Practical No: 01

Aim: Write a program for implementing Client Server communication model using TCP.

Practical 1A: A client server based program using TCP to find if the number entered is prime.

Code:

1. **tcpServerPrime.java**

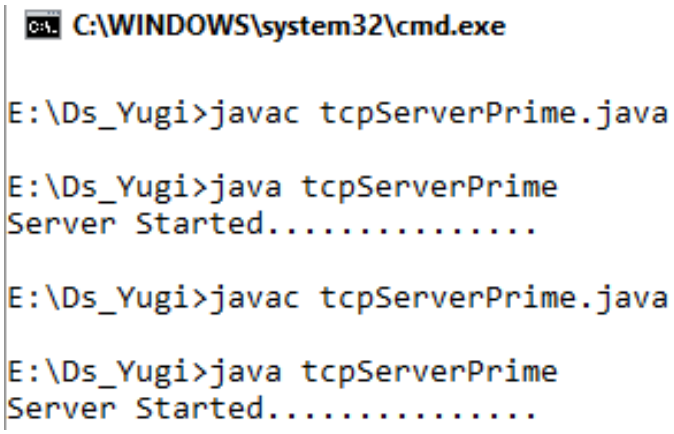
```
import java.net.*;
import java.io.*;

class tcpServerPrime

{
	public static void main(String args[])
	{
		try
		{	ServerSocket ss = new ServerSocket(8001);
			System.out.println("Server Started.		");
			Socket s = ss.accept();

			DataInputStream in = new
			DataInputStream(s.getInputStream()); int x= in.readInt();

			DataOutputStream otc = new
			DataOutputStream(s.getOutputStream()); int y = x/2;

			if(x ==1 || x ==2 || x ==3)
			{
				otc.writeUTF(x + "is Prime");
				System.exit(0);
			}
			for(int i=2; i<=y; i++)
			{
				if(x%i != 0)
				{
					otc.writeUTF(x + " is Prime");
				}
				else
				{
					otc.writeUTF(x + " is not Prime");
				}

			}
		}
		catch(Exception e)
```

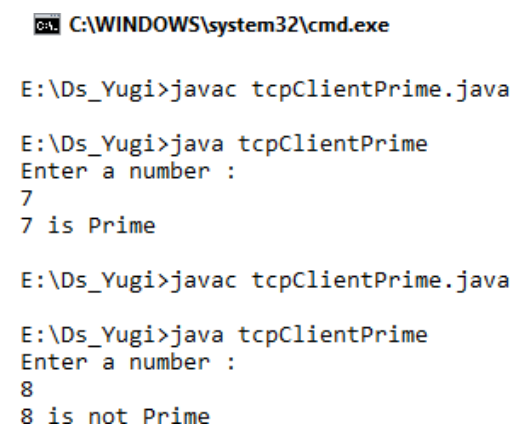
```
		{
			System.out.println(e.toString());
		}
	}
}
```

2. **tcpClientPrime.java**

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

class tcpClientPrime
{
	public static void main(String args[])
	{
		try
		{
			Socket cs = new Socket("LocalHost",8001);
			BufferedReader infu = new BufferedReader(new
InputStreamReader(System.in));
			System.out.println("Enter a number : ");

			int a = Integer.parseInt(infu.readLine());
			DataOutputStream out = new
DataOutputStream(cs.getOutputStream());
			out.writeInt(a);

			DataInputStream in = new
			DataInputStream(cs.getInputStream());
			System.out.println(in.readUTF()); cs.close();

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

Output:





Practical 1B: A client server TCP based chatting application.

Code:

1. **ChatServer.java**

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

class ChatServer
{
        public static void main(String args[])
        {
                try
                {
                        ServerSocket ss = new ServerSocket(8000);
                        System.out.println("Waiting for client to
                        connect.."); Socket s = ss.accept();

                        BufferedReader br = new
BufferedReader(new InputStreamReader(System.in));
```

```
                        DataOutputStream out = new
                        DataOutputStream(s.getOutputStream()); DataInputStream in = new
                        DataInputStream(s.getInputStream()); String receive, send;

                        while((receive = in.readLine()) != null)
                        {
                                if(receive.equals("STOP"))

                                        break;
                                System.out.println("Client Says : "+receive);
                                System.out.print("Server Says : ");
                                send = br.readLine();
                                out.writeBytes(send+"\n");
                        }
                        br.close();
                        in.close();
                        out.close();


                        s.close();
                }
                catch(Exception e)
                {
                        e.printStackTrace();
                }
        }
}
```

2. ChatClient.java

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

class ChatClient

{
        public static void main(String args[])
        {
                try
                {

                        Socket s = new Socket("Localhost",8000);
                        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

                        DataOutputStream out = new
                        DataOutputStream(s.getOutputStream()); DataInputStream in = new
                        DataInputStream(s.getInputStream()); String msg;

                        System.out.println("To stop chatting with server type
                        STOP"); System.out.print("Client Says: "); while((msg =
                        br.readLine()) != null)
```

```
                {
                        out.writeBytes(msg+"\n");
                        if(msg.equals("STOP"))
                                break;
                        System.out.println("Server Says : "+in.readLine());
                        System.out.print("Client Says : ");
                }
                br.close();
                in.close();
                out.close();
                s.close();
        }
        catch(Exception e)

        {
                e.printStackTrace();
        }
    }
}
```

Output:

Server:

```
E:\Ds_Yugi>java ChatServer
Waiting for client to connect..
Client Says : Hii...
Server Says : Hello
Client Says : How are you?
Server Says : Fine, thank you..
```

```
E:\Ds_Yugi>java ChatClient
To stop chatting with server type STOP
Client Says: Hii...
Server Says : Hello
Client Says : How are you?
Server Says : Fine, thank you..
Client Says : STOP
```

Practical No: 02

Aim: Write a program for implementing Client Server communication model using UDP.

Practical 2A: A client server based program using UDP to find if the number entered is even or odd.

Code:

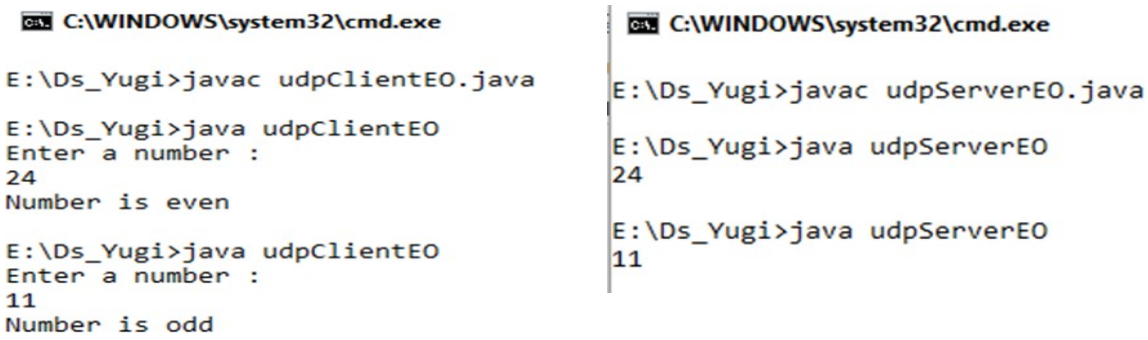
1. **udpServerEO.java**

```
/*Program which finds entered number is even or odd */
import java.io.*;
import java.net.*;


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


	{
		try
		{
			DatagramSocket ds = new DatagramSocket(2000);
			byte b[] = new byte[1024];

			DatagramPacket dp = new DatagramPacket(b,b.length);
			ds.receive(dp);

			String str = new
			String(dp.getData(),0,dp.getLength());
			System.out.println(str);

			int a= Integer.parseInt(str);
			String s= new String();
			if (a%2 == 0)
				s = "Number is even";
			else
				s = "Number is odd";

			byte b1[] = new byte[1024];
			b1 = s.getBytes();
			DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);
			ds.send(dp1);
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
	}
}
```

2. **udpClientEO.java**

```
/*Program which finds entered number is even or odd*/

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

public class udpClientEO
{
	public static void main(String args[])
	{
		try
		{

			DatagramSocket ds = new DatagramSocket(1000);
			BufferedReader br = new BufferedReader(new

InputStreamReader(System.in));
			System.out.println("Enter a number : ");
			String num = br.readLine();
			byte b[] = new byte[1024];
			b=num.getBytes();
			DatagramPacket dp = new
DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);


			ds.send(dp);
			byte b1[] = new byte[1024];

			DatagramPacket dp1 = new
			DatagramPacket(b1,b1.length); ds.receive(dp1);

			String str = new
			String(dp1.getData(),0,dp1.getLength());
			System.out.println(str);

		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
	}
}
```

Output:



Practical 2B: A client server based program using UDP to find the factorial of the entered number.

