CLOUD COMPUTING

PRACTICAL 1

1. VMware Tutorial: Install ESXi Hypervisor

A hypervisor is software that allows a server to run multiple operating systems. It allows you to create virtual machines (VM), which enables users to customize hardware resources and perform actions such as taking a VM snapshot and migrating virtual machines to other hosts without downtime.

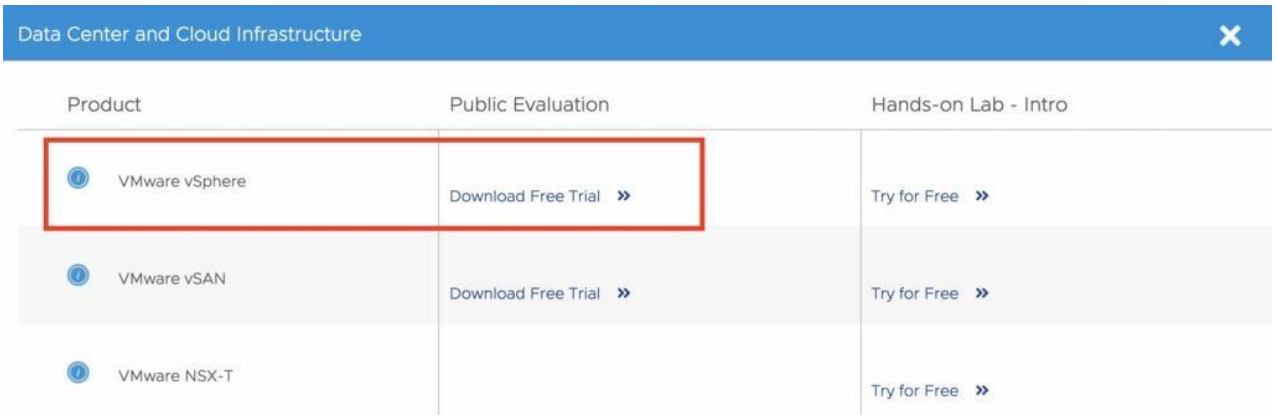
VMware ESXi is one of the most commonly used type-1 hypervisors in a data center environment. It runs directly using the host's hardware, rather than acting as a standard Windows or Mac application.

Do you know that you can install ESXi hypervisors at home for experimental purposes? This article introduces the steps to install VMware ESXi hypervisor on a virtual machine created in VMware Workstation.

Step 1: Download VMware ESXi Hypervisor and VMware Workstation

Head over to VMware's official website to download an evaluation version (ISO version) of VMware vSphere. Note that you will need to register a free account to download the ISO.

https://www.vmware.com/try-vmware.html



Download VMware vSphere ESXi

For ease of installation, you can install VMware Workstation (a type-2 hypervisor, also known as VMware Fusion on macOS platform) on your operating system to create a virtual machine, and install ESXi hypervisor (a type-1 hypervisor) on the VM.

https://www.vmware.com/products/workstation-pro.html

Tips #1: If your computer does not have enough processing power, or if you prefer to skip the installation step, I would recommend you to try out VMware Hands-on Lab. It provides an interactive interface for you to use ESXi software on a web browser – without the need to install any software. You can register a free account by visiting the below website.

https://hol.vmware.com/

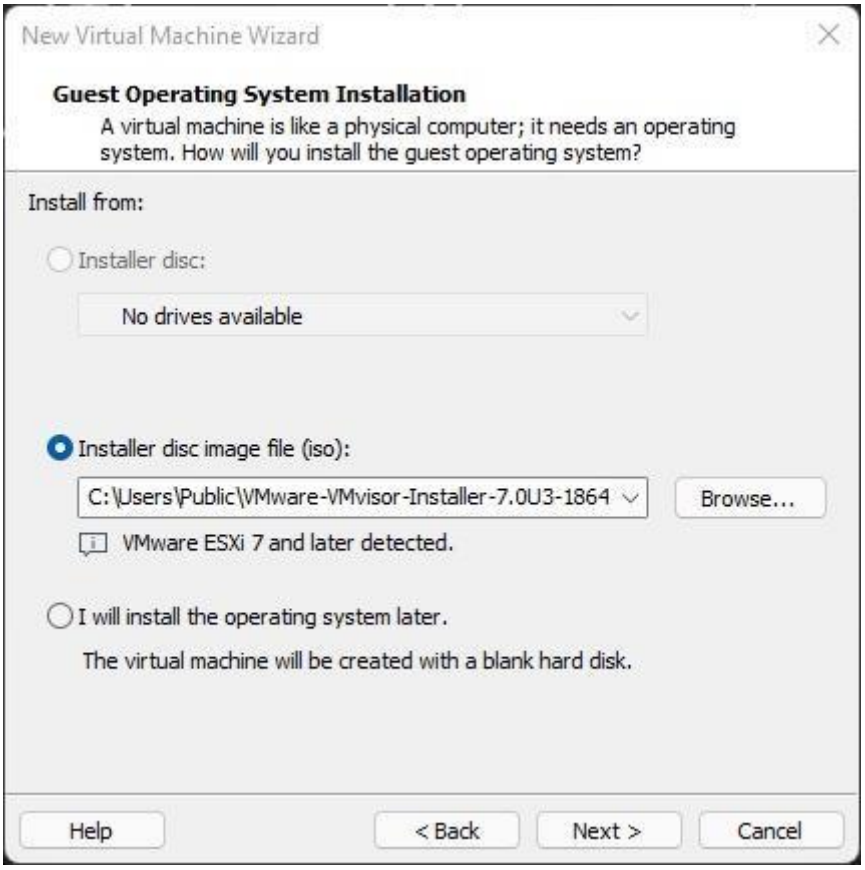
Tips #2: If you would like to use the evaluation software for an extended period, consider joining VMware User Group (VMUG) and upgrading to Advantage membership. You can get 365-day evaluation licenses for most VMware products including ESXi, vCenter, NSX, etc. for an annual membership fee.

