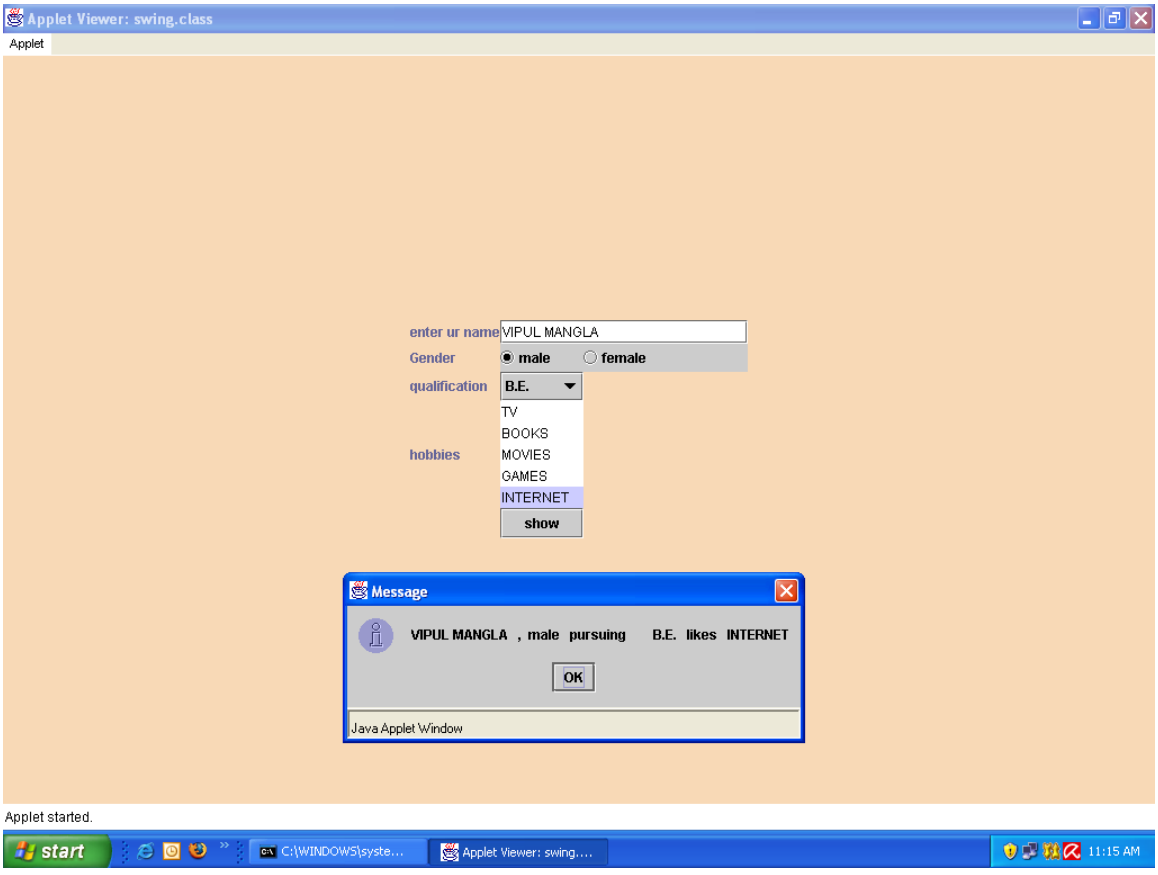
```
bg.add(m);
    gl.setConstraints(m,gbc);
    p.add(m);

    gbc.gridx=2;
    gbc.gridy=2;
    gbc.anchor=GridBagConstraints.CENTER;
 bg.add(f);
    gl.setConstraints(f,gbc);
 p.add(f);

    gbc.gridx=0;
    gbc.gridy=4;
    gbc.anchor=GridBagConstraints.CENTER;
    gl.setConstraints(l3,gbc);
    p.add(l3);

    gbc.gridx=1;
    gbc.gridy=4;
    gbc.anchor=GridBagConstraints.CENTER;


    gl.setConstraints(cb,gbc);
    p.add(cb);
```

```
gbc.gridx=0;
    gbc.gridy=6;
    gbc.anchor=GridBagConstraints.CENTER;
    gl.setConstraints(l5,gbc);
    p.add(l5);

    gbc.gridx=1;
    gbc.gridy=6;
    gbc.anchor=GridBagConstraints.CENTER;
    gl.setConstraints(cb1,gbc);
    p.add(cb1);

    gbc.gridx=0;
    gbc.gridy=8;
    gbc.anchor=GridBagConstraints.CENTER;
    gl.setConstraints(l4,gbc);
    p.add(l4);

    gbc.gridx=1;
    gbc.gridy=8;
    gbc.anchor=GridBagConstraints.NORTH;
    gl.setConstraints(jl,gbc);
    p.add(jl);

    gbc.gridx=1;
```

```
            gbc.gridy=10;
            gbc.gridwidth=1;
            gbc.fill=GridBagConstraints.HORIZONTAL;

            gbc.anchor=GridBagConstraints.CENTER;
            gl.setConstraints(b,gbc);
            p.add(b);

        getContentPane().add(p);
        b.addActionListener(this);
    }
    public void actionPerformed(ActionEvent a)
    {
        String s;
        s=tf.getText();

        if(m.isSelected())
        {
            s="Mr." +s+", " ;
        }
        else
        {
            s="Ms." +s+", ";
        }
```

```
            s=s+" pursuing "+String.valueOf(cb.getSelectedItem());
            s=s+" " +String.valueOf(cb1.getSelectedItem());
            s=s+" likes "+String.valueOf(jl.getSelectedValue());
            JOptionPane.showMessageDialog(this,s);
        }
}
```

OUTPUT - 7



PROGRAM - 8

WAP TO DEVELOP AN RMI SYSTEM

Step 1: Make an Interface "Hello.java":

```
import java.rmi.*;
public interface Hello extends Remote
{
    public String sayHello() throws RemoteException;
}
```

Step 2: Implementation "HelloImpl.java":

```
import java.rmi.*;
import java.rmi.server.*;
public class HelloImpl extends UnicastRemoteObject implements Hello
{
  public HelloImpl() throws RemoteException
  {
      super();
  }
  public String sayHello() throws RemoteException
  {
     return "Hello!peter smith";
  }
```

```
}
```

Step 3: Rebind the Server "HelloServer.java":

```
import java.rmi.*;
import java.rmi.server.*;
public class HelloServer
{
 public static void main(String args[])
 {
   try
   {
     System.setSecurityManager( new RMISecurityManager());
     Hello h=new HelloImpl();
     Naming.rebind("server",h);
     System.out.println("object is registered");
     System.out.println("now servaer is waiting");
     }

        catch(Exception e)
     {
     System.out.println("error:"+e);
     }
     }
     }
```

Step 4: LookUp the Client "HelloClient.java":

```
import java.rmi.*;
public class HelloClient
{
public static void main(String args[])
{
try
{
Hello h=(Hello)Naming.lookup("rmi://192.168.0.66/server");
System.out.println("client: Hello!");
System.out.println("server:" +h.sayHello());
}
catch(Exception e)
{
System.out.println("Error:"+e);
}
}
}
```

Step 5: Compile the four source files

Step 6: Generate the Stub & Skelton using:

rmic HelloImpl

Step 7: Start the RMI registry

start rmiregistry

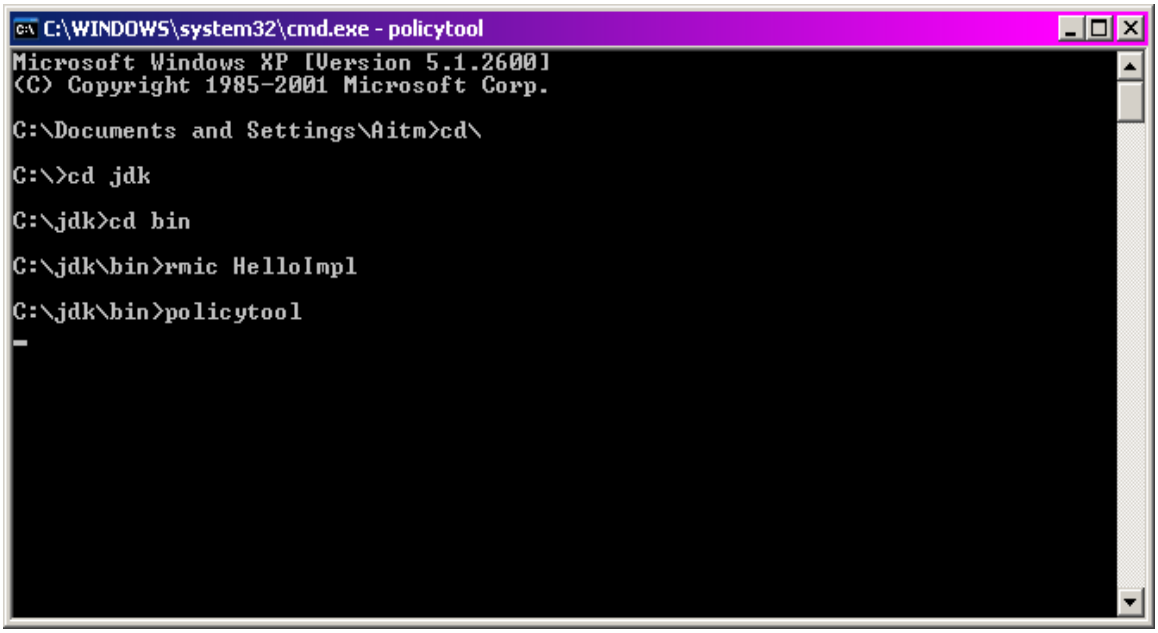
Step 8: start the server

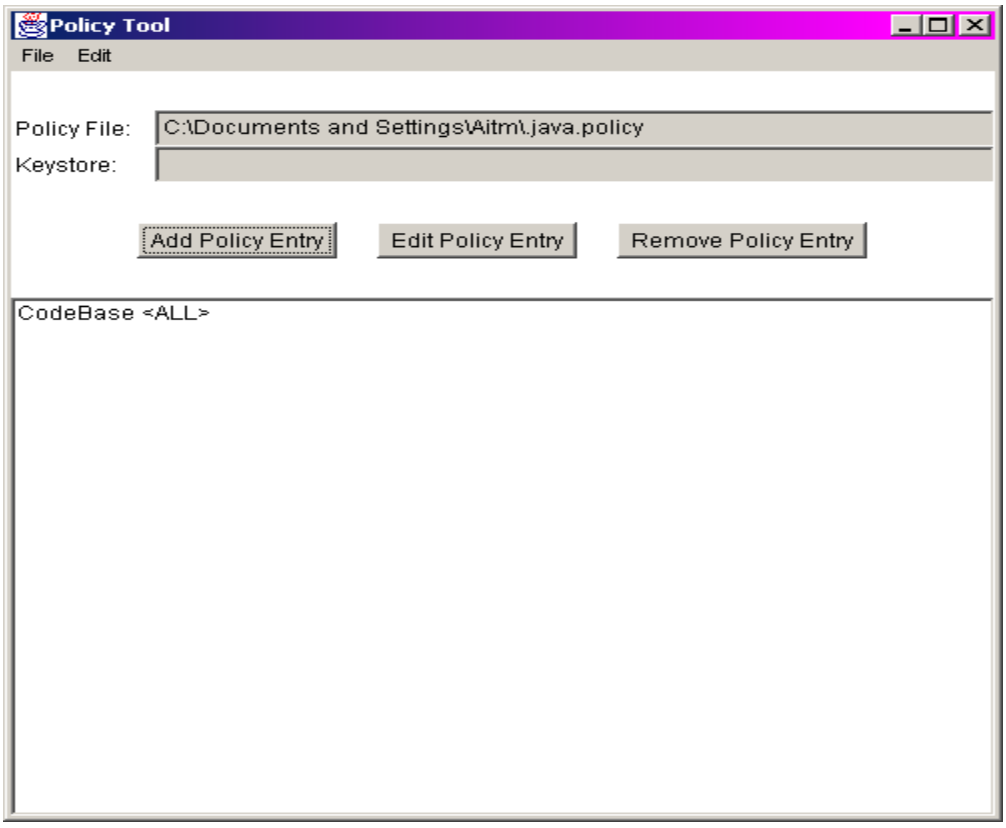
java HelloServer

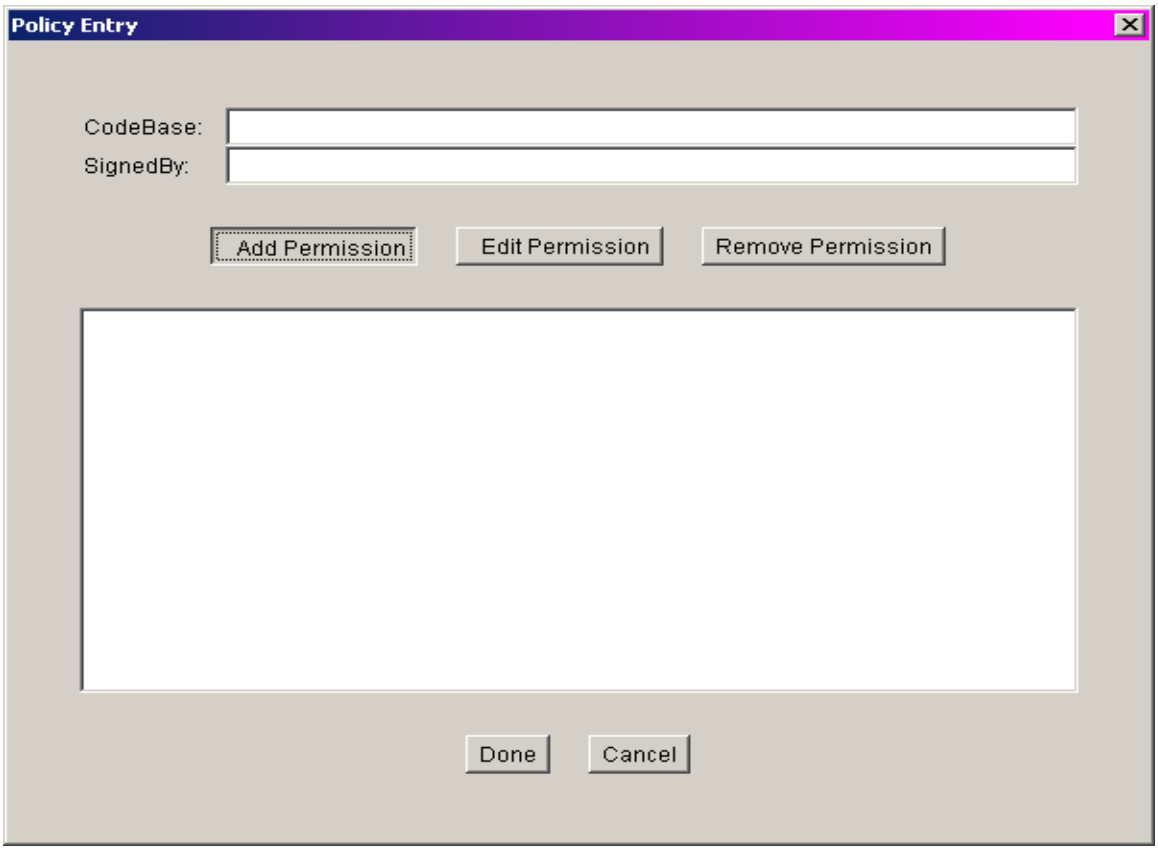
Step 9: Start the client

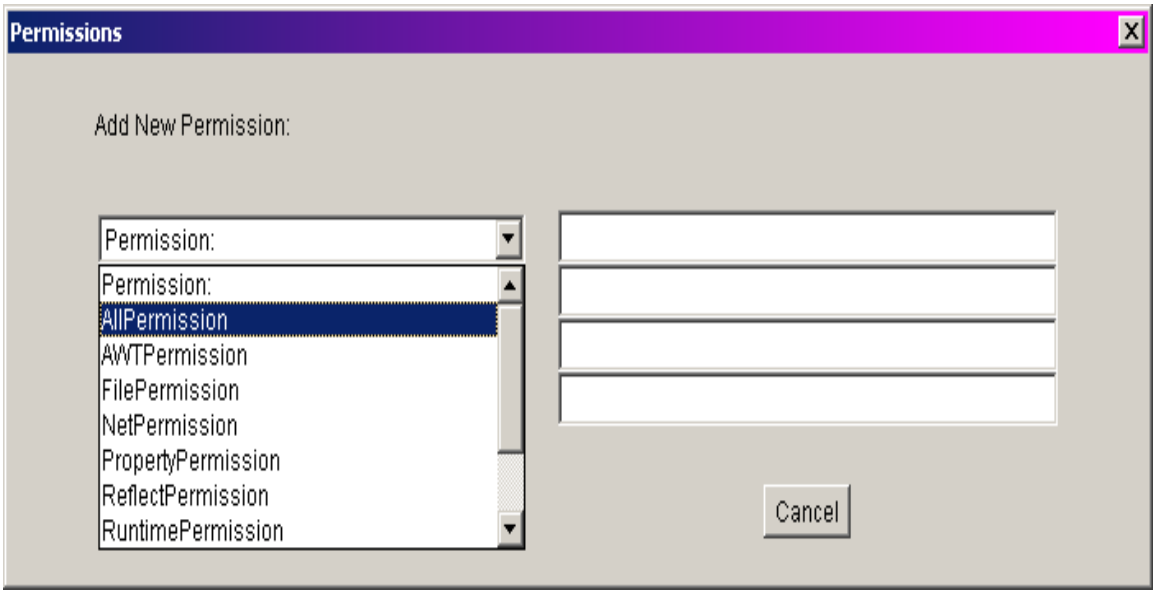
java HelloClient

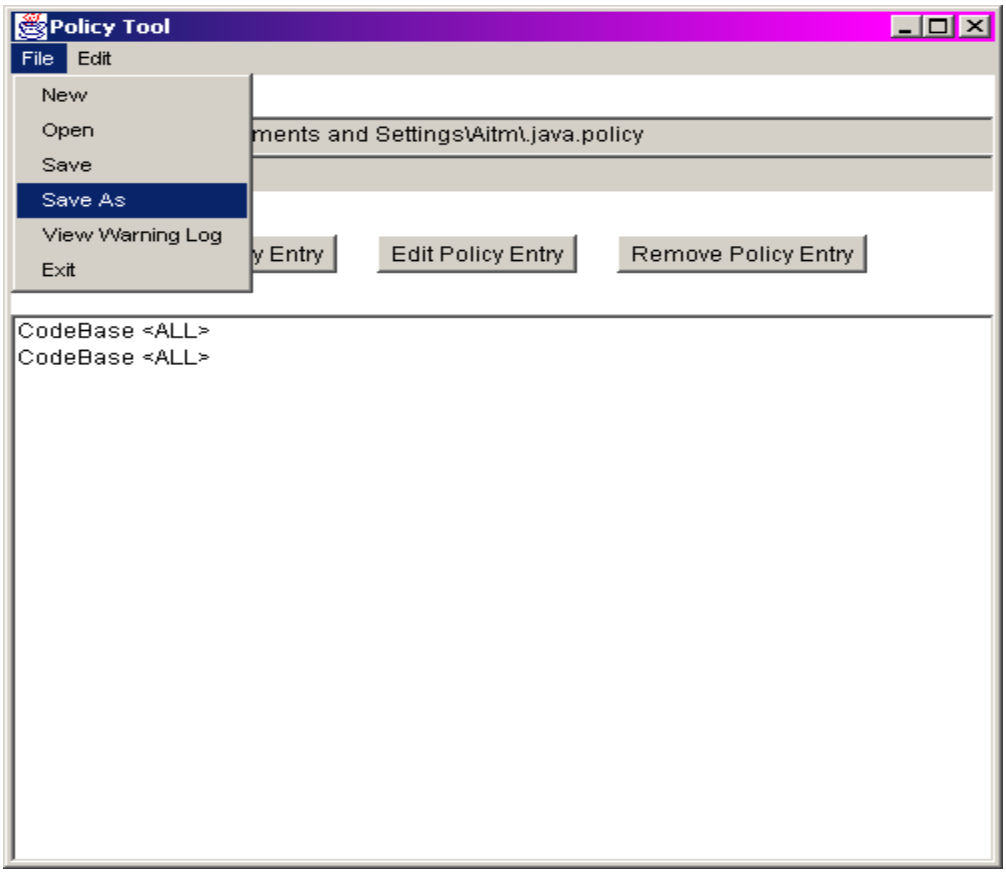
OUTPUT - 8

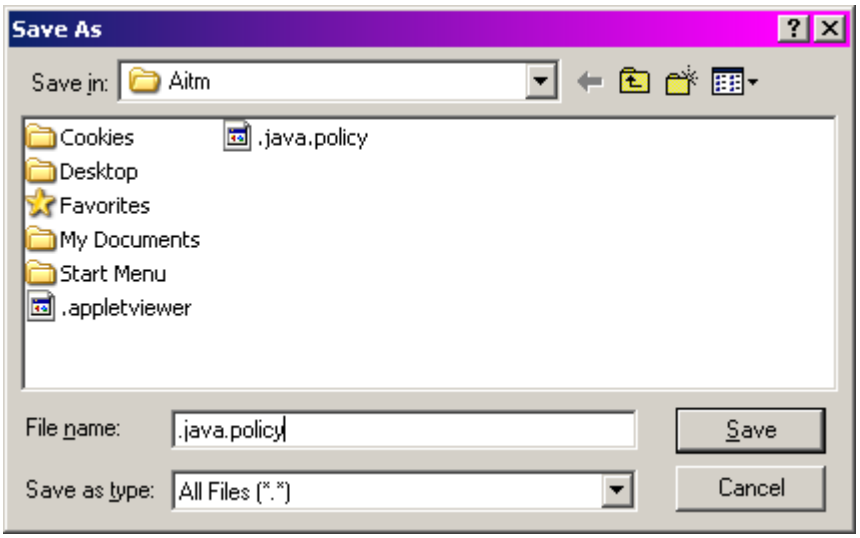

C:\WINDOWS\system32\cmd.exe - policytool
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Aitm>cd\
C:\>cd jdk
C:\jdk>cd bin
C:\jdk\bin>rmic HelloImpl
C:\jdk\bin>policytool

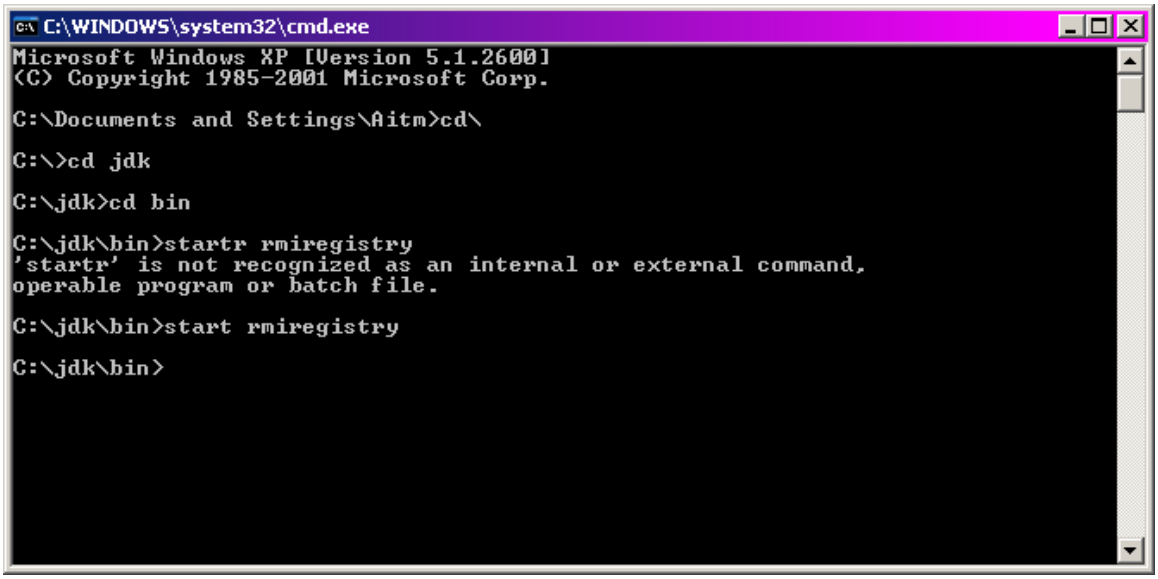

Policy Tool
File Edit
Policy File:
C:\Documents and Settings\Aitm\.java.policy
Keystore:
Add Policy Entry
Edit Policy Entry
Remove Policy Entry
CodeBase <ALL>

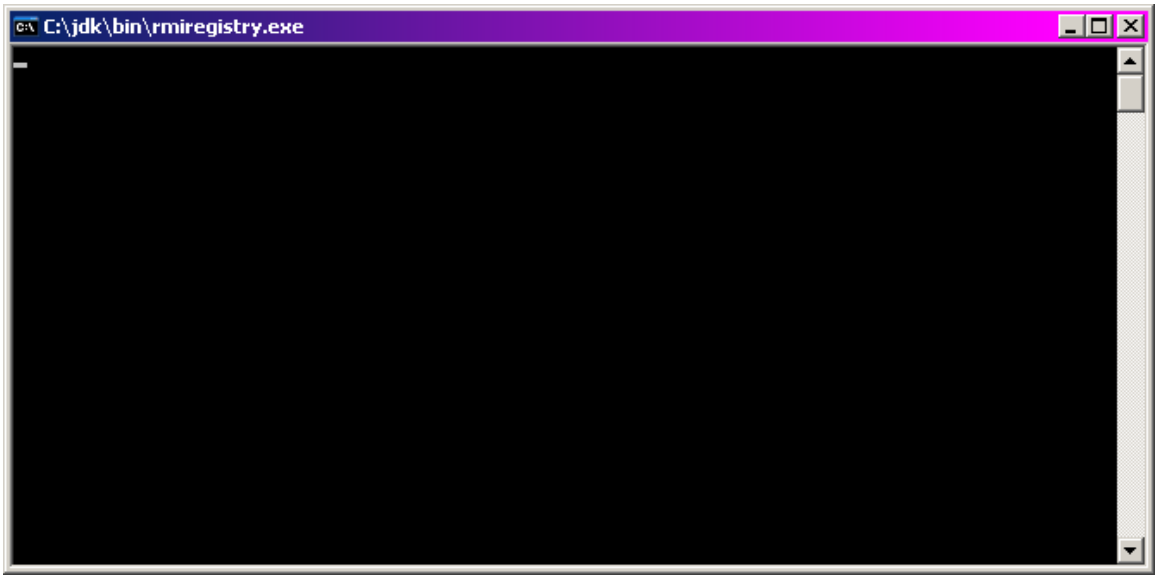

Policy Entry
CodeBase:
SignedBy:
Add Permission
Edit Permission
Remove Permission
Done
Cancel