Code:

1. **udpServerFact.java**

```
/*Program which calculate factorial of a number*/
import java.io.*;
import java.net.*;

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

		try
		{
			DatagramSocket ds = new DatagramSocket(2000);
			byte b[] = new byte[1024];

			DatagramPacket dp = new DatagramPacket(b,b.length);
			ds.receive(dp);

			String str = new
			String(dp.getData(),0,dp.getLength());
			System.out.println(str);

			int a= Integer.parseInt(str);
			int f = 1, i;
			String s= new String();
			for(i=1;i<=a;i++)
			{
				f=f*i;
			}
			s=Integer.toString(f);
```

```
                String str1 = "The Factorial of " + str + " is : " +
                f; byte b1[] = new byte[1024]; b1 =
                str1.getBytes();

                DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);
                ds.send(dp1);
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }

        }
}
```

2. **udpClientFact.java**

```
/*Program which calculate factorial of a number*/

import java.io.*;


import java.net.*;

public class udpClientFact
{
        public static void main(String args[])
        {
            try
            {

                DatagramSocket ds = new DatagramSocket(1000);
                BufferedReader br = new BufferedReader(new

InputStreamReader(System.in));
                System.out.println("Enter a number : ");
                String num = br.readLine();
                byte b[] = new byte[1024];
                b=num.getBytes();

                DatagramPacket dp = new
DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);
                ds.send(dp);
                byte b1[] = new byte[1024];

                DatagramPacket dp1 = new DatagramPacket(b1,b1.length);
                ds.receive(dp1);

                String str = new
                String(dp1.getData(),0,dp1.getLength());
                System.out.println(str);

            }
            catch(Exception e)
            {
```

```
                    e.printStackTrace();
                }
            }
        }
```

Output:

```
C:\WINDOWS\system32\cmd.exe

E:\Ds_Yugi>javac udpServerFact.java

E:\Ds_Yugi>java udpServerFact
9

E:\Ds_Yugi>java udpServerFact
5
```

```
C:\WINDOWS\system32\cmd.exe

E:\Ds_Yugi>javac udpClientFact.java

E:\Ds_Yugi>java udpClientFact
Enter a number :
9
The Factorial of 9 is : 362880

E:\Ds_Yugi>java udpClientFact
Enter a number :
5
The Factorial of 5 is : 120
```

Practical 2C: A program to implement simple calculator operations like addition, subtraction, multiplication and division.

Code:

1. **RPCServer.java**

```
import java.util.*;
import java.net.*;

class RPCServer

{
	DatagramSocket ds;
	DatagramPacket dp;
	String str,methodName,result;
	int val1,val2;

	RPCServer()
	{
		try
		{	ds=new DatagramSocket(1200);
			byte b[]=new byte[4096];

			while(true)
			{
				dp=new DatagramPacket(b,b.length);
				ds.receive(dp);
				str=new String(dp.getData(),0,dp.getLength());

				if(str.equalsIgnoreCase("q"))
				{
					System.exit(1);
				}
				else
				{

					StringTokenizer st = new StringTokenizer(str,"
					"); int i=0;

					while(st.hasMoreTokens())
```

```
                {
                        String token=st.nextToken();
                        methodName=token;
                        val1 = Integer.parseInt(st.nextToken());
                        val2 = Integer.parseInt(st.nextToken());
                }


                }
                System.out.println(str);
                InetAddress ia = InetAddress.getLocalHost();

                if(methodName.equalsIgnoreCase("add"))
                {
                        result= "" + add(val1,val2);
                }
                else if(methodName.equalsIgnoreCase("sub"))
                {
                        result= "" + sub(val1,val2);
                }
                else if(methodName.equalsIgnoreCase("mul"))
                {
                        result= "" + mul(val1,val2);

                }
                else if(methodName.equalsIgnoreCase("div"))
                {
                        result= "" + div(val1,val2);
                }
                byte b1[]=result.getBytes();

                DatagramSocket ds1 = new DatagramSocket();
                DatagramPacket dp1 = new

DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300);
                System.out.println("result :
                "+result+"\n"); ds1.send(dp1);

                }
        }
        catch (Exception e)
        {
                e.printStackTrace();
        }
}

public int add(int val1, int val2)
{
        return val1+val2;
}
public int sub(int val3, int val4)
```

```
	{
		return val3-val4;
	}

	public int mul(int val3, int val4)
	{
		return val3*val4;
	}
	public int div(int val3, int val4)
	{
		return val3/val4;
	}



	public static void main(String[] args)
	{
		new RPCServer();
	}
}
```

2. **RPCClient.java**

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

class RPCClient
{
	RPCClient()

	{
		try
		{	InetAddress ia = InetAddress.getLocalHost();
			DatagramSocket ds = new DatagramSocket();
			DatagramSocket ds1 = new DatagramSocket(1300);
			System.out.println("\nRPC Client\n");
			System.out.println("Enter method name and parameter like add 3

4\n");			while (true)
			{

				BufferedReader br = new
BufferedReader(new InputStreamReader(System.in));

				String str = br.readLine();
				byte b[] = str.getBytes();
				DatagramPacket dp = new
DatagramPacket(b,b.length,ia,1200);
				ds.send(dp);
				dp = new DatagramPacket(b,b.length);
				ds1.receive(dp);
```

```
                        String s = new String(dp.getData(),0,dp.getLength());
                        System.out.println("\nResult = " + s + "\n");
                    }
                }
                catch (Exception e)
                {
                    e.printStackTrace();

                }
        }

        public static void main(String[] args)
        {
            new RPCClient();
        }
}
```

Output:

```
C:\WINDOWS\system32\cmd.exe - java RPCClient

E:\Ds_Yugi>javac RPCClient.java

E:\Ds_Yugi>java RPCClient

RPC Client

Enter method name and parameter like add 34

sub 10 8

Result = 2

mul 27 2

Result = 54

add 20 7

Result = 27

div 10 2
```

```
C:\WINDOWS\system32\cmd.exe - java RPCServer

E:\Ds_Yugi>javac RPCServer.java

E:\Ds_Yugi>java RPCServer
sub 10 8
result : 2

mul 27 2
result : 54

add 20 7
result : 27

div 10 2
result : 5
```

Practical 2D: A program that finds the square, square root, cube and cube root of the entered number.

Code:

1. **RPCNumServer.java**

```
import java.util.*;
import java.net.*;
import  java.io.*;
class RPCNumServer
```

```
{
	DatagramSocket ds;

	DatagramPacket dp;
	String str,methodName,result;
	int val;
	RPCNumServer()
	{
		try
		{	ds=new DatagramSocket(1200);
			byte b[]=new byte[4096];
			while(true)
			{
				dp=new DatagramPacket(b,b.length);
				ds.receive(dp);

				str=new
				String(dp.getData(),0,dp.getLength());
				if(str.equalsIgnoreCase("q")) {

					System.exit(1);
				}
				else
				{	StringTokenizer st = new StringTokenizer(str," ");
					int i=0;
					while(st.hasMoreTokens())
					{
						String token=st.nextToken();
						methodName=token;
						val = Integer.parseInt(st.nextToken());
					}
				}
				System.out.println(str);
				InetAddress ia = InetAddress.getLocalHost();
				if(methodName.equalsIgnoreCase("square"))
				{
					result= "" + square(val);
				}
				else if(methodName.equalsIgnoreCase("squareroot"))
				{
					result= "" + squareroot(val);
				}
				else if(methodName.equalsIgnoreCase("cube"))
				{
					result= "" + cube(val);

				}
				else if(methodName.equalsIgnoreCase("cuberoot"))
				{
```

```
                                        result= "" + cuberoot(val);
                                }
                                byte b1[]=result.getBytes();
                                DatagramSocket ds1 = new DatagramSocket();


                                DatagramPacket dp1 = new

DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300);
                                System.out.println("result :
                                "+result+"\n"); ds1.send(dp1);

                        }
                }
                catch (Exception e)
                {
                        e.printStackTrace();
                }
        }
        public double square(int a) throws Exception
        {
                double ans;
                ans = a*a;

                return ans;
        }
        public double squareroot(int a) throws Exception
        {
                double ans;
                ans = Math.sqrt(a);
                return ans;
        }
        public double cube(int a) throws Exception
        {
                double ans;
                ans = a*a*a;
                return ans;
        }
        public double cuberoot(int a) throws Exception
        {
                double ans;
                ans = Math.cbrt(a);
                return ans;
        }
        public static void main(String[] args)
        {
                new RPCNumServer();
        }
}
```

2. **RPCNumClient.java**

```
import java.io.*;
import java.net.*;
class RPCNumClient
{
	RPCNumClient()
	{
		try


		{

			InetAddress ia = InetAddress.getLocalHost();
			DatagramSocket ds = new DatagramSocket();
			DatagramSocket ds1 = new DatagramSocket(1300);
			System.out.println("\nRPC Client\n");

			System.out.println("1. Square of the number - square\n2. Square root
of the number - squareroot\n3. Cube of the number - cube\n4. Cube root of the number -
cuberoot");

			System.out.println("Enter method name and the number\n");
			while (true)
			{

				BufferedReader br = new
BufferedReader(new InputStreamReader(System.in));

				String str = br.readLine();
				byte b[] = str.getBytes();

				DatagramPacket dp = new
DatagramPacket(b,b.length,ia,1200);
				ds.send(dp);
				dp = new DatagramPacket(b,b.length);
				ds1.receive(dp);