https://www.vmug.com/

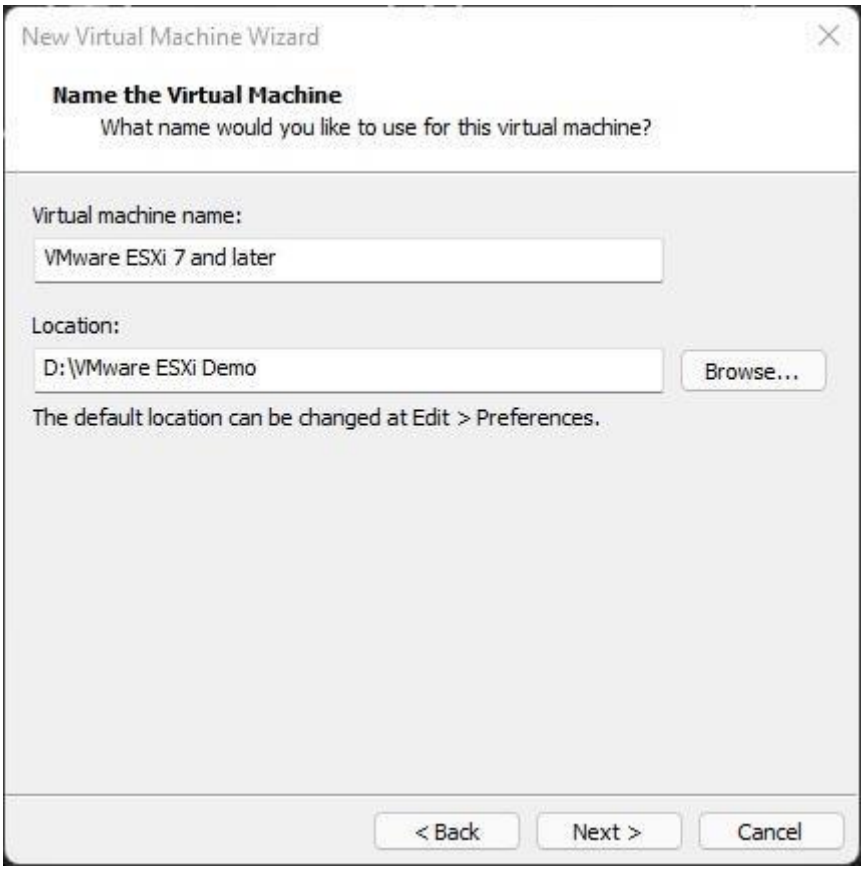
Step 2: Create a virtual machine for ESXi installation on VMware Workstation

Start by creating a VM using the "New Virtual Machine Wizard" on VMware Workstation.

Create a new Virtual Machine

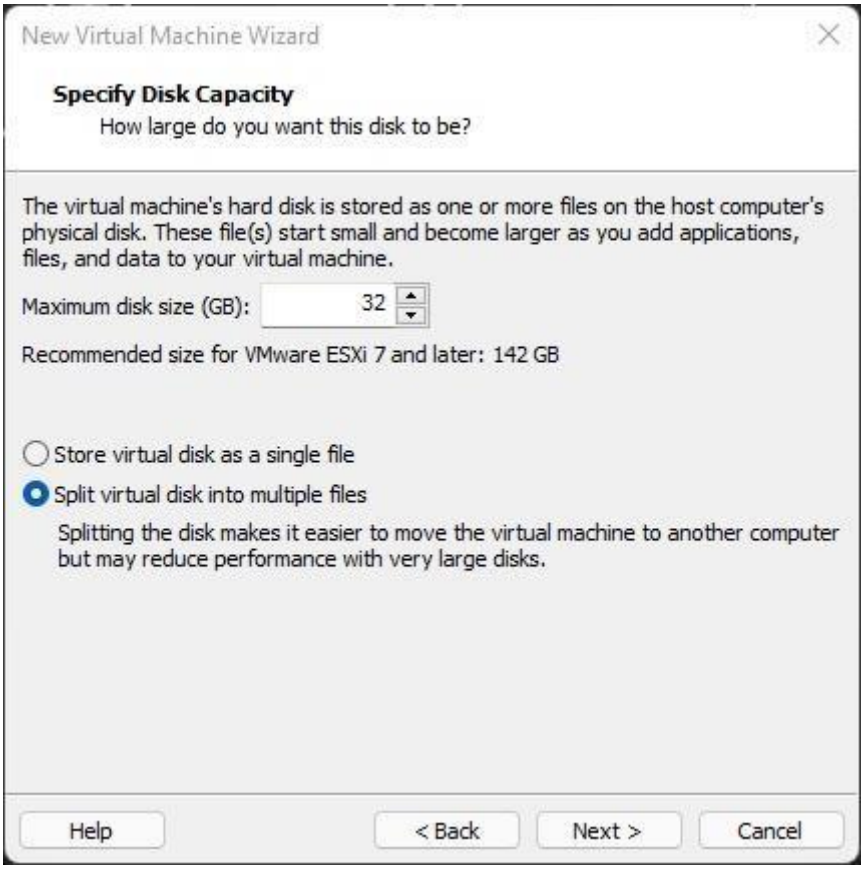


Select the ESXi ISO



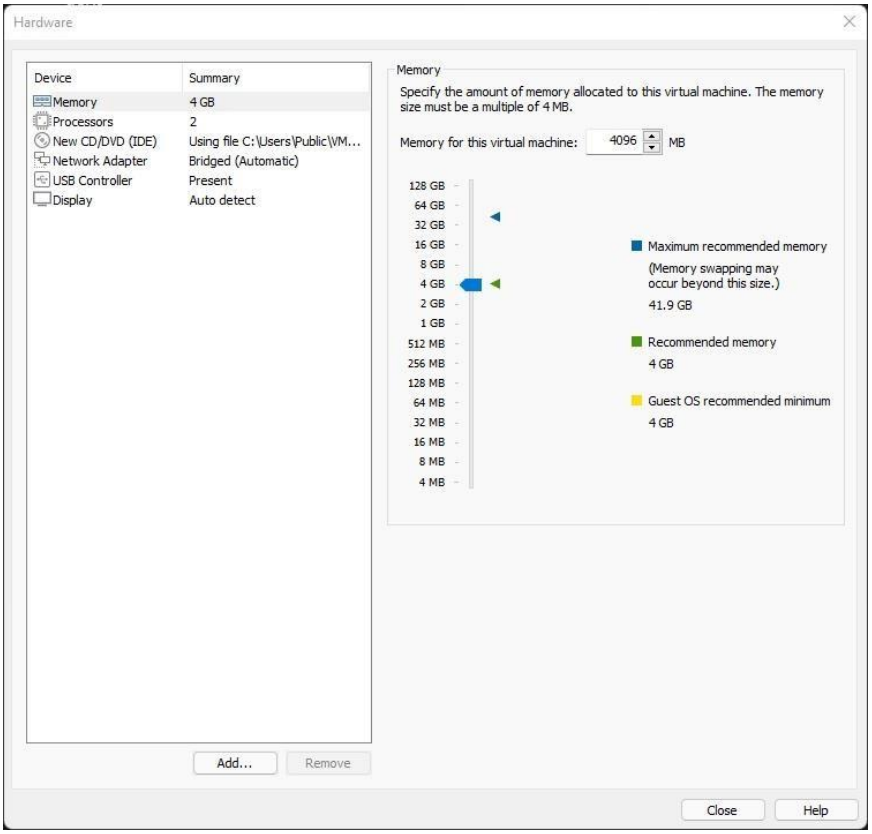
Define VM Name and Location

Refer to the official documentation for the minimum hardware requirements for ESXi 7.0 (eg. 4GB ram, 2 CPU cores, and 32GB disk space)