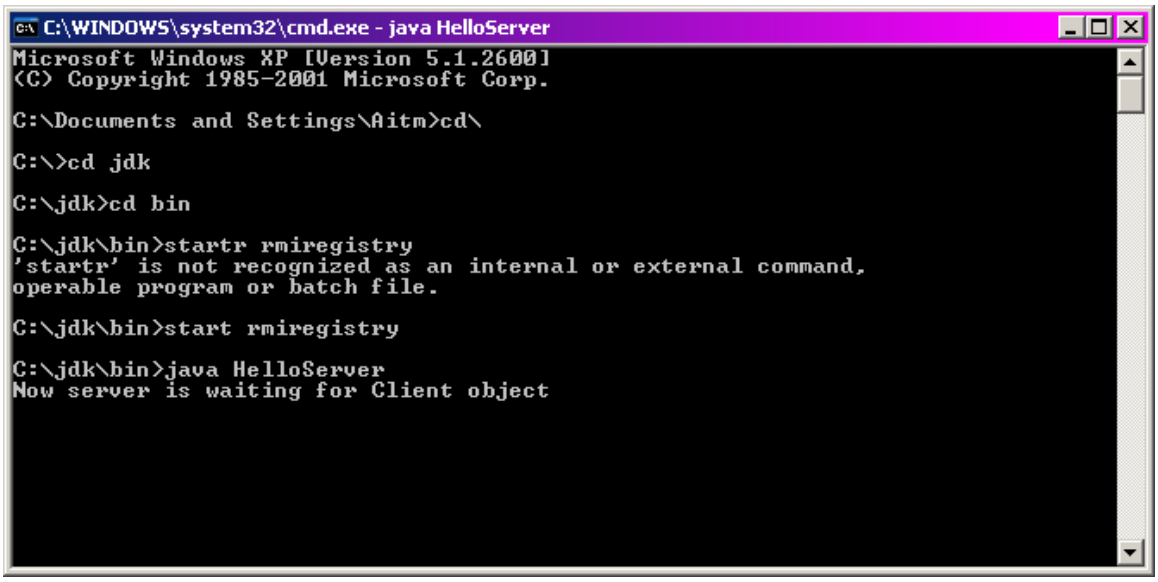

Permissions
Add New Permission:
Permission:
Permission:
AllPermission
AWTPermission
FilePermission
NetPermission
PropertyPermission
ReflectPermission
RuntimePermission
Cancel

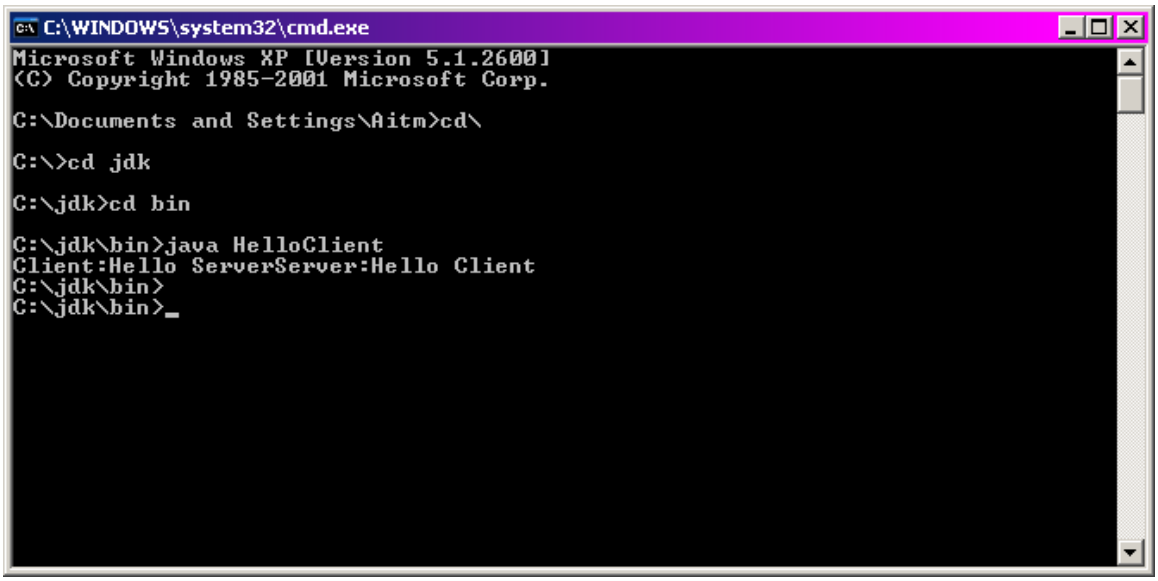

Policy Tool
File
Edit
New
Open
Save
Save As
View Warning Log
Exit
ments and Settings\Aitm\.java.policy
y Entry
Edit Policy Entry
Remove Policy Entry
CodeBase <ALL>
CodeBase <ALL>


Save As
Save in:
Aitm
Cookies
Desktop
Favorites
My Documents
Start Menu
.appletviewer
.java.policy
File name:
.java.policy
Save
Save as type:
All Files (*.*)
Cancel


C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Aitm>cd\
C:\>cd jdk
C:\jdk>cd bin
C:\jdk\bin>startr rmiregistry
'startr' is not recognized as an internal or external command,
operable program or batch file.
C:\jdk\bin>start rmiregistry
C:\jdk\bin>


C:\jdk\bin\rmiregistry.exe

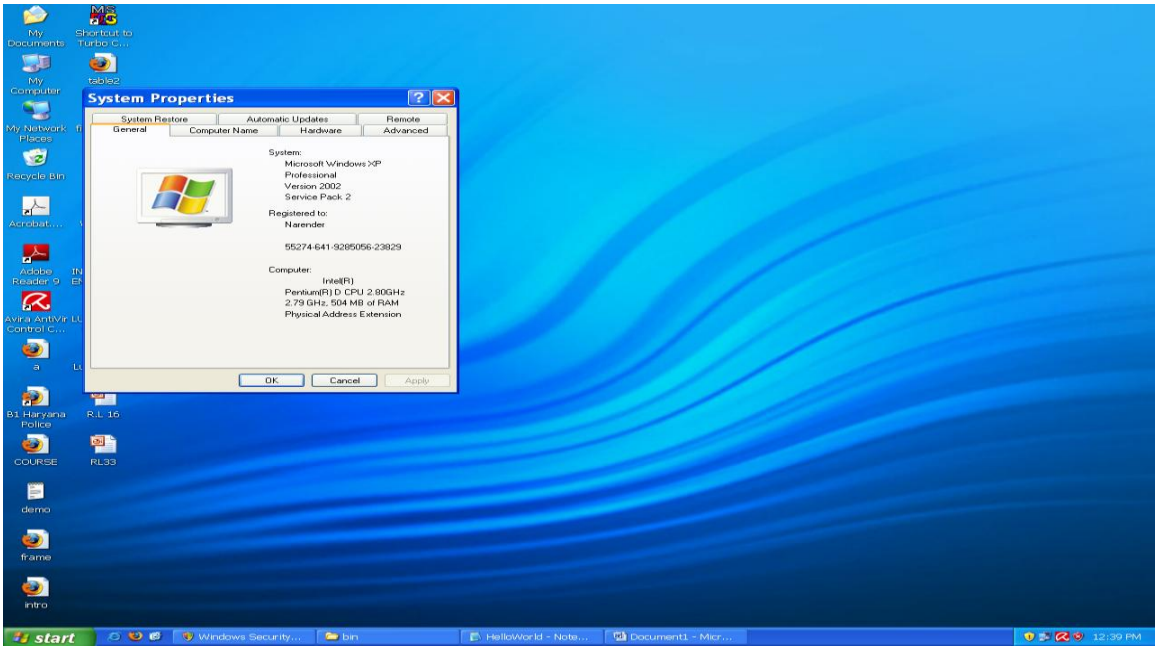

C:\WINDOWS\system32\cmd.exe - java HelloServer
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Aitm>cd\
C:\>cd jdk
C:\jdk>cd bin
C:\jdk\bin>startr rmiregistry
'startr' is not recognized as an internal or external command,
operable program or batch file.
C:\jdk\bin>start rmiregistry
C:\jdk\bin>java HelloServer
Now server is waiting for Client object