				String s = new String(dp.getData(),0,dp.getLength());
				System.out.println("\nResult = " + s + "\n");
			}
		}
		catch (Exception e)
		{
			e.printStackTrace();
		}
	}
	public static void main(String[] args)
	{
		new RPCNumClient();
	}
```

}

Output:

```
C:\WINDOWS\system32\cmd.exe - java RPCNumServer

E:\Ds_Yugi>javac RPCNumServer.java

E:\Ds_Yugi>java RPCNumServer
square 7
result : 49.0

squareroot 25
result : 5.0

cube 3
result : 27.0

cuberoot 27
result : 3.0
```

```
C:\WINDOWS\system32\cmd.exe - java RPCNumClient

1. Square of the number - square
2. Square root of the number - squareroot
3. Cube of the number - cube
4. Cube root of the number - cuberoot
Enter method name and the number

square 7

Result = 49.0

squareroot 25

Result = 5.0

cube 3

Result = 27.0

cuberoot 27

Result = 3.0
```

Practical No: 03

Aim: A multicast Socket example.

Code:

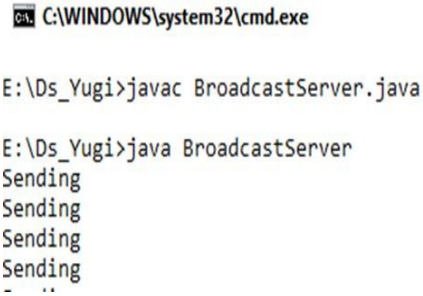
1. **BroadcastServer.java**

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

public class BroadcastServer
{
        public static final int PORT = 1234;

        public static void main(String args[])throws
        Exception {

                MulticastSocket  socket;
                DatagramPacket packet;
                InetAddress address;

                // set the multicast address to your local subnet
                address = InetAddress.getByName("239.1.2.3");
                socket = new MulticastSocket();

                // join a Multicast group and send the group
                messages socket.joinGroup(address);

                byte[] data =
                null; for(;;)

                {

                        Thread.sleep(10000);
                        System.out.println("Sending "); String
                        str = ("This is Neha Calling    "); data
                        = str.getBytes();

                        packet = new DatagramPacket(data, str.length(),address,PORT);

                        // Sends the packet
                        socket.send(packet);
                } // end for
        } // end main
} // end class BroadcastServer
```

2. **BroadcastClient.java**

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

public class BroadcastClient
{
        public static final int PORT = 1234;

        public static void main(String args[])throws
        Exception {

                MulticastSocket  socket;
                DatagramPacket packet;
                InetAddress address;

                // set the mulitcast address to your local subnet
                address = InetAddress.getByName("239.1.2.3");
                socket = new MulticastSocket(PORT);

                //join a Multicast group and wait for a
                message socket.joinGroup(address); byte[]
                data = new byte[100];

                packet = new DatagramPacket(data,data.length);

                for(;;)
                {
                        // receive the packets
                        socket.receive(packet);

                        String str = new String(packet.getData()); System.out.println("Message
                        received from "+ packet.getAddress() + "
Message is : "+str);
                } // for
        } // main
                        } // end BroadcastClient
```

Output:



```
E:\Ds_Yugi>javac BroadcastClient.java

E:\Ds_Yugi>java BroadcastClient

Message received from /10.29.26.232 Message is: This is Neha Calling....

Message received from /10.29.26.232 Message is: This is Neha Calling....

Message received from /10.29.26.232 Message is: This is Neha Calling....

Message received from /10.29.26.232 Message is: This is Neha Calling....
```

Practical No: 04

Aim: Write a program to show the object communication using RMI.

Practical 4A: A RMI based application program to display current date and time.

Code:

1. **InterDate.java**

```
import java.rmi.*;

public interface InterDate extends Remote
{
	public String display() throws Exception;

}
```

2. **ServerDate.java**

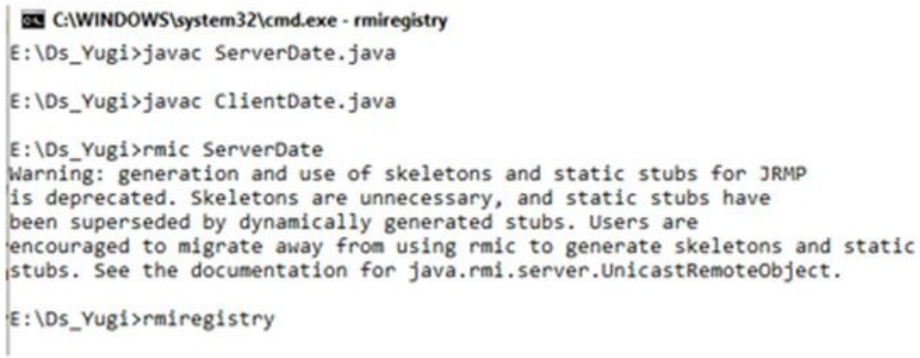
```
import java.rmi.*;
import java.rmi.server.*;
import java.util.*;

public class ServerDate extends UnicastRemoteObject implements
InterDate {

	public ServerDate() throws Exception
	{
	}

	public String display() throws Exception
	{
		String str = "";
		Date d = new Date();
		str = d.toString();
		return str;
```

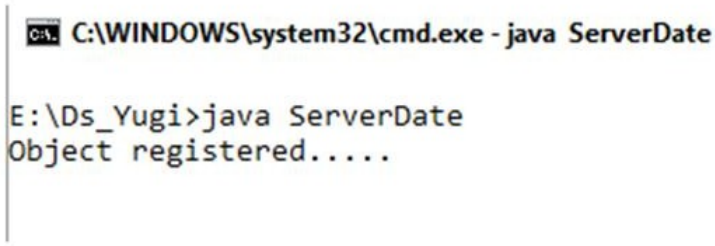
```
    }

    public static void main(String args[]) throws
    Exception {

        ServerDate s1 = new ServerDate();
        Naming.bind("DS",s1);
        System.out.println("Object registered.    ");

    }
}
```

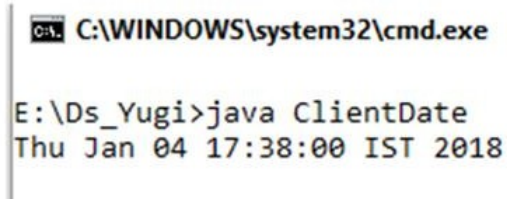
3. **ClientDate.java**

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



public class ClientDate
{
        public static void main(String args[]) throws
        Exception {

                String s1;
                InterDate h1 = (InterDate)Naming.lookup("DS");
                s1 = h1.display();
                System.out.println(s1);
        }
}
```

Output:







Practical 4B: A RMI based application program that converts digits to words, e.g. 123 will be converted to one two three.

Code:

1. **InterConvert.java**

```
import java.rmi.*;

public interface InterConvert extends Remote
{
	public String convertDigit(String no) throws Exception;
}
```

1. **ServerConvert.java**

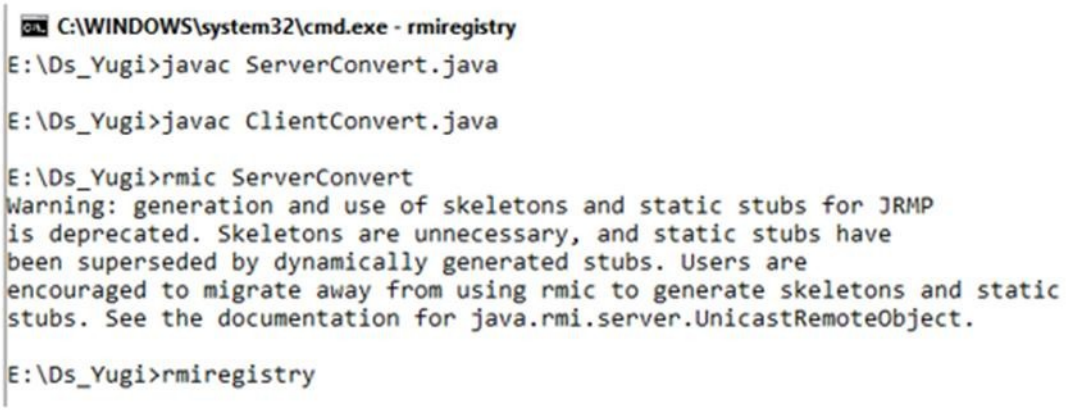
```
import java.rmi.*;
import java.rmi.server.*;

public class ServerConvert extends UnicastRemoteObject implements
InterConvert {

	public ServerConvert() throws Exception
	{
	}

	public String convertDigit(String no) throws Exception
	{

		String str = "";
		for(int i = 0; i < no.length(); i++)
		{
			int p = no.charAt(i);
			if( p == 48)
			{
				str += "zero ";
			}
			if( p == 49)
			{
				str += "one ";
			}
			if( p == 50)
			{
				str += "two ";

			}
			if( p == 51)
			{
				str += "three ";
			}
```