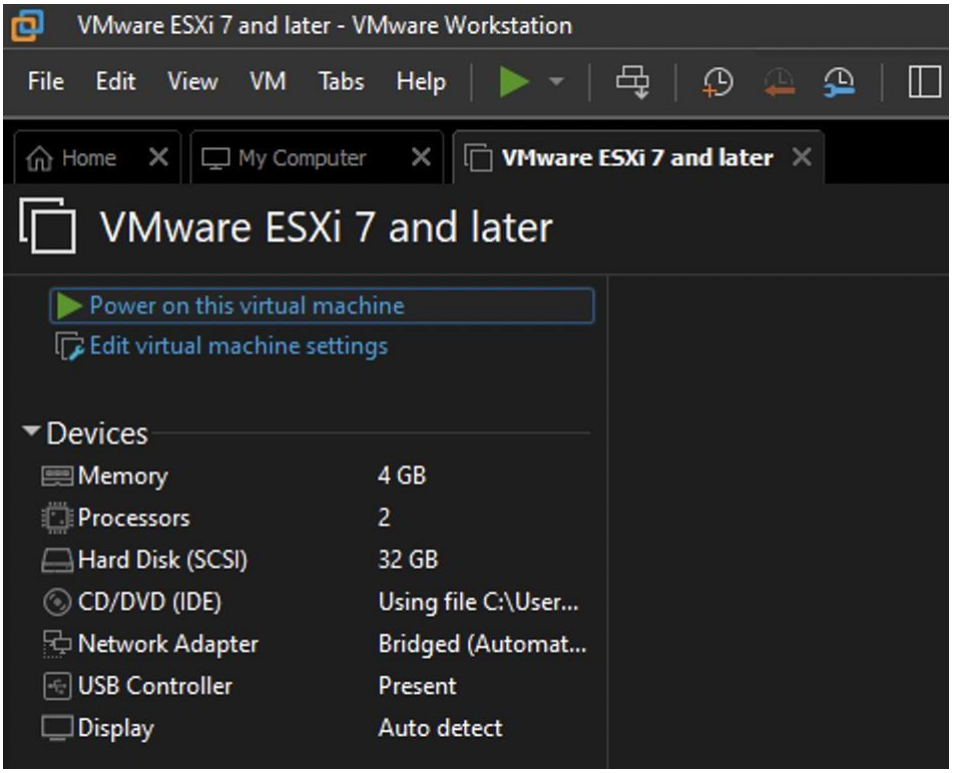
Specify Disk Capacity

Note that you can select “Bridged” mode for the Network Adapter setting if you wish to access the [ESXi host client] from devices other than your host PC after installation.



Customize VM Hardware
Step 3: Power on Virtual Machine and proceed with installation

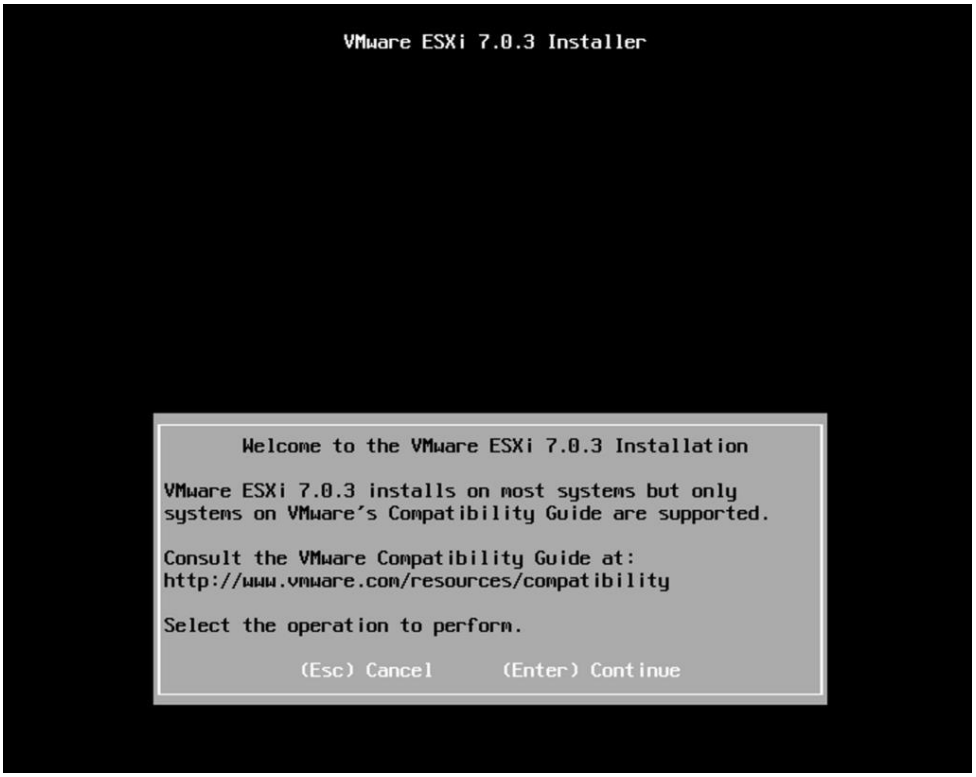
After creating the virtual machine, power on the VM to continue the ESXi installation.



Power on Virtual Machine

ESXi Installer #1

ESXi Installer #2

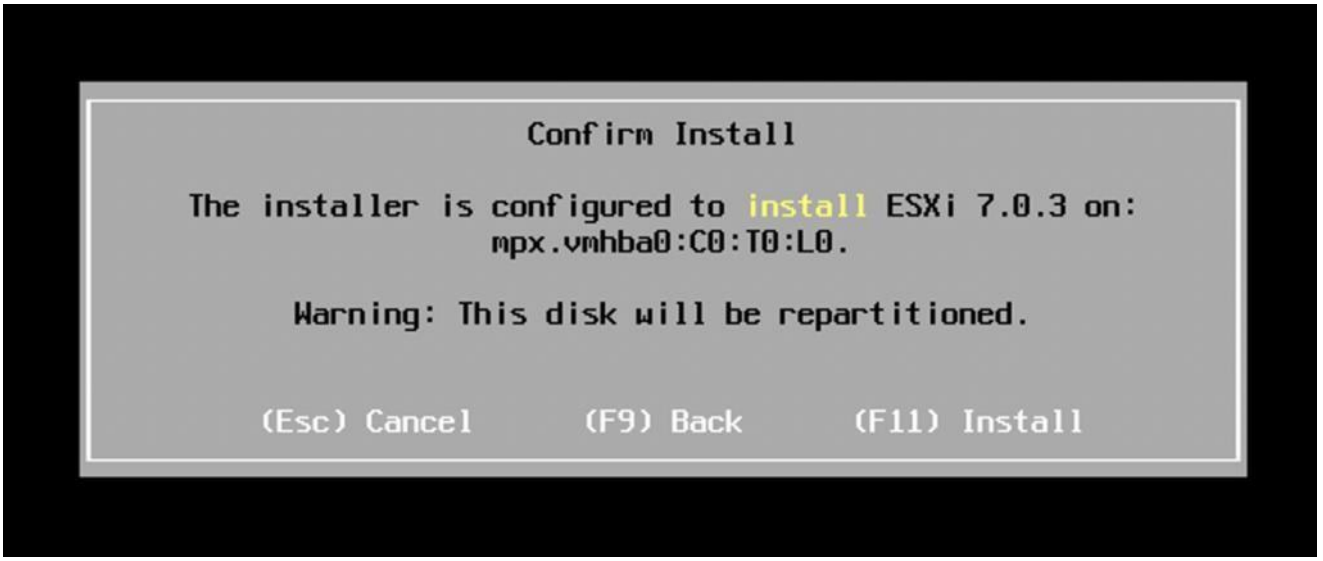


ESXi Installer Start Wizard
After starting the installation, you will be asked to select a disk location to install ESXi. As the VM is configured with a single local 32GB hard disk (which is the minimum amount of disk space for ESXi 7.0 installation), it is the only storage device displayed on the ESXi installation wizard. Continue by selecting the disk.