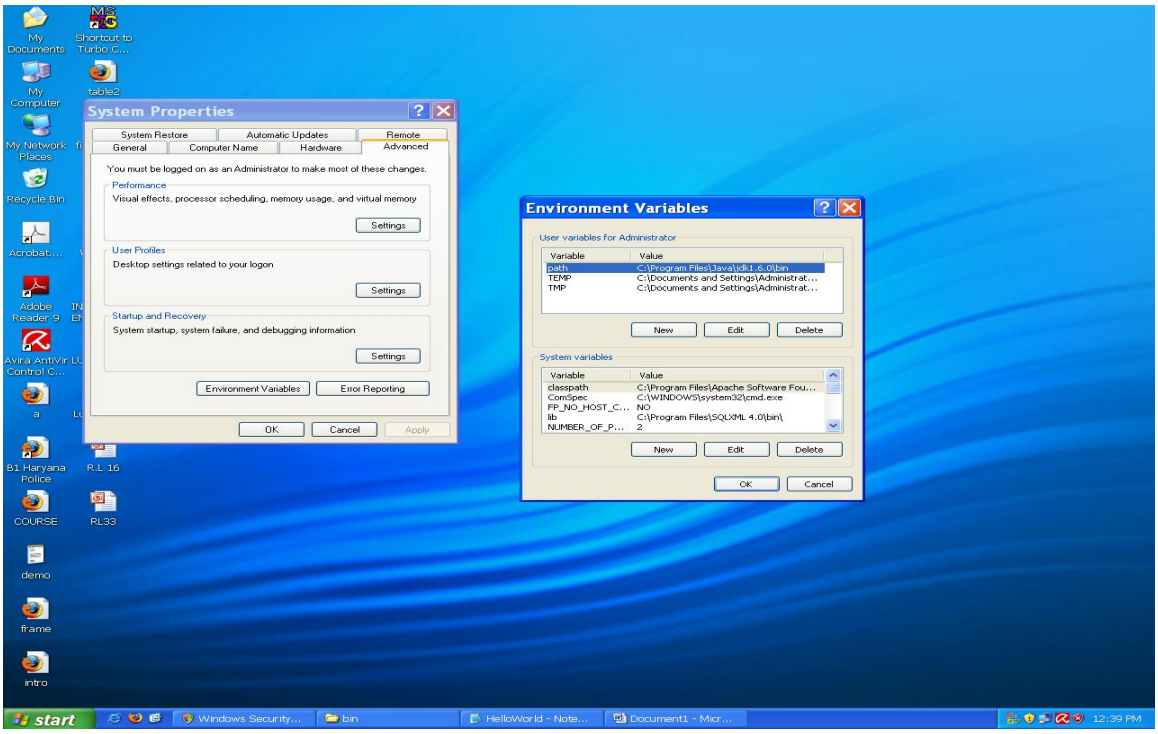

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Aitm>cd\
C:\>cd jdk
C:\jdk>cd bin
C:\jdk\bin>java HelloClient
Client:Hello ServerServer:Hello Client
C:\jdk\bin>
C:\jdk\bin>

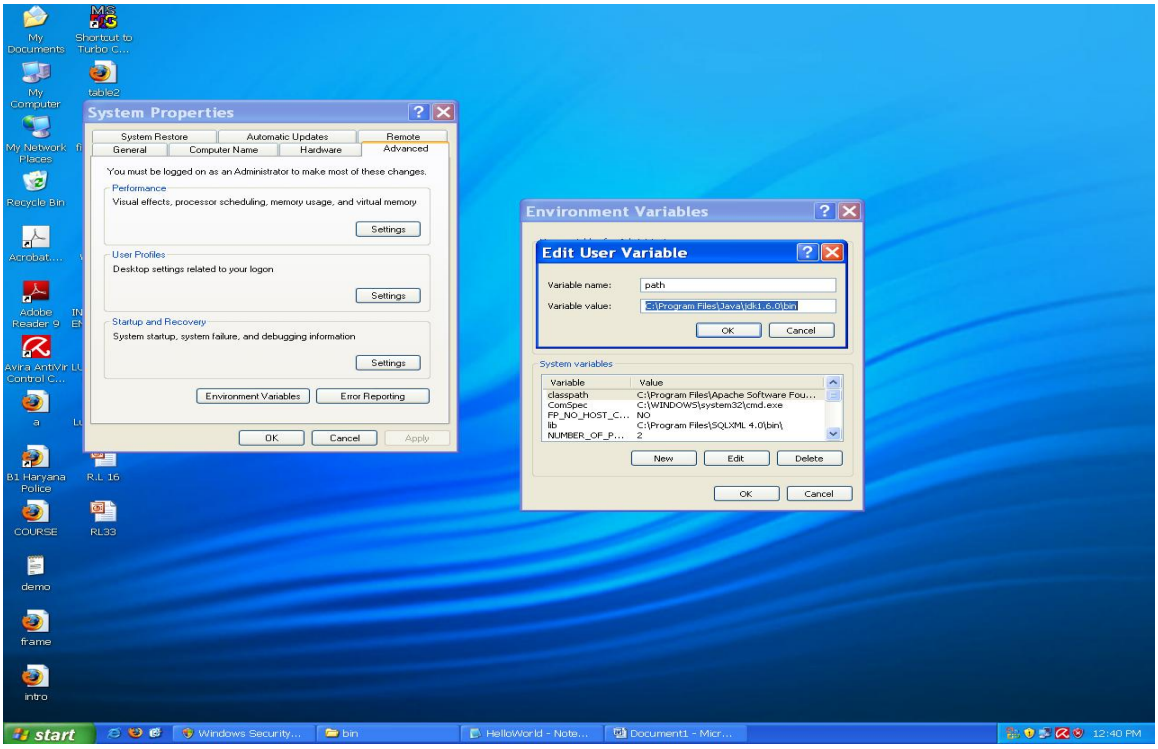
PROGRAM - 9

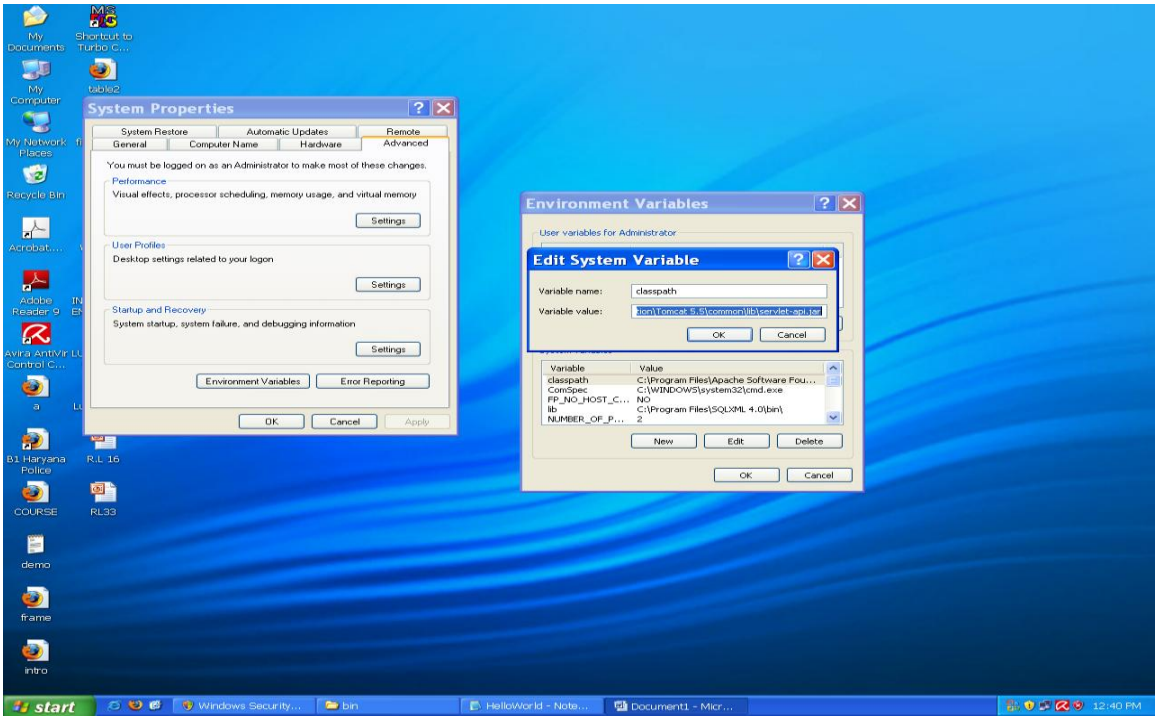
WAP TO IMPLEMENT SERVLETS

Step 1) Create a folder in Tomcat's webapps folder, say examples.