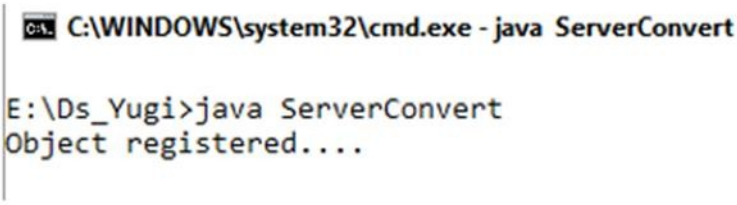
```
if( p == 52)
{
        str += "four ";


}
if( p == 53)
{
        str += "five ";
}
if( p == 54)
{
        str += "six ";
}
if( p == 55)
{
```

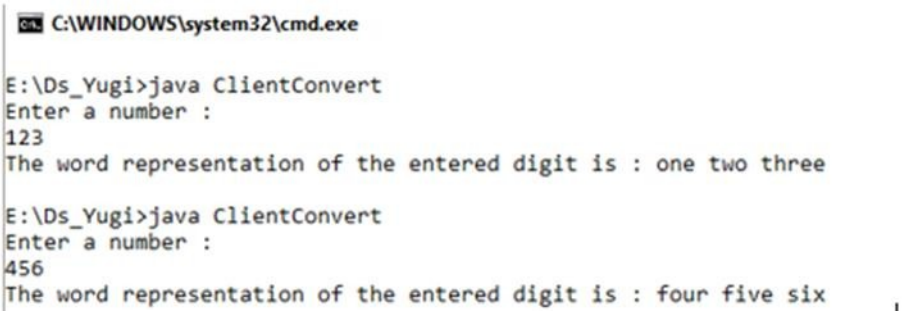
```
                    str += "seven ";
                }
                if( p == 56)
                {

                    str += "eight ";
                }
                if( p == 57)
                {
                    str += "nine ";
                }
            }
            return str;
        }

        public static void main(String args[]) throws
        Exception {

            ServerConvert s1 = new ServerConvert();
            Naming.bind("Wrd",s1);
            System.out.println("Object registered.   ");
        }
}
```

1. **ClientConvert.java**

```
import java.rmi.*;
import java.io.*;
public class ClientConvert
{

        public static void main(String args[]) throws
        Exception {

            InterConvert h1 =
            (InterConvert)Naming.lookup("Wrd"); BufferedReader
            br = new BufferedReader(new

InputStreamReader(System.in));
            System.out.println("Enter a number :
            \t"); String no = br.readLine();

            String ans = h1.convertDigit(no);
            System.out.println("The word representation of the entered digit is : " +ans);
        }
}
```


C:\WINDOWS\system32\cmd.exe - rmiregistry
E:\Ds_Yugi>javac ServerConvert.java
E:\Ds_Yugi>javac ClientConvert.java
E:\Ds_Yugi>rmic ServerConvert
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
E:\Ds_Yugi>rmiregistry

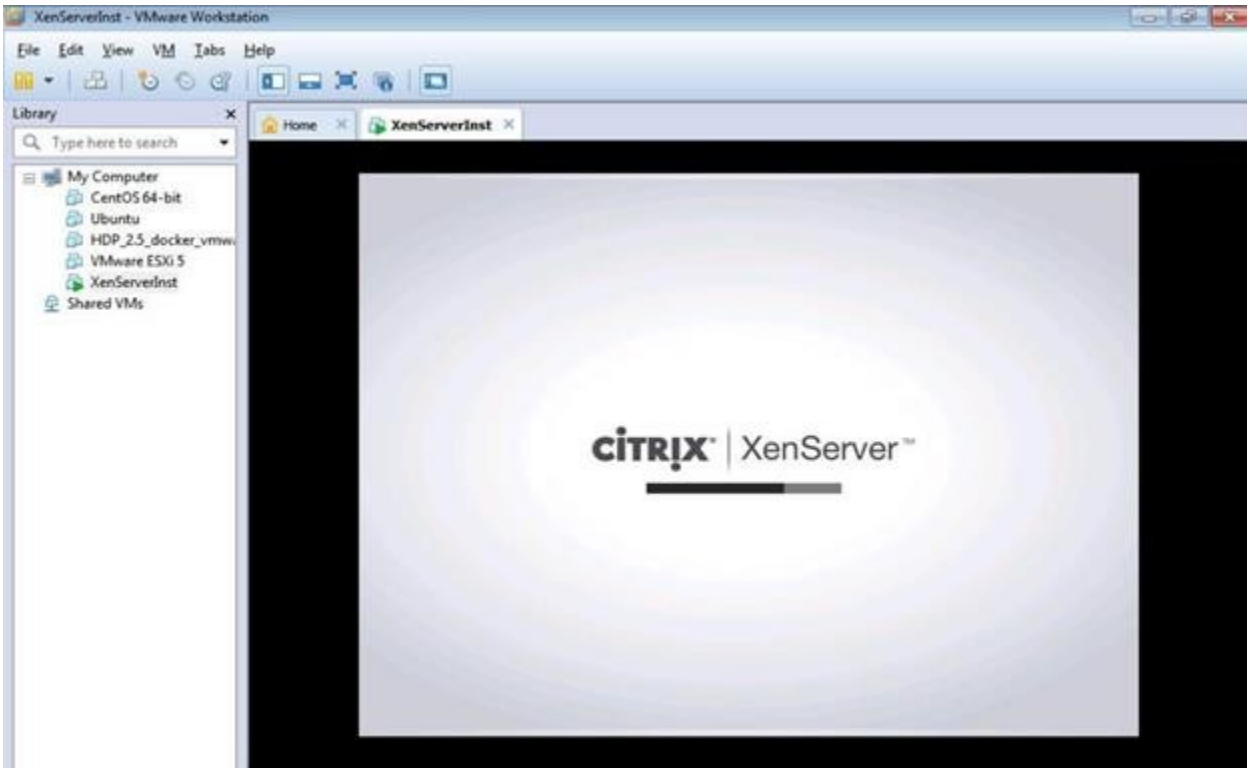

C:\WINDOWS\system32\cmd.exe - java ServerConvert
E:\Ds_Yugi>java ServerConvert
Object registered....


C:\WINDOWS\system32\cmd.exe
E:\Ds_Yugi>java ClientConvert
Enter a number :
123
The word representation of the entered digit is : one two three
E:\Ds_Yugi>java ClientConvert
Enter a number :
456
The word representation of the entered digit is : four five six

Practical: 05

Aim: Implement Xen virtualization and manage with Xen Center

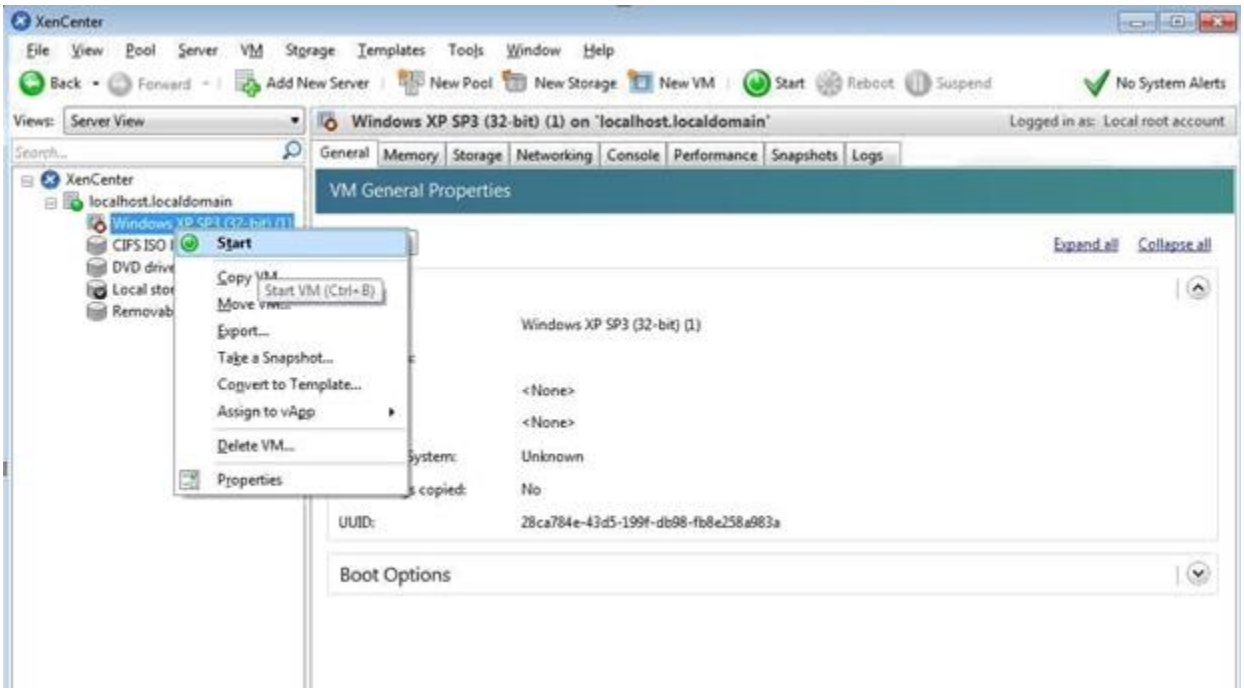
- Install **XenServer** in VMware Workstation and select Guest operating system as Linux.