ESXi Installer Select Disk
Upon selecting the storage device for the installation, you will be asked to enter an ESXi root password, which you can later use to log in to the ESXi host client, as well as the Direct Console User Interface (DCUI).

ESXi Installer Enter Root Password

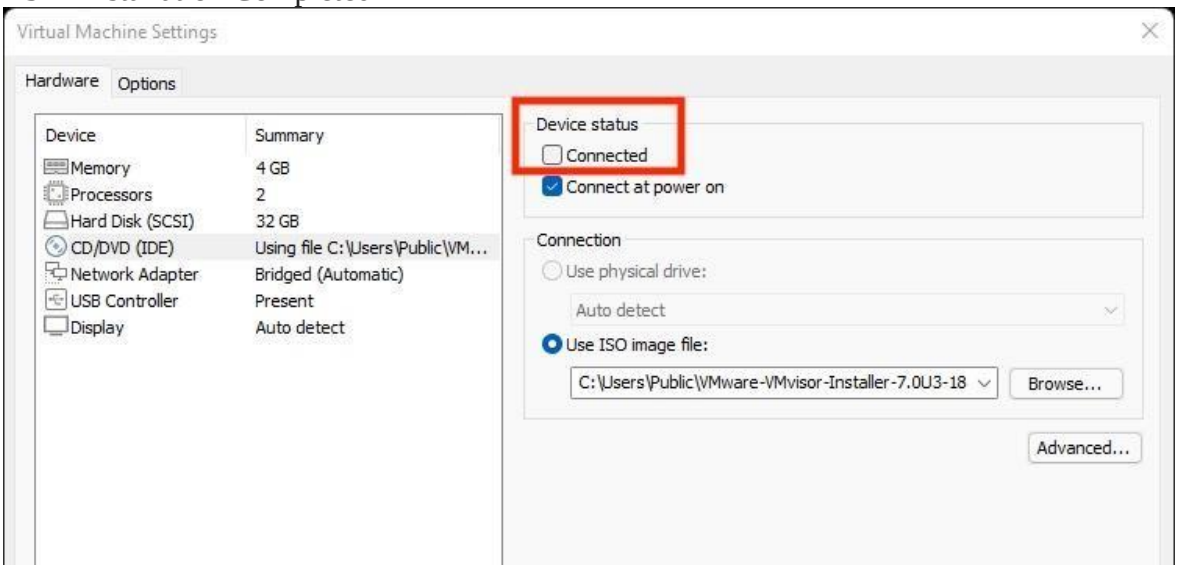
Press F11 to confirm installing ESXi on the selected disk location:



ESXi Installer Confirm Install

After installation has been completed, select “configure hardware” on VMware Workstation, remove the ESXi installation media (e.g. CD Rom, USB drive), and press “Enter” to reboot into .

ESXi Installation Completed

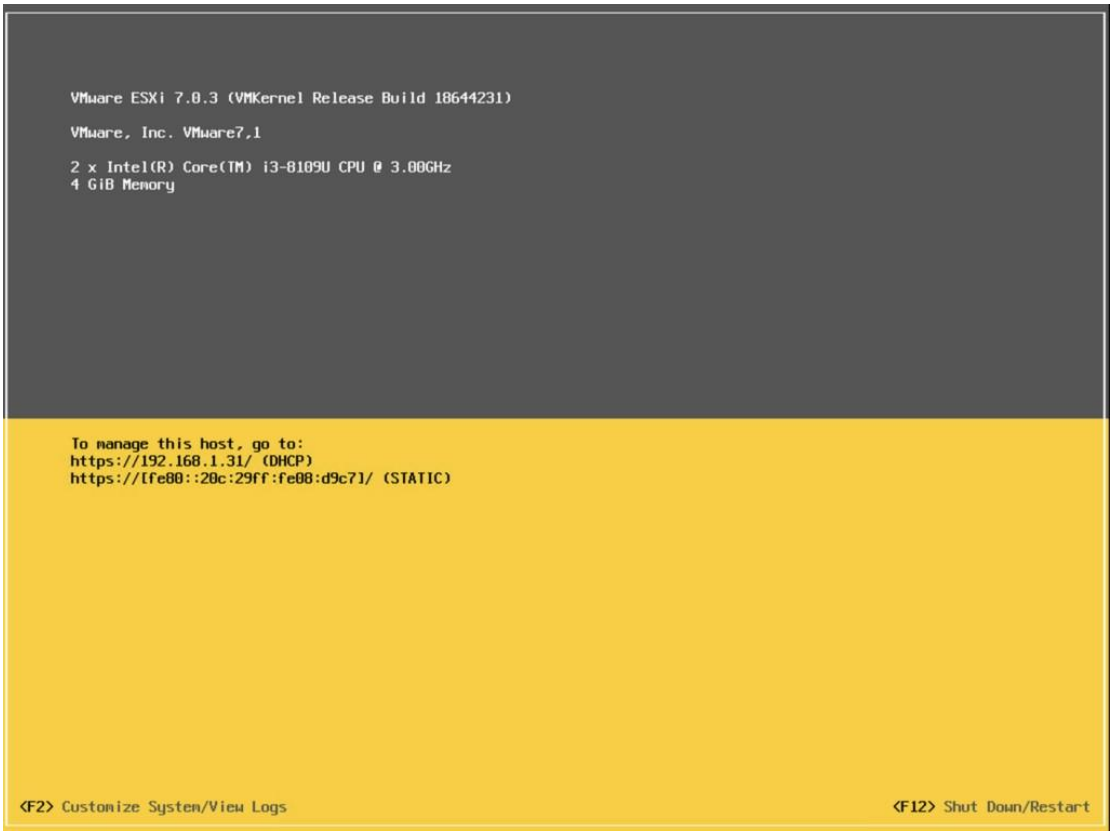


Disconnect CD/DVD from the Virtual Machine

Post-installation: Access Direct Console User Interface (DCUI) and ESXi Host Client