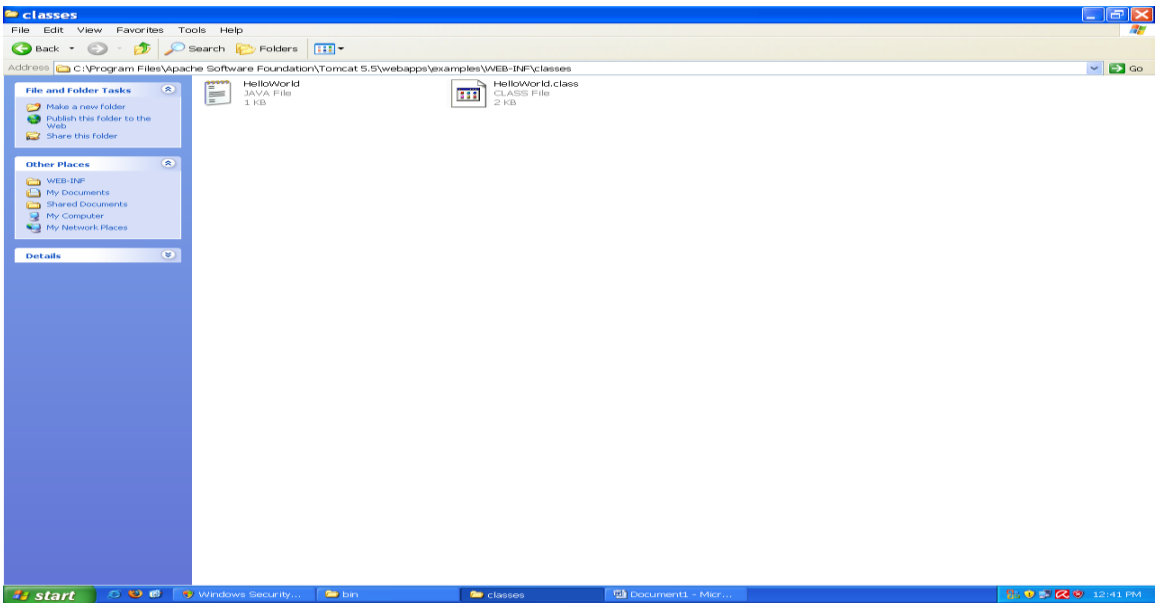



System Properties
System Restore
Automatic Updates
Remote
General
Computer Name
Hardware
Advanced
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings
User Profiles
Desktop settings related to your logon
Settings
Startup and Recovery
System startup, system failure, and debugging information
Settings
Environment Variables
Error Reporting
OK
Cancel
Apply
Environment Variables
User variables for Administrator
Variable
Value
path
C:\Program Files\Java\jdk1.6.0\bin
TEMP
C:\Documents and Settings\Administrat...
TMP
C:\Documents and Settings\Administrat...
New
Edit
Delete
System variables
Variable
Value
classpath
C:\Program Files\Apache Software Fou...
ComSpec
C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...
NO
lib
C:\Program Files\SQLXML 4.0\bin\
NUMBER_OF_P...
2
New
Edit
Delete
OK
Cancel
start
Windows Security...
bin
HelloWorld - Note...
Document1 - Micr...
12:39 PM


System Properties
System Restore
Automatic Updates
Remote
General
Computer Name
Hardware
Advanced
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings
User Profiles
Desktop settings related to your logon
Settings
Startup and Recovery
System startup, system failure, and debugging information
Settings
Environment Variables
Error Reporting
OK
Cancel
Apply
Environment Variables
Edit User Variable
Variable name:
path
Variable value:
C:\Program Files\Java\jdk1.6.0\bin
OK
Cancel
System variables
Variable
Value
classpath
C:\Program Files\Apache Software Fou...
ComSpec
C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...
NO
lib
C:\Program Files\SQLXML 4.0\bin\
NUMBER_OF_P...
2
New
Edit
Delete
OK
Cancel
start
Windows Security...
bin
HelloWorld - Note...
Document1 - Micr...
12:40 PM



Step 2) Create folders as examples -> WEB-INF -> classes



Step 3) Write the following program in a file "HelloWorld.java" and save in classes folder

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {

public HelloWorld() {
super();
}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
// TODO Auto-generated method stub
response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.println("<html>");
pw.println("<head><title>Hello World</title></title>");
pw.println("<body>");
pw.println("<h1>Hello World</h1>");
pw.println("</body></html>");
response.addHeader("Refresh", "1");
pw.println(new Date().toString());
```