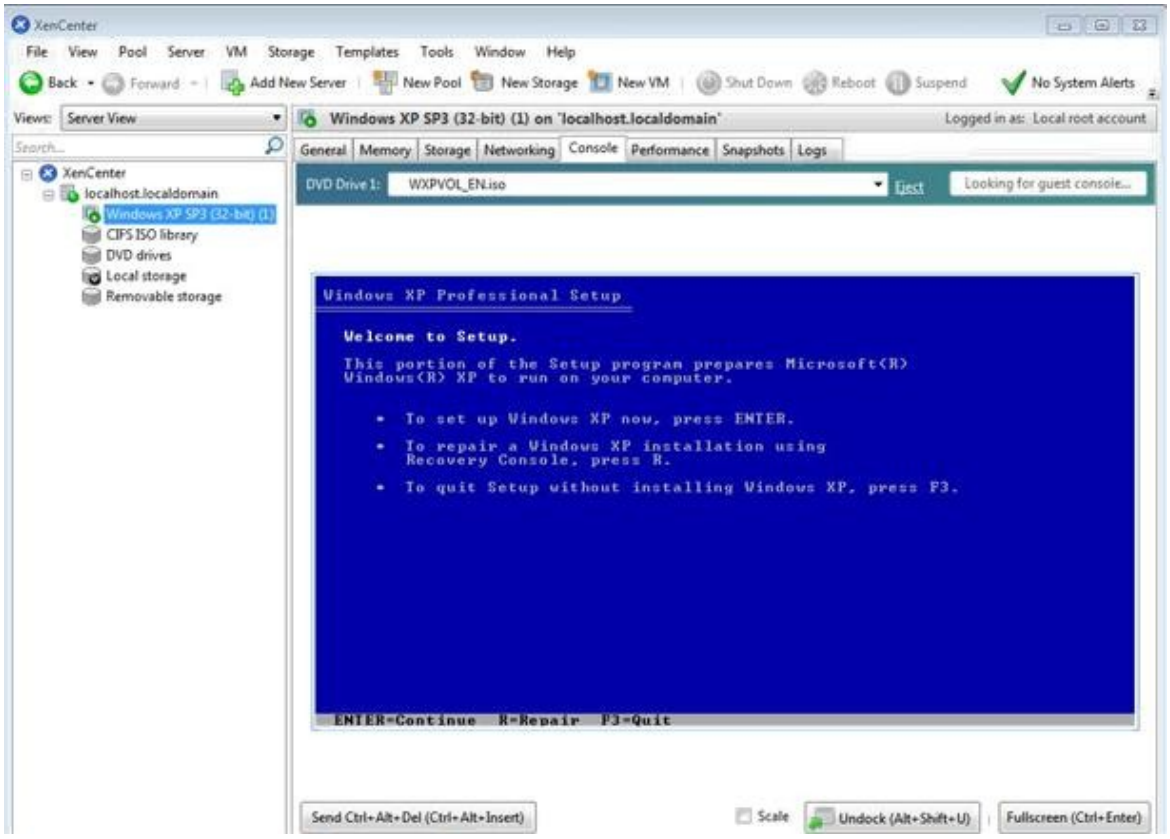
Note IP Address – " 192.168.124.137" ping it from command prompt.

- Now Install Citrix App if not installed
- Now Open Citrix XenCenter – and Click and **Add Server.**

- Fill IP address copied from Installation and User name as “root” and Password as “root123” which we had given during installation and Click on Add.
- Then click on Ok
- Now Click on New Storage
- Select Window File Sharing (CIFS) and click on next
- Uncheck Auto generate option Click on Next.
- Provide the path of shared windows XP image and enter local pc credential , click on Finish
- Click on New VM – and Windows XP SP3
- Select ISO file and click on next –
- Click on Next –
- Uncheck – Start the new VM and click on create now
- Now Right click on Windows XP and Start -



Installation is successful and virtual node has been created if we get below Welcome screen of Windows XP machine.

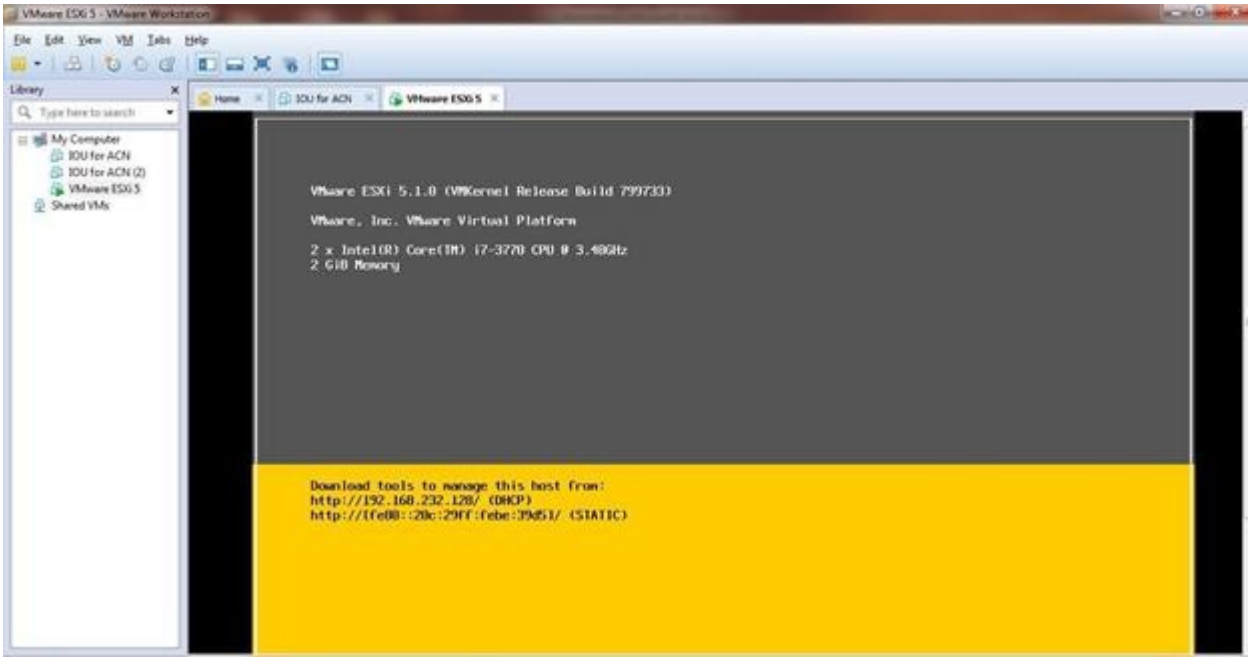

XenCenter
File View Pool Server VM Storage Templates Tools Window Help
Back Forward Add New Server New Pool New Storage New VM Shut Down Reboot Suspend No System Alerts
Views: Server View
Search...
XenCenter
localhost.localdomain
Windows XP SP3 (32-bit) (1)
CIFS ISO library
DVD drives
Local storage
Removable storage
Windows XP SP3 (32-bit) (1) on 'localhost.localdomain'
Logged in as: Local root account
General Memory Storage Networking Console Performance Snapshots Logs
DVD Drive 1: WXPVOL_EN.iso
Eject
Looking for guest console...
Windows XP Professional Setup
Welcome to Setup.
This portion of the Setup program prepares Microsoft(R) Windows(R) XP to run on your computer.
• To set up Windows XP now, press ENTER.
• To repair a Windows XP installation using Recovery Console, press R.
• To quit Setup without installing Windows XP, press F3.
ENTER=Continue R=Repair F3=Quit
Send Ctrl+Alt+Del (Ctrl+Alt+Insert)
Scale
Undock (Alt+Shift+U)
Fullscreen (Ctrl+Enter)

Practical: 06

1. **Aim:** Implement virtualization using VMWare ESXi Server and managing with vCenter

Steps:

Install **ESXi iso** in VMWare workstation.



Install VMware vSphere Client

In vSphere create new **Virtual Machine.** Install Windows XP iso file and open it.