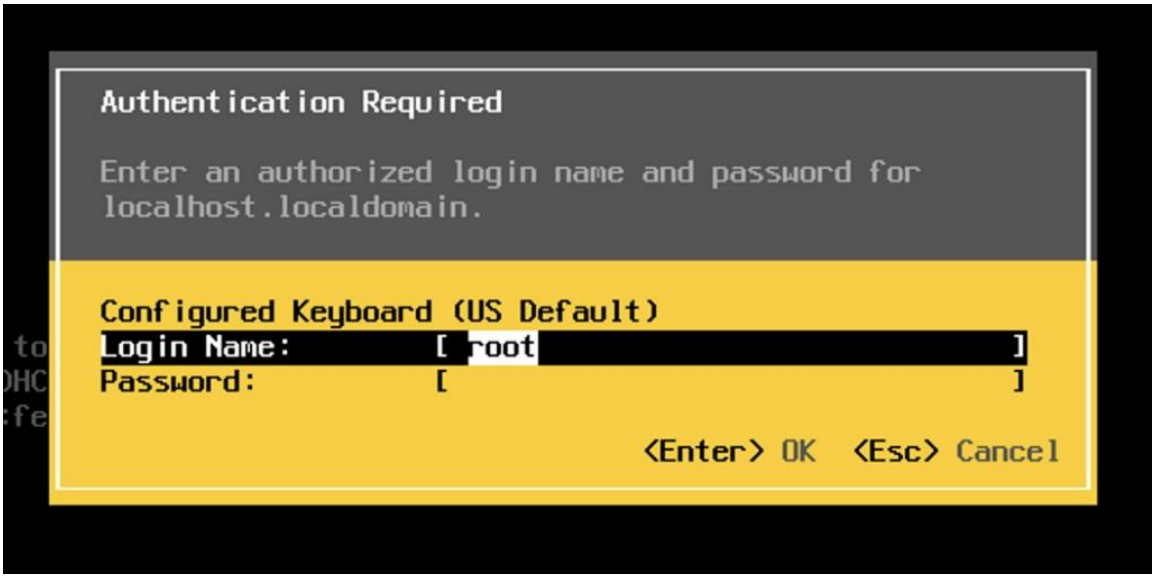
#1) Access Direct Console User Interface (DCUI) on the Connected Display Device

After installing ESXi and rebooting, you can view the Direct Console User Interface (DCUI). Basic information including ESXi version, CPU, memory, and management IP address will be shown. Note that DCUI offers limited configuration as it is not designed for daily operation. For most use cases, system admins only access DCUI via Integrated Management Module (IMM) to perform tasks such as changing ESXi management IP and checking system status when the network connection to the ESXi is lost. For performing most operational and management tasks, you will need to use the ESXi host client on a web browser.



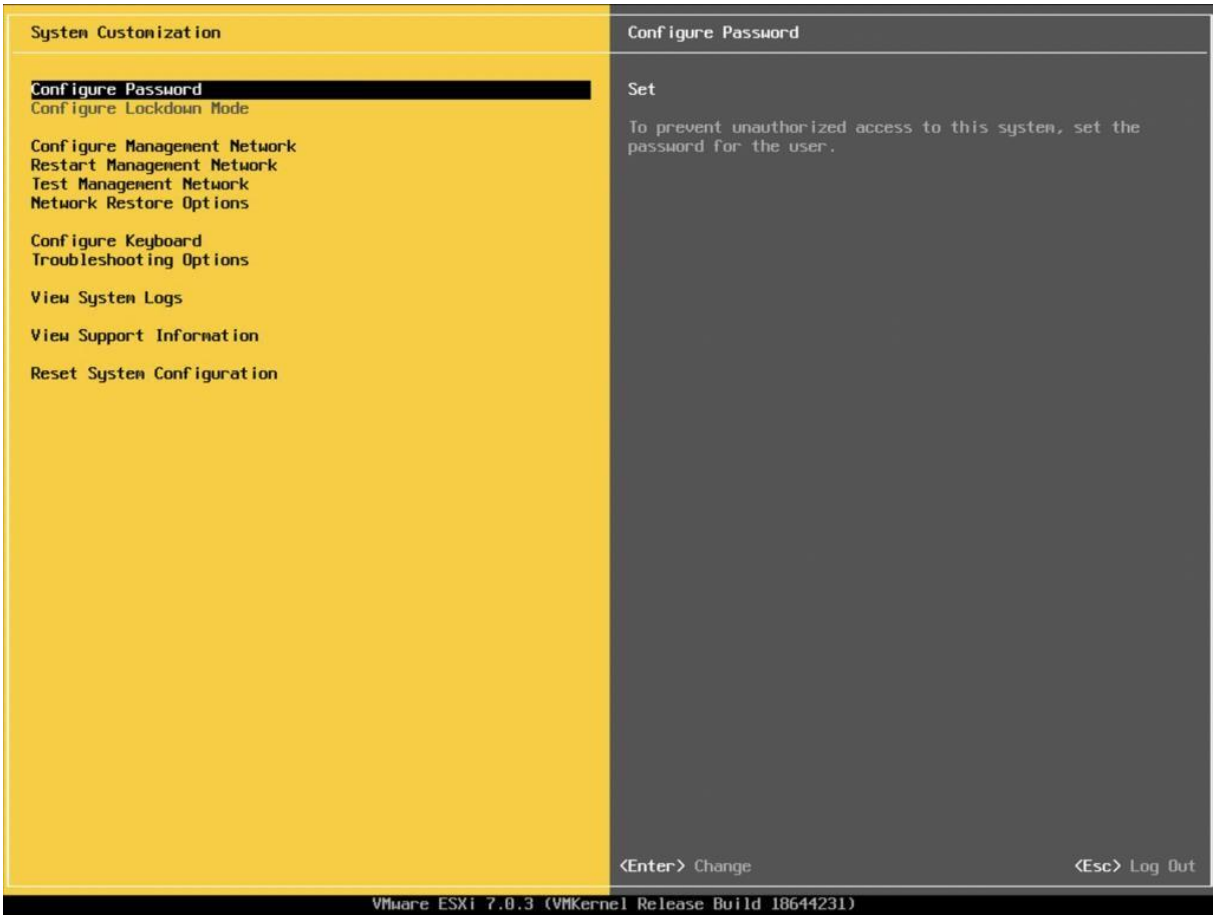
VMware ESXi Direct Console User Interface (DCUI)

For customization such as configuring the management IP address, press F2 and enter the root user credential you entered during the ESXi installation.