```
}
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
// TODO Auto-generated method stub
}
}
```

Step 4) open the command prompt using cmd and set the drive location as:

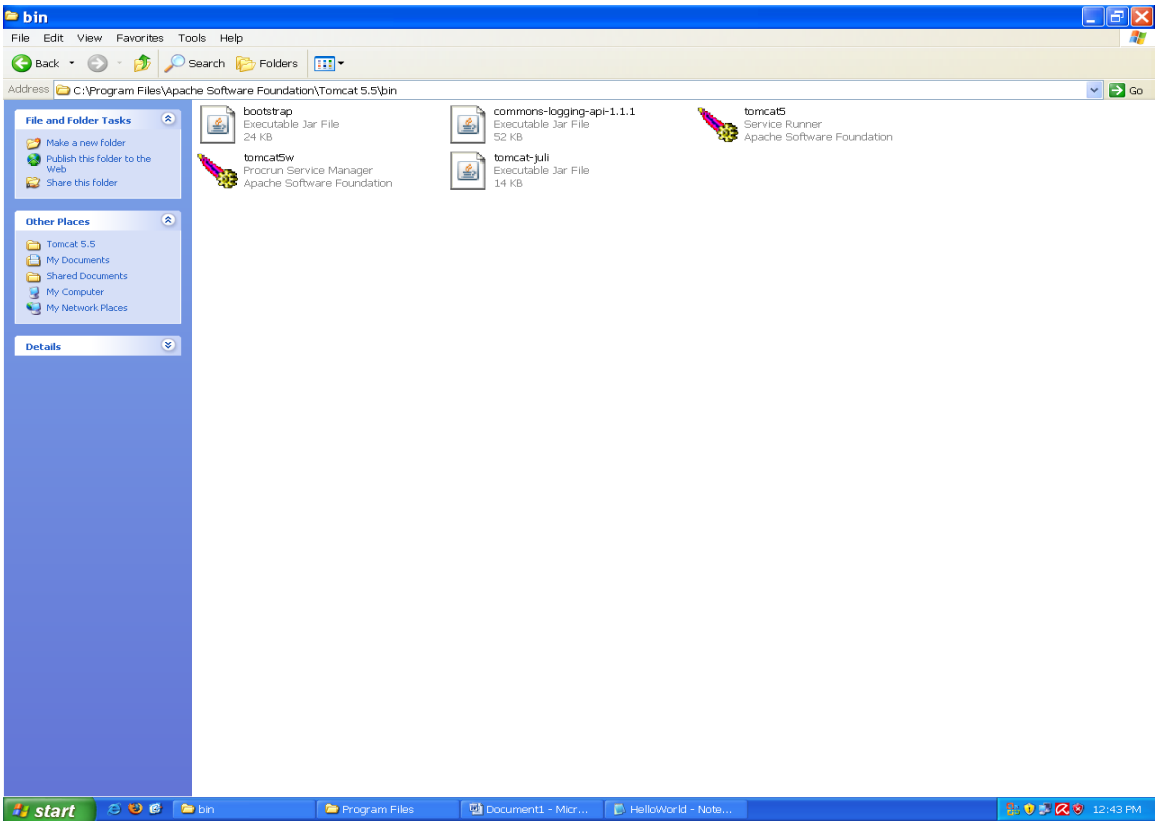
C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\examples\WEB-INF\classes.

Step 5) Compile the above program using "javac HelloWorld.java".

Step 6) Copy paste the web.xml file from webapps -> ROOT -> WEB-INF

To webapps -> examples -> WEB-INF

Step 7) Edit the web.xml file to set the servlet name and servlet-mappings according to your servlet.



Step 8) Start the Tomcat Service Runner in "C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin"

Step 9) Open the web browser and type the following URL to run the servlet from any client system:

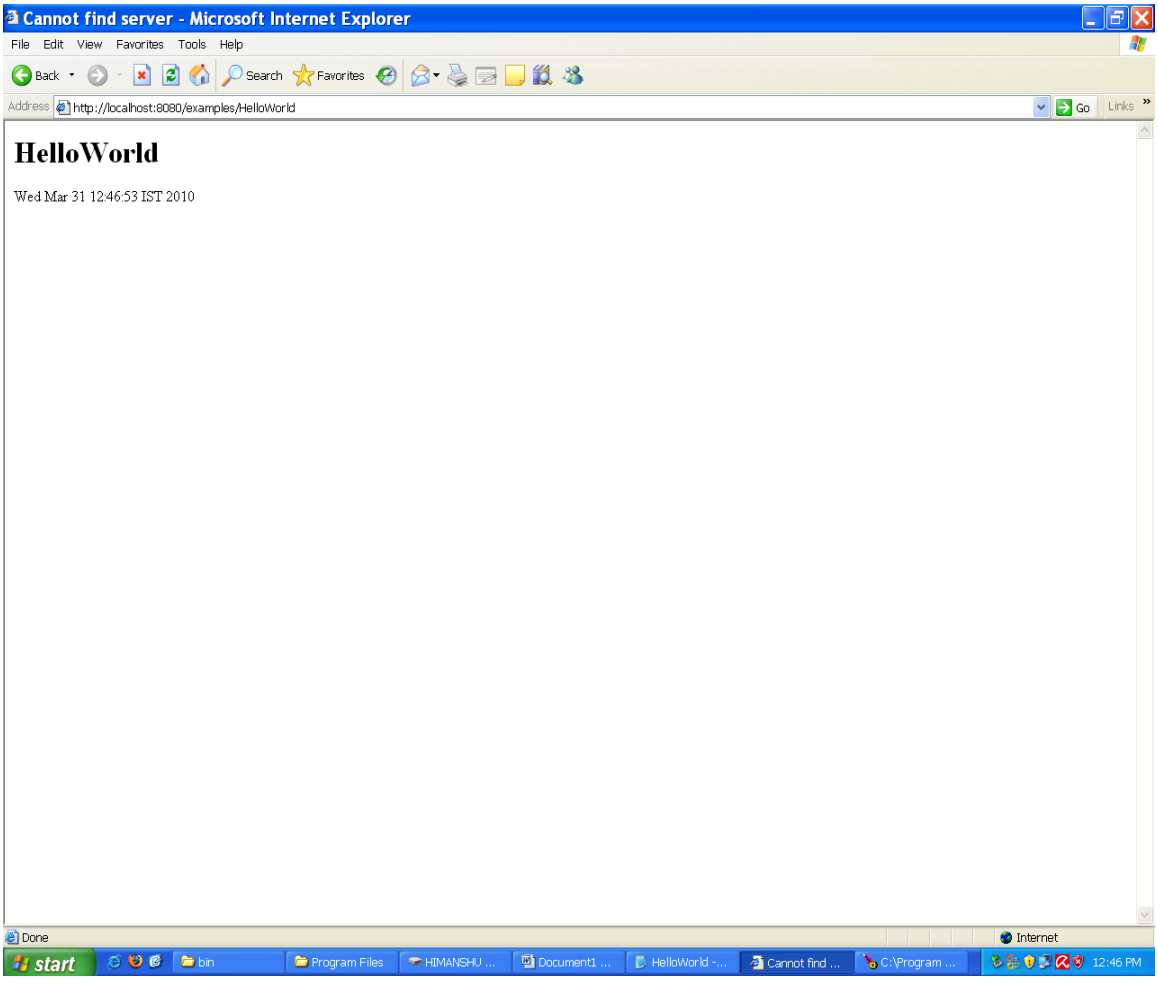
http://localhost:8080/examples/HelloWorld

or

http:// <IP address of server> /examples/HelloWorld

and press enter.

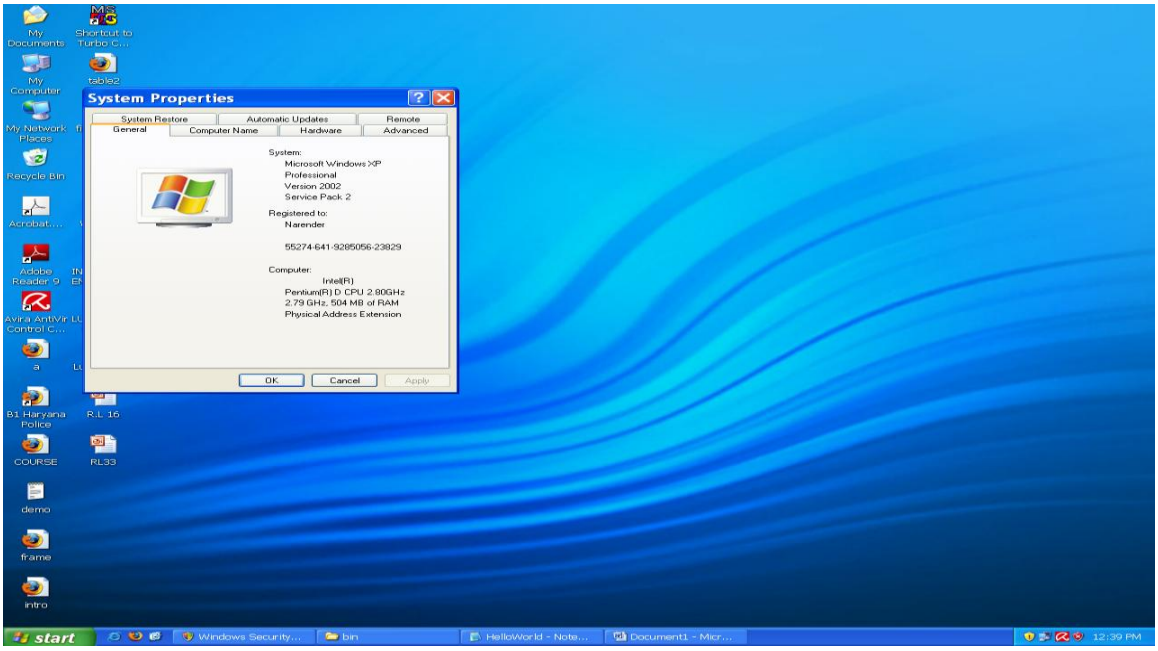
OUTPUT - 9

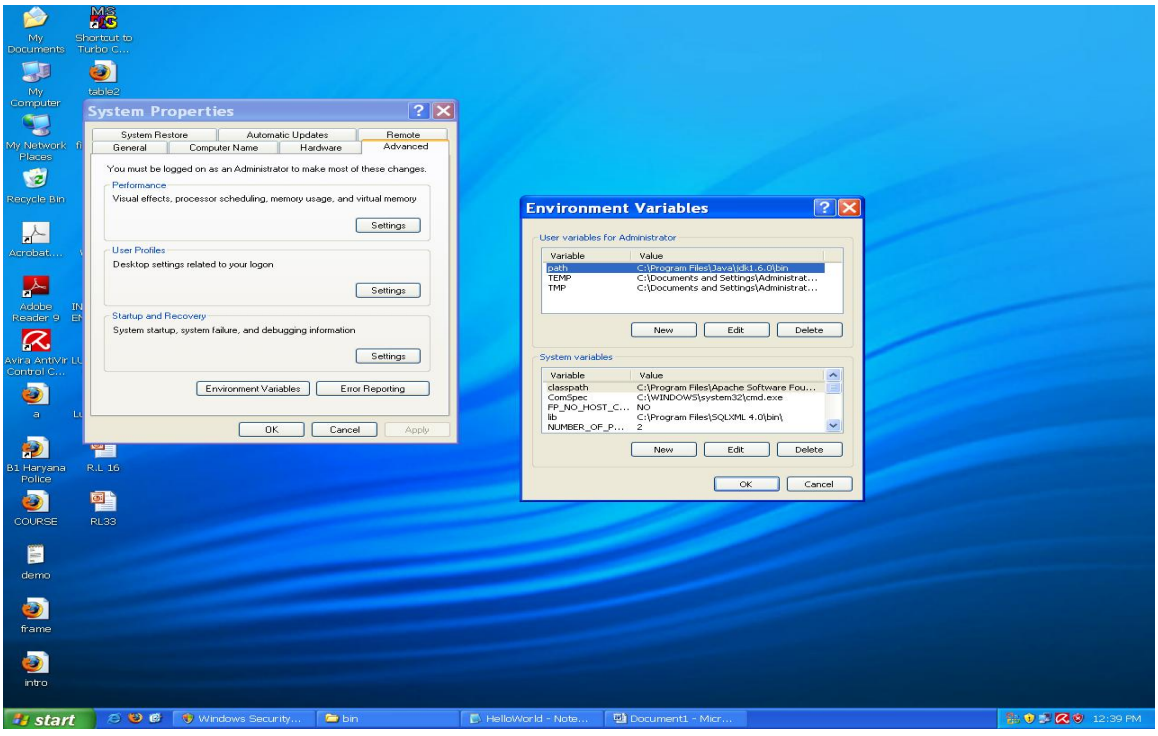

Cannot find server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/examples/HelloWorld
Go Links
HelloWorld
Wed Mar 31 12:46:53 IST 2010
Done
Internet
start
bin
Program Files
HIMANSHU ...
Document1 ...
HelloWorld -...
Cannot find ...
C:\Program ...
12:46 PM

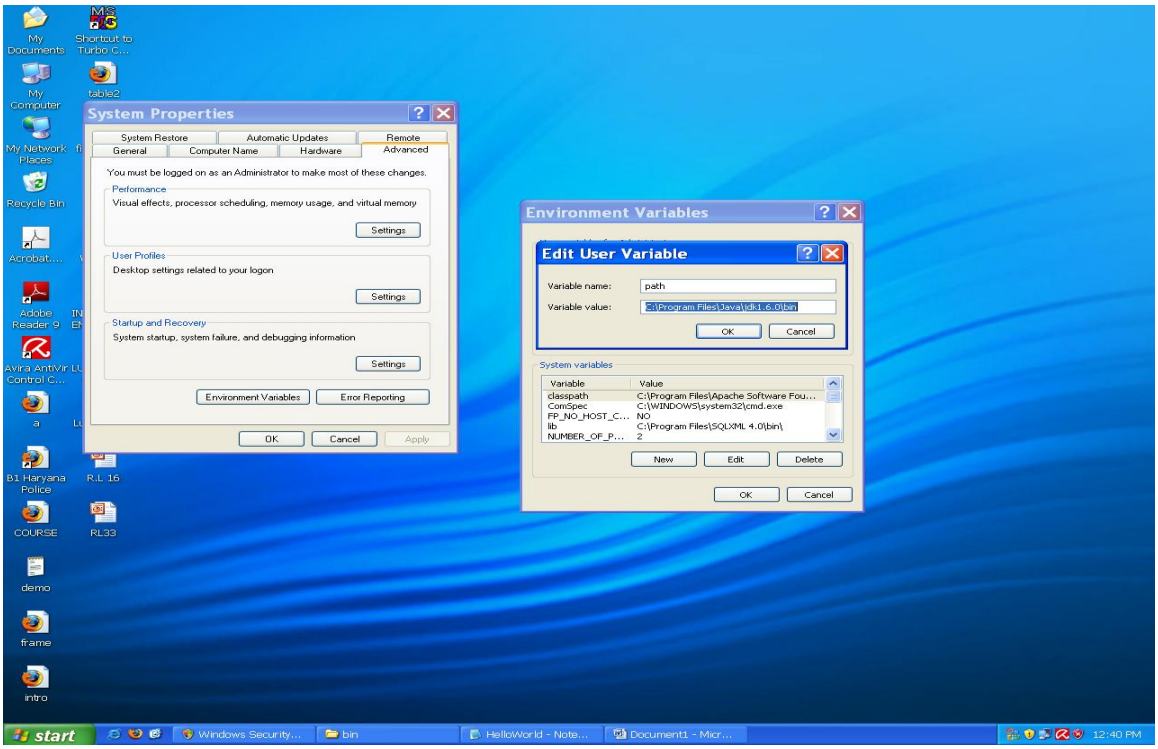
PROGRAM - 10

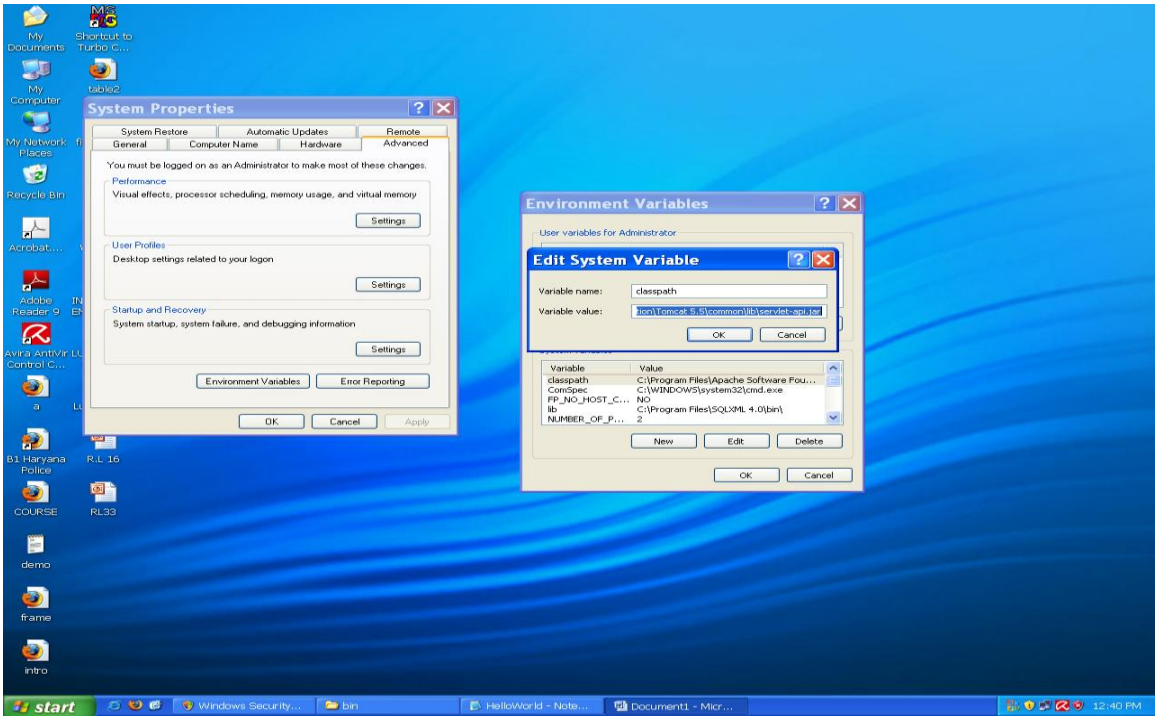
WAP TO IMPLEMENT JSP

Step 1) Create a folder in Tomcat's webapps folder, say examples.






System Properties
System Restore
Automatic Updates
Remote
General
Computer Name
Hardware
Advanced
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings
User Profiles
Desktop settings related to your logon
Settings
Startup and Recovery
System startup, system failure, and debugging information
Settings
Environment Variables
Error Reporting
OK
Cancel
Apply
Environment Variables
Edit User Variable
Variable name: path
Variable value: C:\Program Files\Java\jdk1.6.0\bin
OK
Cancel
System variables
Variable
Value
classpath
C:\Program Files\Apache Software Fou...
ComSpec
C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...
NO
lib
C:\Program Files\SQLXML 4.0\bin\
NUMBER_OF_P...
2
New
Edit
Delete
OK
Cancel
start
Windows Security...
bin
HelloWorld - Note...
Document1 - Micr...
12:40 PM


System Properties
System Restore
Automatic Updates
Remote
General
Computer Name
Hardware
Advanced
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings
User Profiles
Desktop settings related to your logon
Settings
Startup and Recovery
System startup, system failure, and debugging information
Settings
Environment Variables
Error Reporting
OK
Cancel
Apply
Environment Variables
User variables for Administrator
Edit System Variable
Variable name: classpath
Variable value: tion\Tomcat 5.5\common\lib\servlet-api.jar
OK
Cancel
Variable
Value
classpath
C:\Program Files\Apache Software Fou...
ComSpec
C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...
NO
lib
C:\Program Files\SQLXML 4.0\bin\
NUMBER_OF_P...
2
New
Edit
Delete
OK
Cancel
start
Windows Security...
bin
HelloWorld - Note...
Document1 - Micr...
12:40 PM

Step 2) Write the following program and save like an html file, with extension jsp, say "time.jsp" in the above created folder