Practical: 07

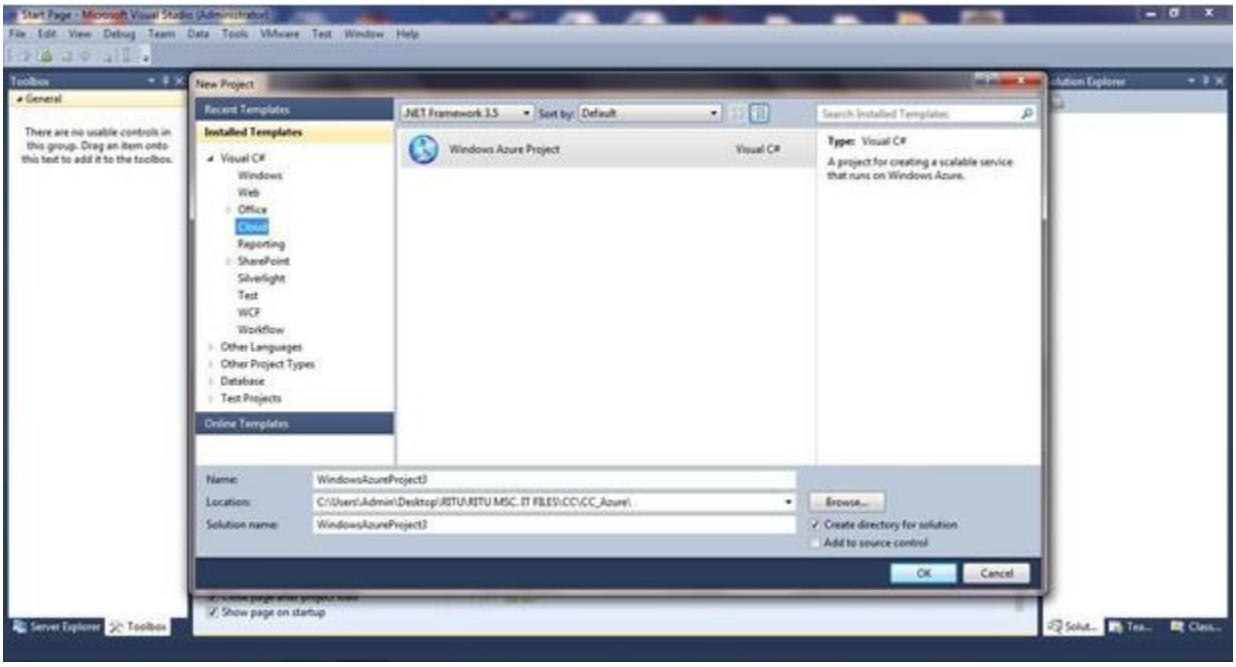
Aim: Develop application for Microsoft Azure.

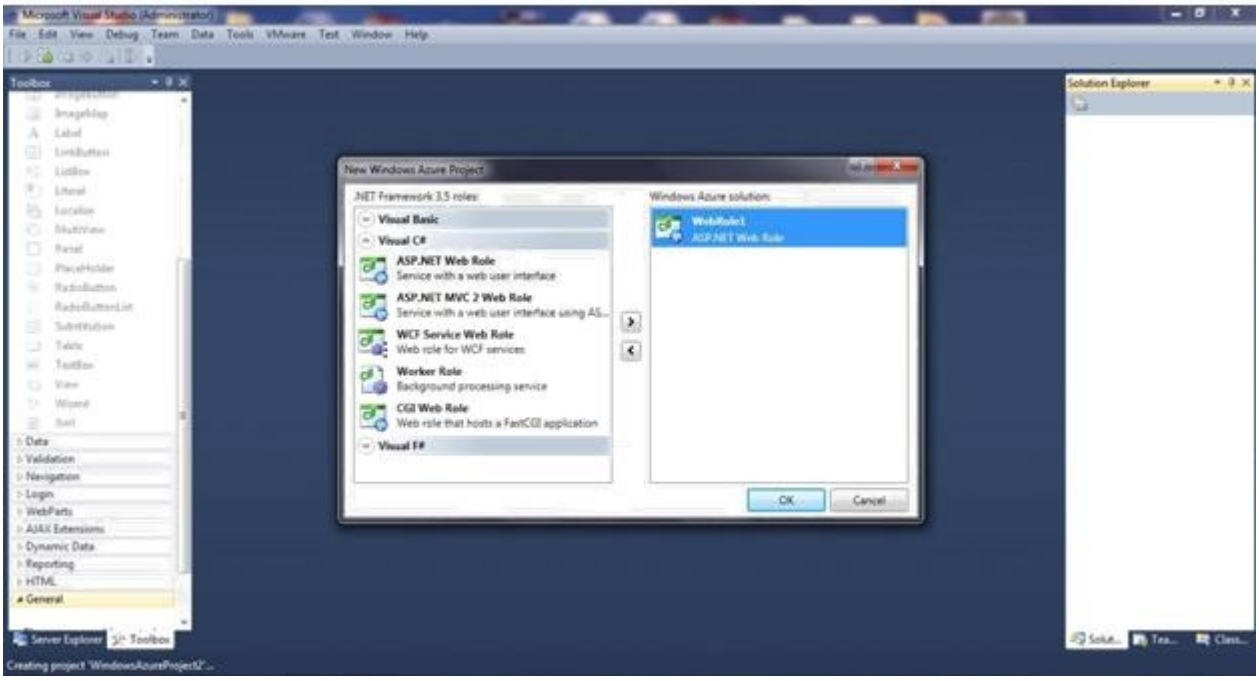
Step 1:
To develop an application for Windows Azure on Visual Studio install the
"Microsoft Azure SDK for .NET (VS 2010) – 2.8.2.1"

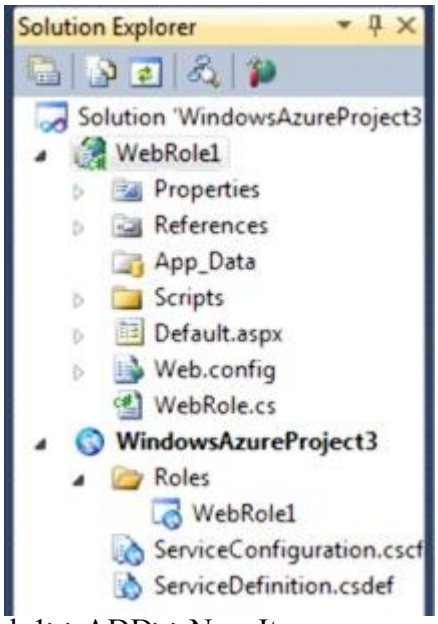
Step2:
Turn windows Features ON or OFF:
Go to Control panel and click on programs.
Turn Windows features on or off.