VMware ESXi DCUI Login

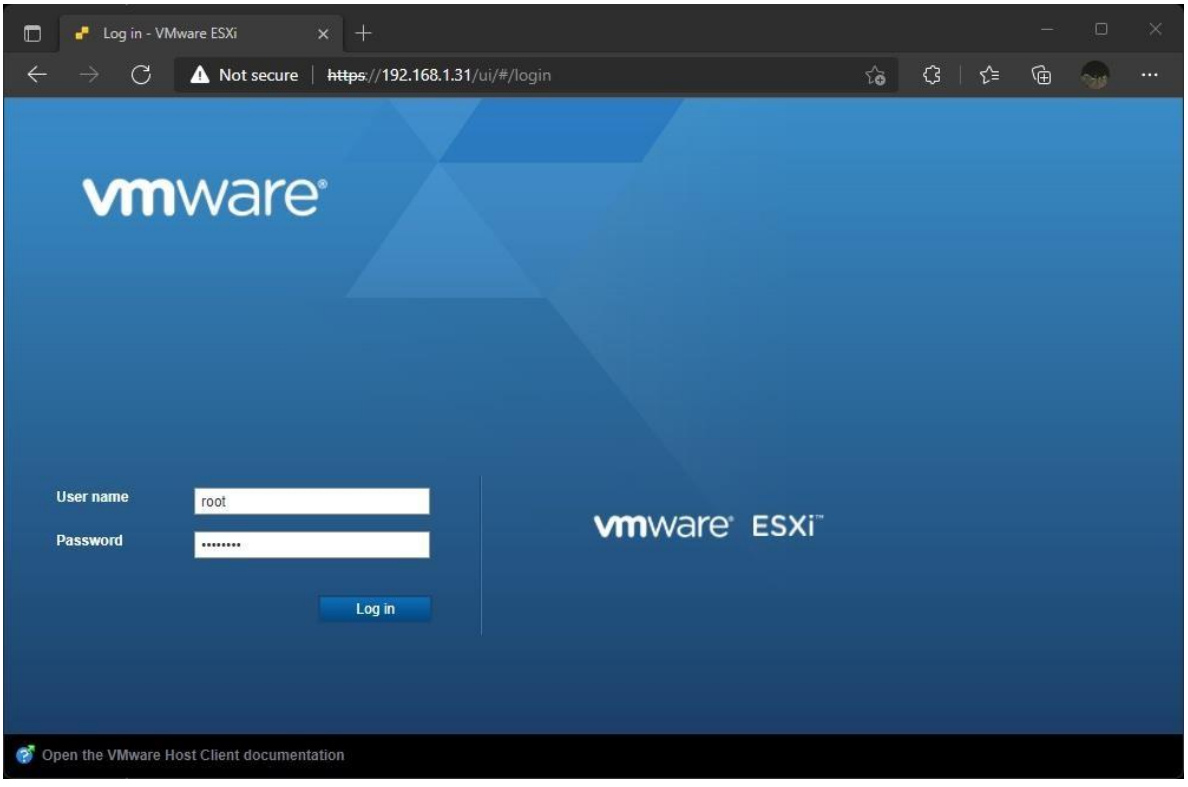
After logging in as a root user, you can perform various configurations on the DCUI including changing the ESXi root password and the management network (e.g. assign a static management IP).



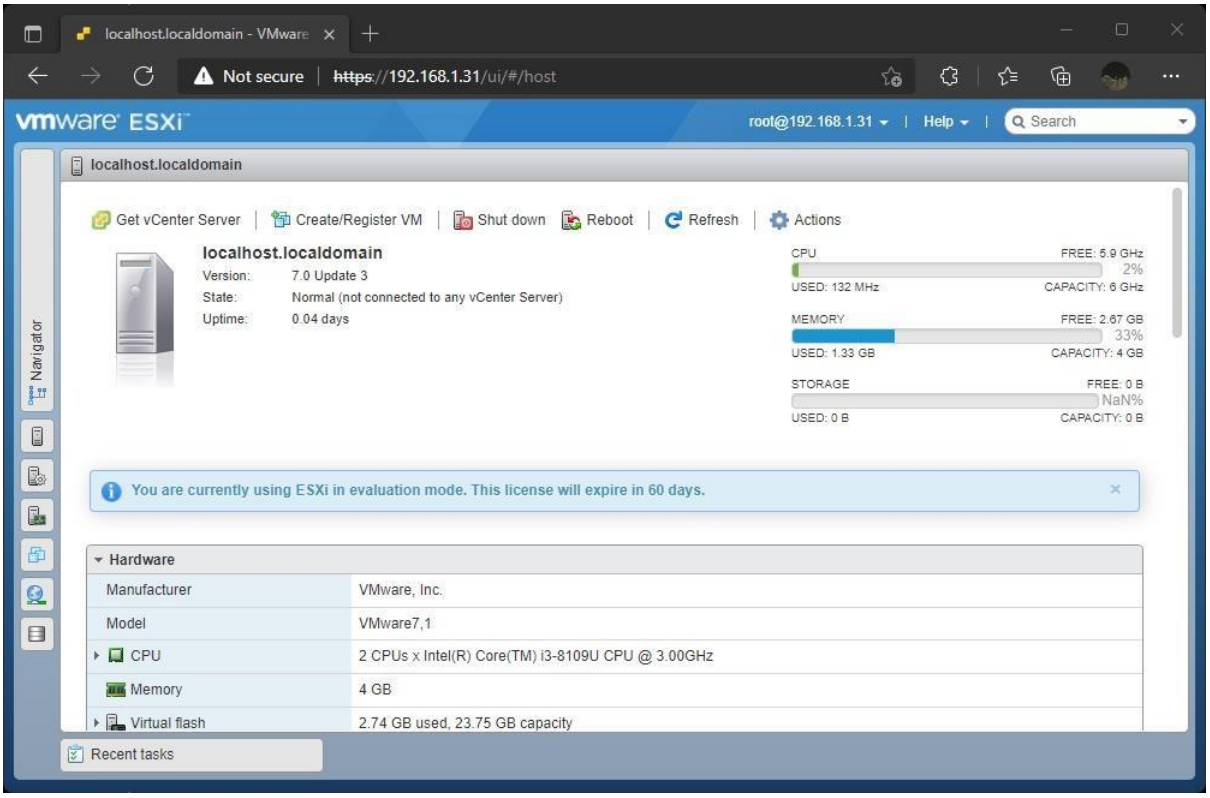
VMware ESXi DCUI Settings

[#2) Access ESXi Host Client via a Remote Web Browser]

You can also access the ESXi management console by entering the management IP of the ESXi on a web browser. This is the preferred way to perform management tasks including datastore and VM configuration etc.



ESXi Host Client Login Page
After login, you can view the system information and perform management tasks.



ESXi Host Client UI

Conclusion

ESXi hypervisor provides a virtualized computing platform where multiple operating systems can be run on a single host. It offers many more capabilities including the ability to provision virtual machines with customized hardware and the option to create a snapshot for the temporary preservation of VM state and data. Installing the ESXi hypervisor is the first step to creating a software-defined data center (SDDC), where computing resources can be pooled to provide a centralized platform for hosting enterprise applications efficiently. Stay tuned for more tutorials on cloud computing and system administration topics.