```
<html>


<head>

<title>

Current TIme

</title>

</head>

<body>


<table align='center' >


<tr>

<td>

<b>Current Date and Time</b>

</td>

<td>

<%=new java.util.Date()%>

</td>

</tr>

</table>


</body>


</html>
```

Step 3) Start the Tomcat Service Runner in "C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin"

Step 4) Open the web browser and type the following URL to run the servlet from any client system:

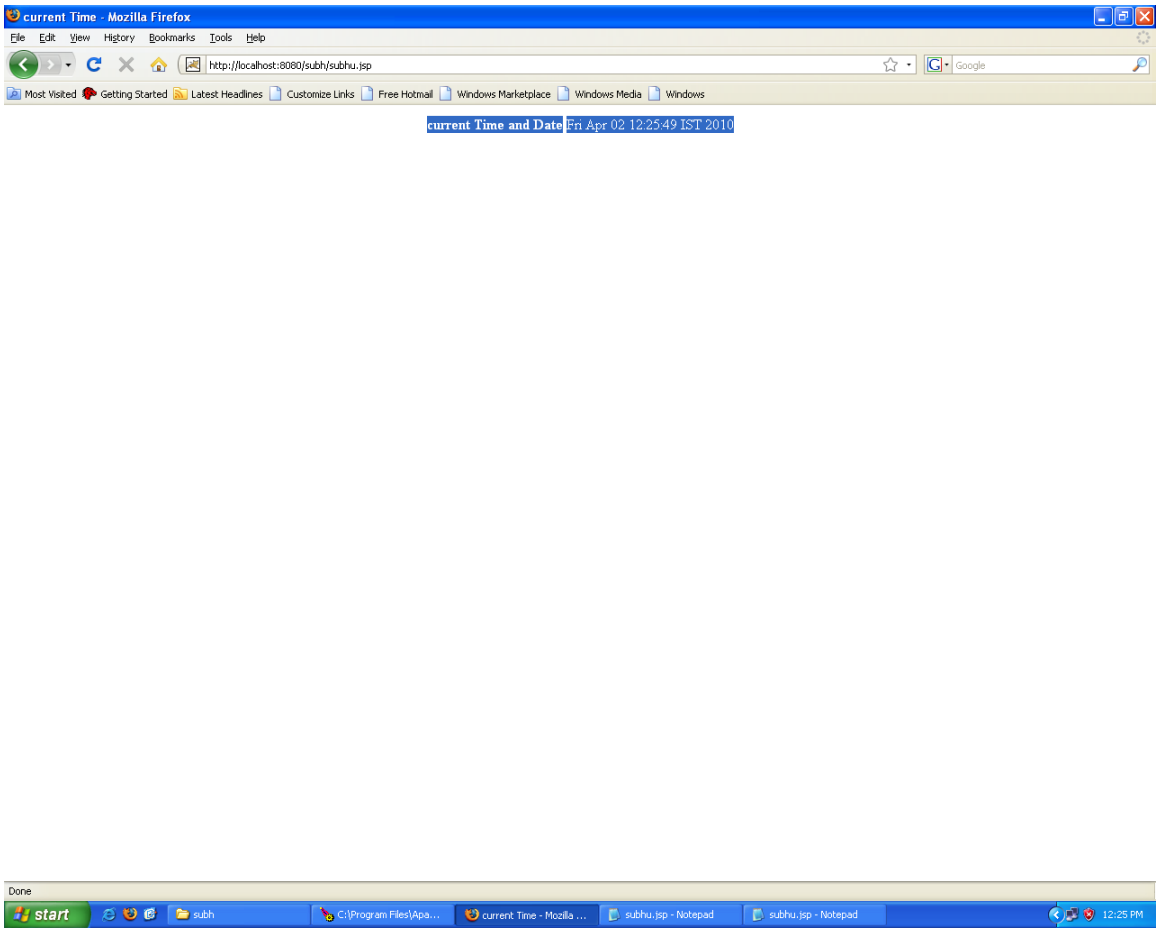
http://localhost:8080/examples/time.jsp

or

http:// <IP address of server> /examples/time.jsp

and press enter.

OUTPUT - 10


current Time - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/subh/subhu.jsp
Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Media Windows
current Time and Date Fri Apr 02 12:25:49 IST 2010
Done
start
subh
C:\Program Files\Apa...
current Time - Mozilla ...
subhu.jsp - Notepad
subhu.jsp - Notepad
12:25 PM