Step3:
Now, Start the visual studio 2010 and Go To File->New->Project

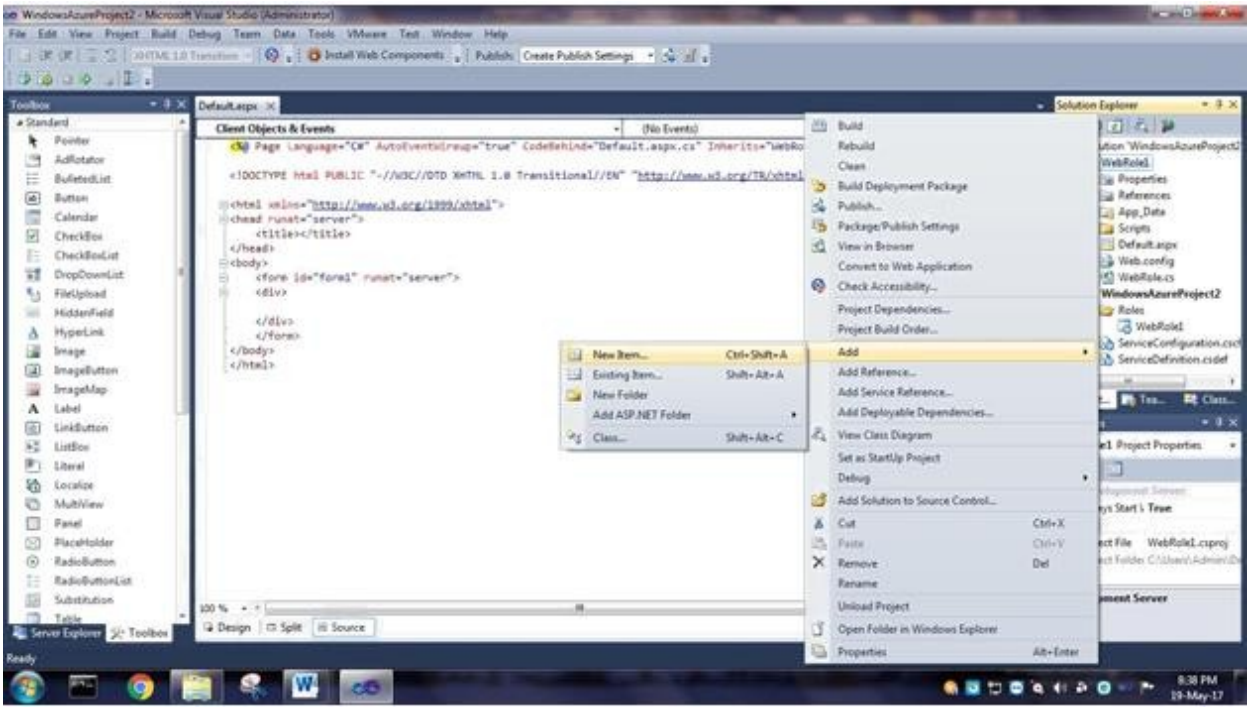
Expand Visual C#-> Select Cloud



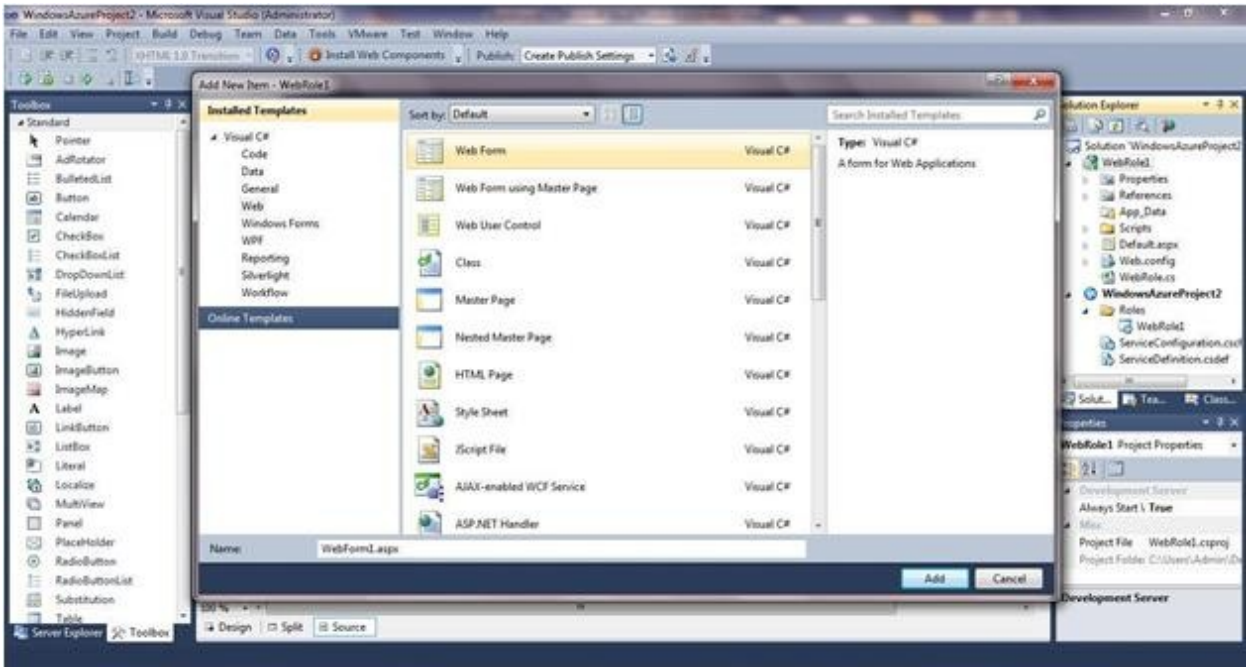




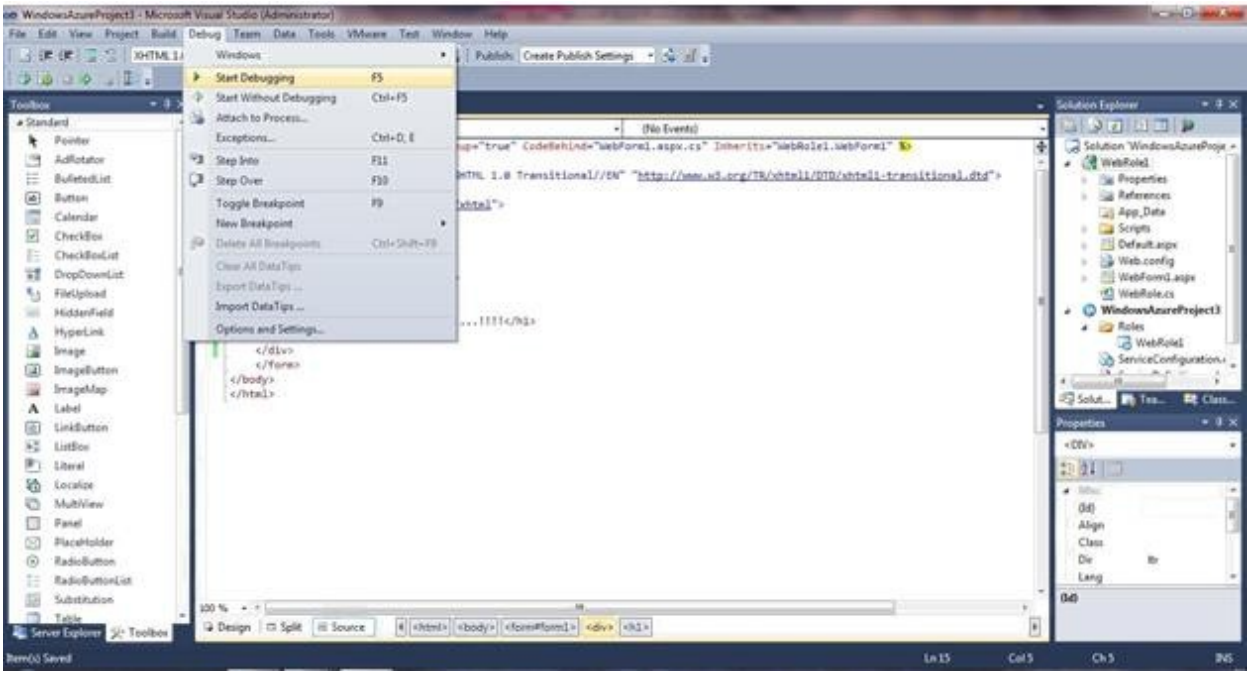
Right Click on WebRole1>>ADD>>New Item



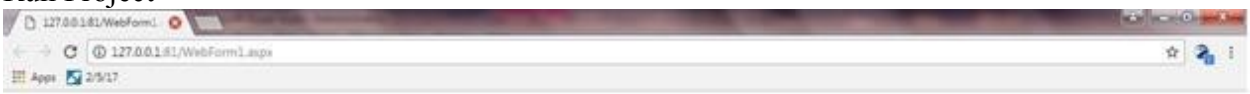
Add a New web Form. Give it a name. Click Add



Deploy the project:



Run Project

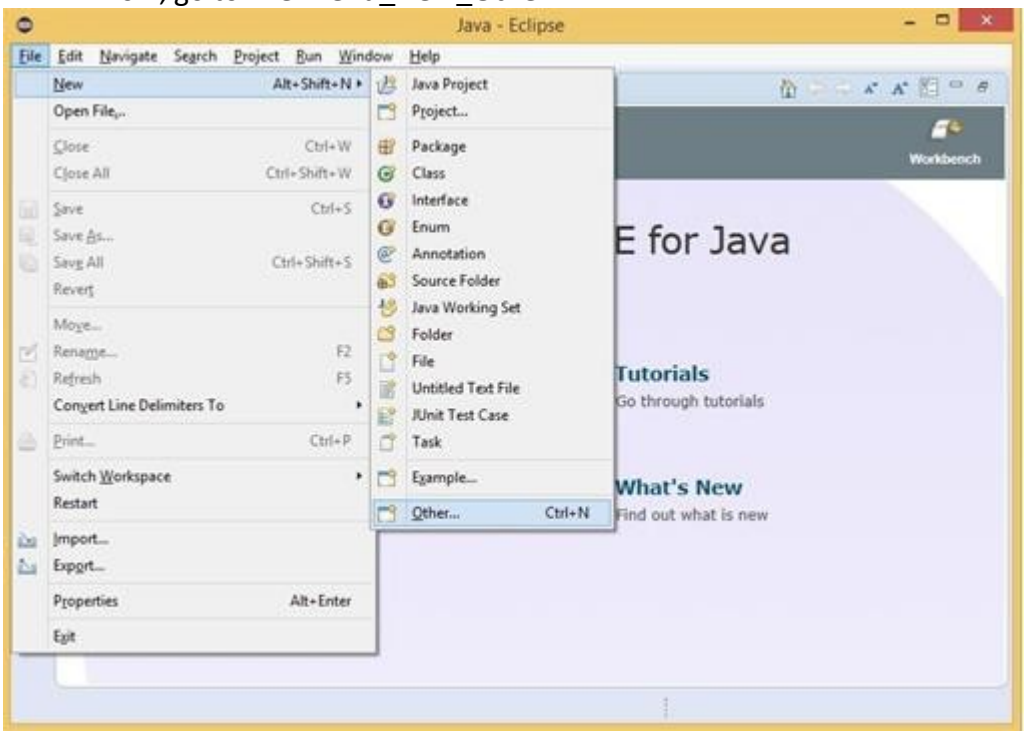


Welcome TO Windows Azure.....!!!!