PRACTICAL 2

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

Aim: Write a program for implementing Client Server communication model using TCP.
Practical 2A: A client server based program using TCP to find if the number entered is prime.
Code:

1. **tcpServerPrime.java**

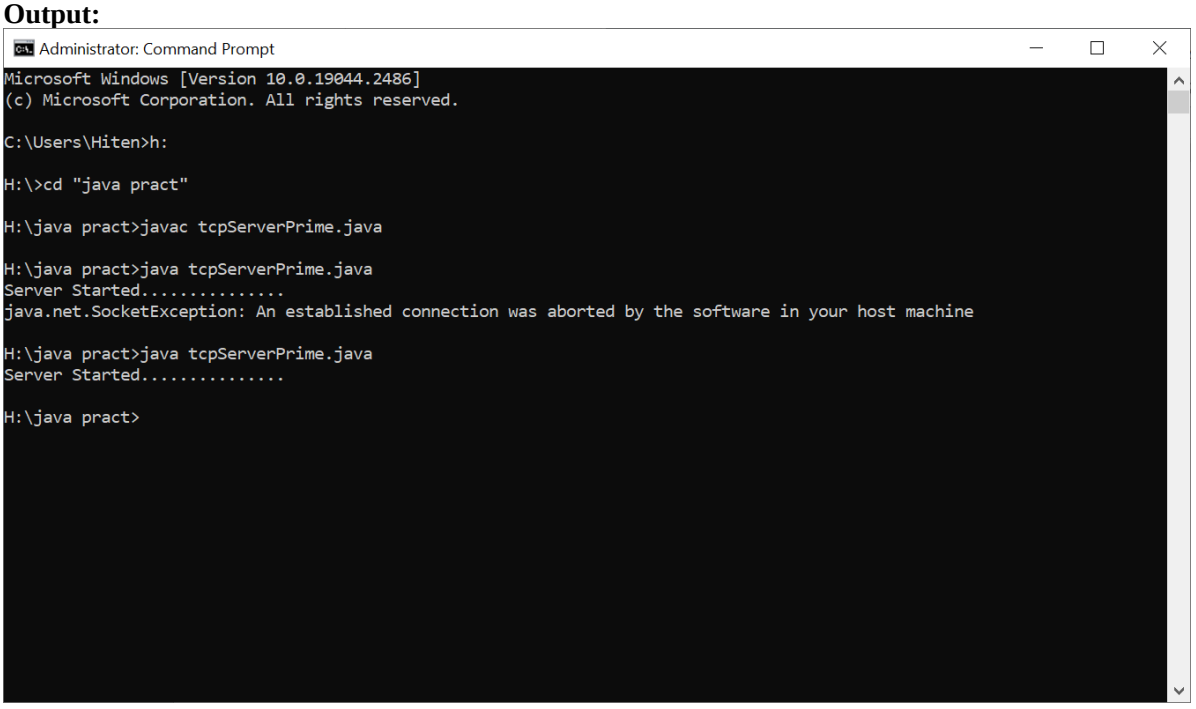
```
import java.net.*;
import java.io.*;
class tcpServerPrime
{
public static void main(String args[])
{
try
{
ServerSocket ss = new ServerSocket(8001);
System.out.println("Server Started. ..... ");
Socket s = ss.accept();
DataInputStream in = new
DataInputStream(s.getInputStream()); int x= in.readInt();
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());
}
}
}
```

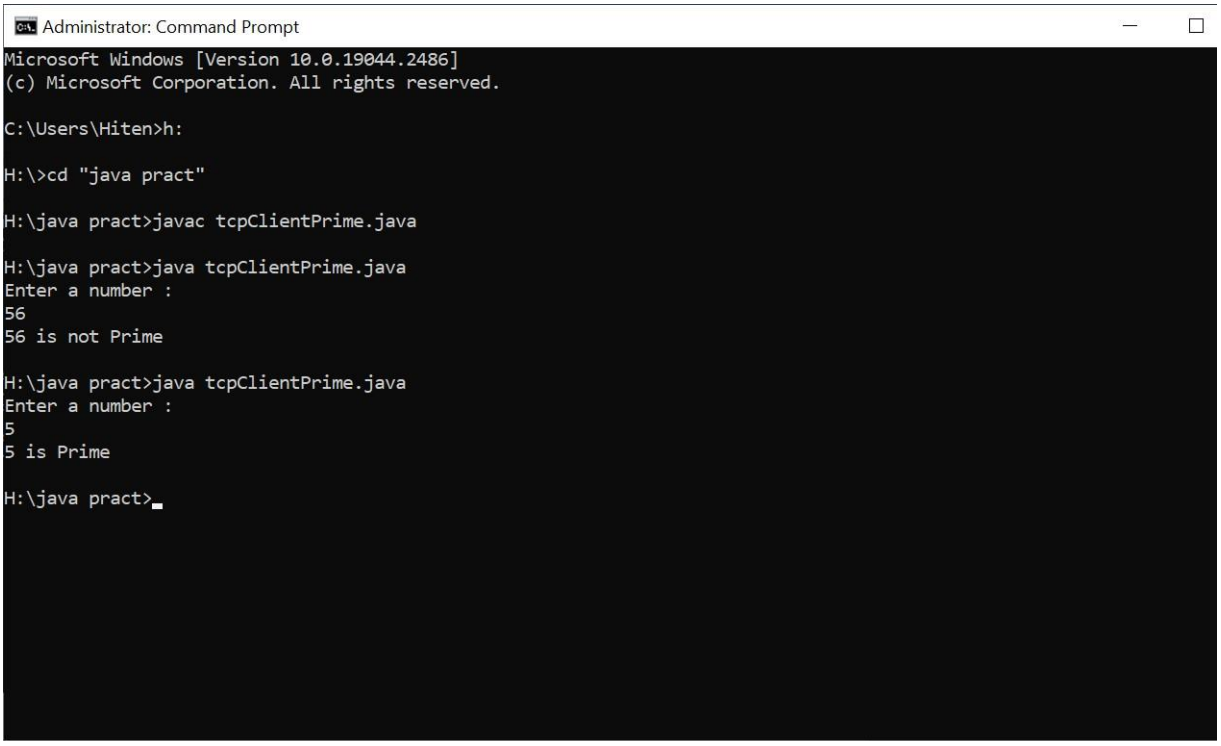
2. **tcpClientPrime.java**

```
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 2B: 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();
```