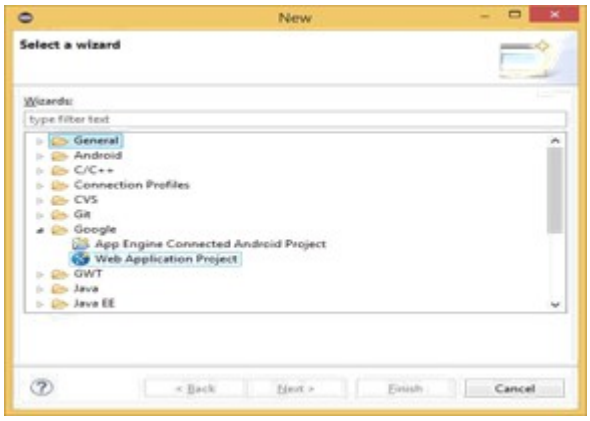
Practical: 8

Aim: Develop application for Google App Engine

- Open Eclipse Luna. Go to **Help Menu Install New Software…**
- In **Install** window Click on the **"Add"** button besides the **Work with** textbox. **Add Repository** window appears. Enter the **Location** as **"https://dl.google.com/eclipse/plugin/4.4"** and click on "**OK**" button.
- From the available softwares select the required softwares and tools as shown in the below image for the **GAE**. Then click on the **"Next"** button.
- In the **Install Details** window click on **"Next"** button.
- In the Next Window "Review the Items to be Installed" then click on **"Next"**
- In the next window for Review Licenses select the option **"I accept……"** and click on **"Finish"**
- button.
- After Installation you will get option to **"Restart Eclipse",** click on **Yes.** So that the software you selected gets updated...
- Now, go to **File Menu_New_Other.**

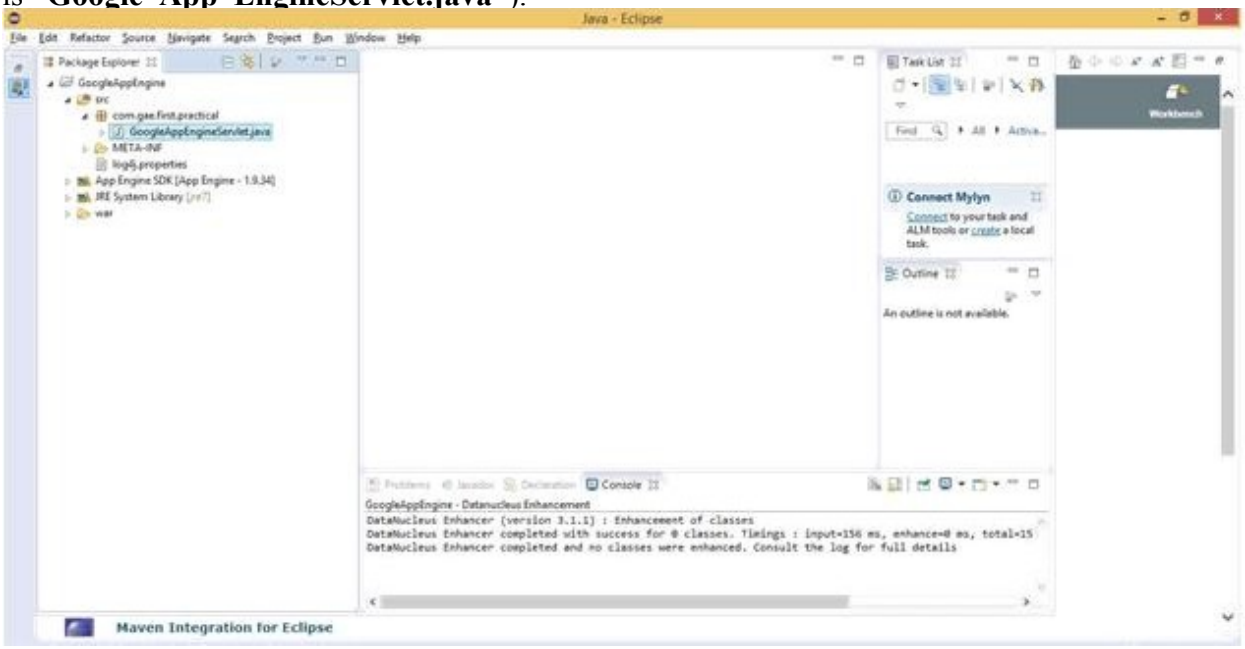


In the New window select **Google_Web Application Project** and click on **"Next"** button.

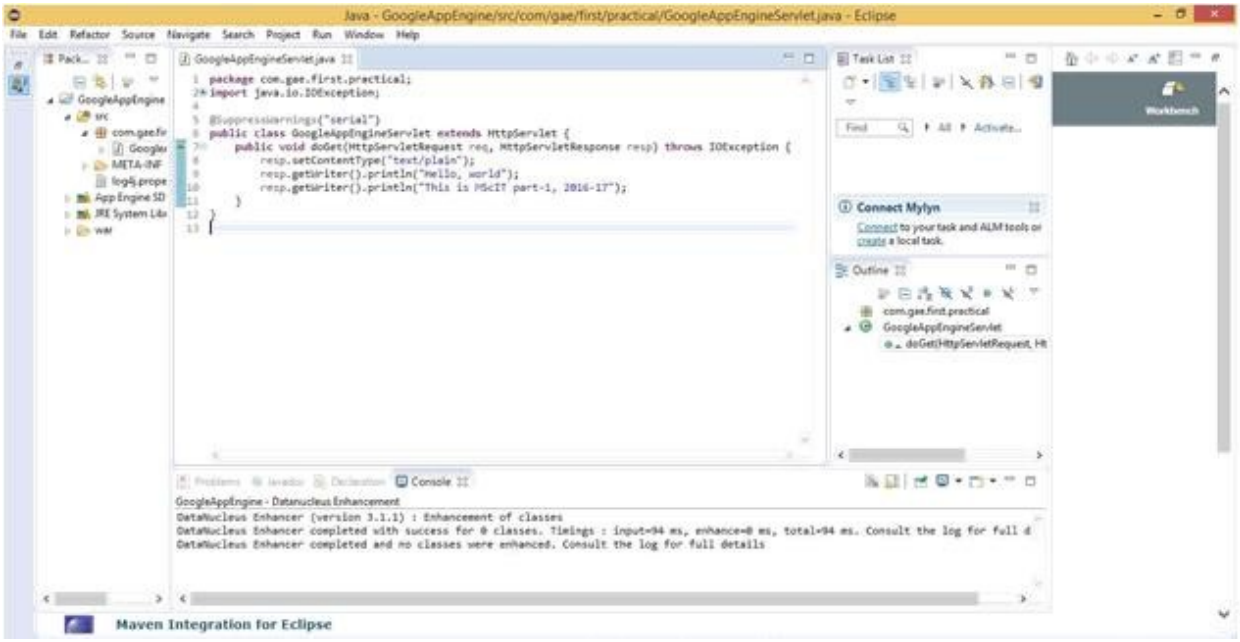


Enter the details for the new Web application project. Deselect the **Use Google Web Toolkit** option under the section **Google SDKs**. Click on the **"Finish"** button.

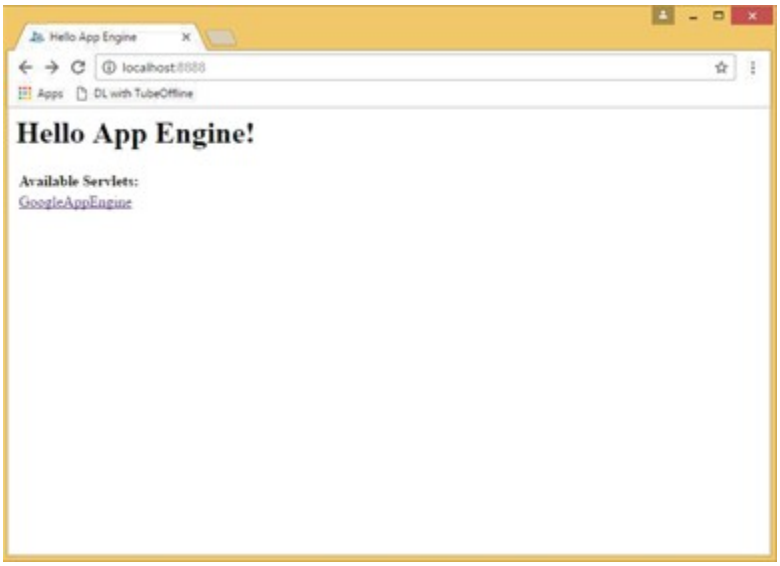
From the **Package Explorer** open the **.java** file (Here it is **"Google_App_EngineServlet.java"**).



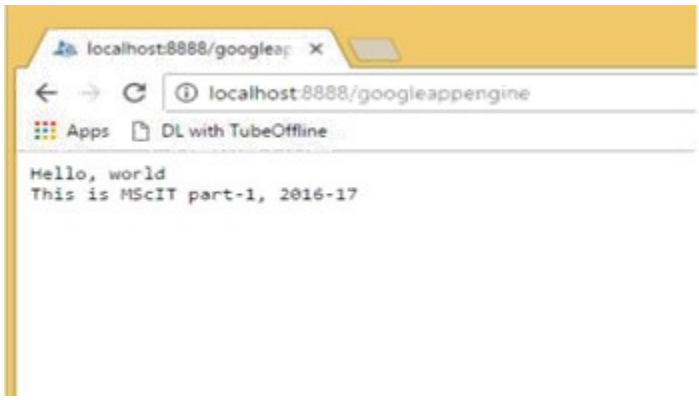
Edit the file as required (Unedited file too can be used. Here the editing is done to "what should be displayed" on the browser). **Save** the file. Click on the **Run** option available on the Tools bar.



In the browser (Here, Google Chrome) type the address as **"localhost:8888" which is "Default"**.



In **localhost:8888** the link to the **Google_App_EngineServlet.java** file as **Google_App_Engine** is displayed. Click on this link. It will direct you to **"localhost:8888/Google_App_Engine"**.



The **output text entered** in the **java** program is **displayed as the output** when clicked the link "Google_App_Engine".