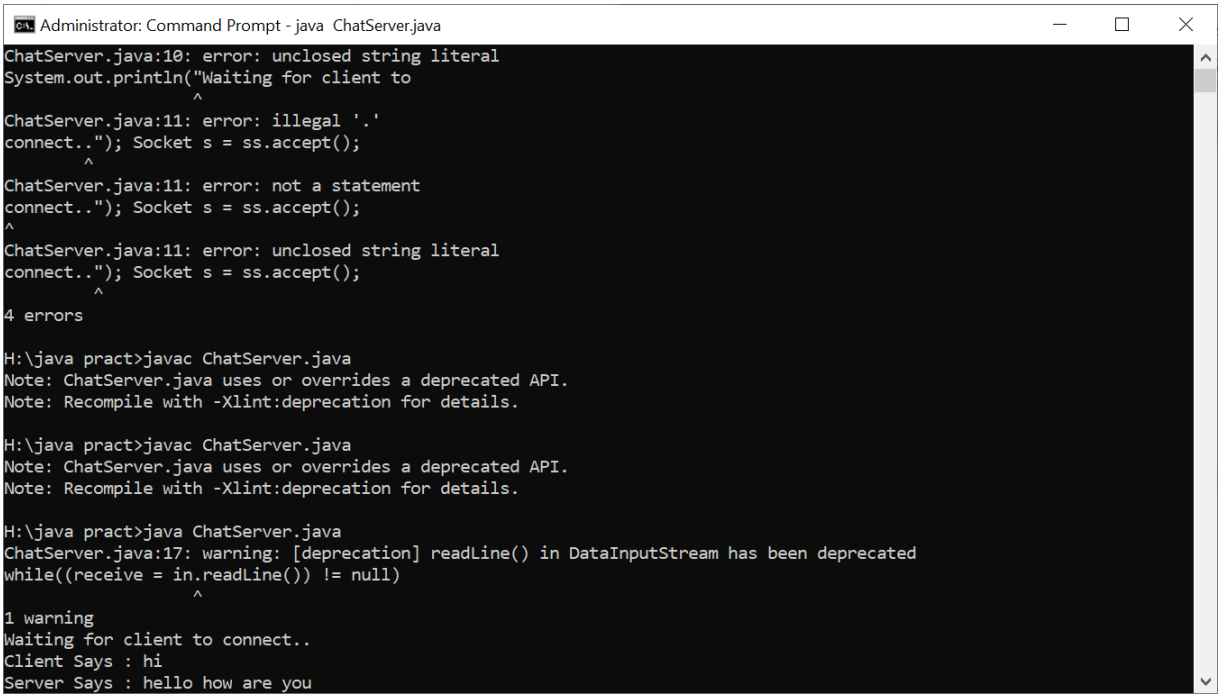
```
in.close();
out.close();
s.close();
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
```

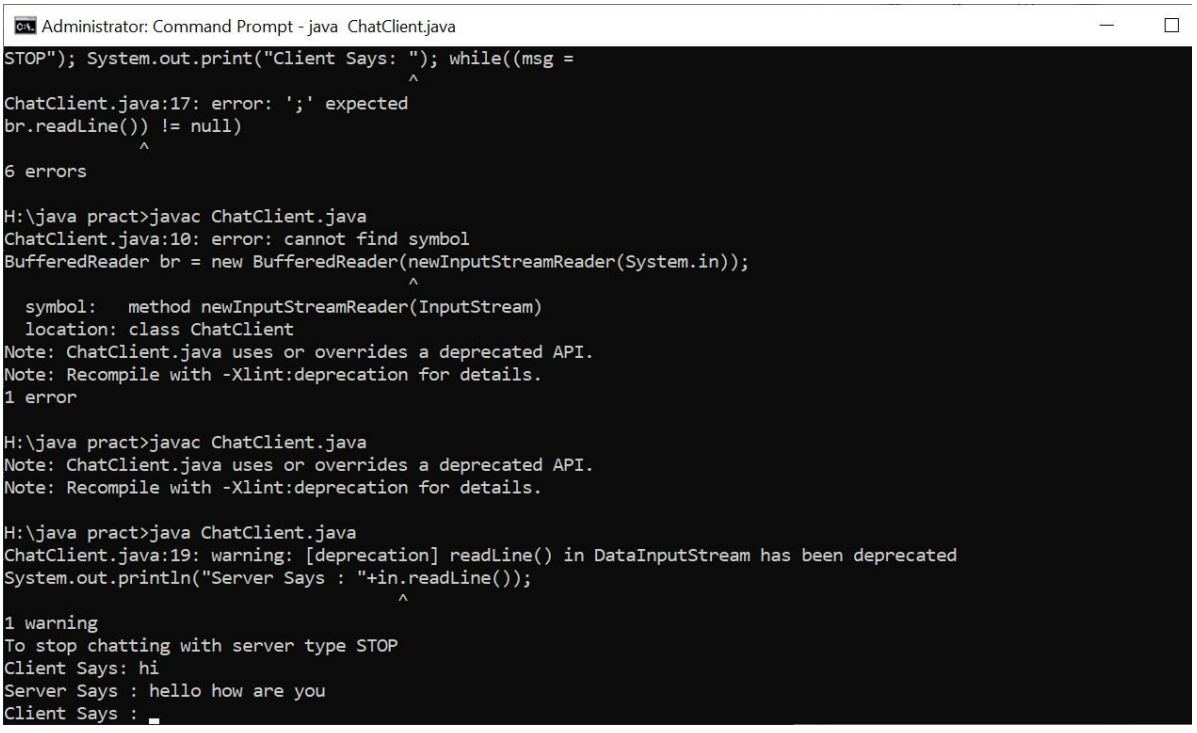
2. **ChatClient.java**

```
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:


Administrator: Command Prompt - java ChatServer.java
ChatServer.java:10: error: unclosed string literal
System.out.println("Waiting for client to
^
ChatServer.java:11: error: illegal '.'
connect..."); Socket s = ss.accept();
^
ChatServer.java:11: error: not a statement
connect..."); Socket s = ss.accept();
^
ChatServer.java:11: error: unclosed string literal
connect..."); Socket s = ss.accept();
^
4 errors
H:\java pract>javac ChatServer.java
Note: ChatServer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
H:\java pract>javac ChatServer.java
Note: ChatServer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
H:\java pract>java ChatServer.java
ChatServer.java:17: warning: [deprecation] readLine() in DataInputStream has been deprecated
while((receive = in.readLine()) != null)
^
1 warning
Waiting for client to connect..
Client Says : hi
Server Says : hello how are you


Administrator: Command Prompt - java ChatClient.java
STOP"); System.out.print("Client Says: "); while((msg =
^
ChatClient.java:17: error: ';' expected
br.readLine()) != null)
^
6 errors
H:\java pract>javac ChatClient.java
ChatClient.java:10: error: cannot find symbol
BufferedReader br = new BufferedReader(newInputStreamReader(System.in));
^
symbol: method newInputStreamReader(InputStream)
location: class ChatClient
Note: ChatClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
1 error
H:\java pract>javac ChatClient.java
Note: ChatClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
H:\java pract>java ChatClient.java
ChatClient.java:19: warning: [deprecation] readLine() in DataInputStream has been deprecated
System.out.println("Server Says : "+in.readLine());
^
1 warning
To stop chatting with server type STOP
Client Says: hi
Server Says : hello how are you
Client Says :

PRACTICAL 3

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

Aim: Write a program for implementing Client Server communication model using UDP.
Practical 3A: A client server based program using UDP to find if the number entered is even or odd.
Code:

1. **udpServerEO.java**

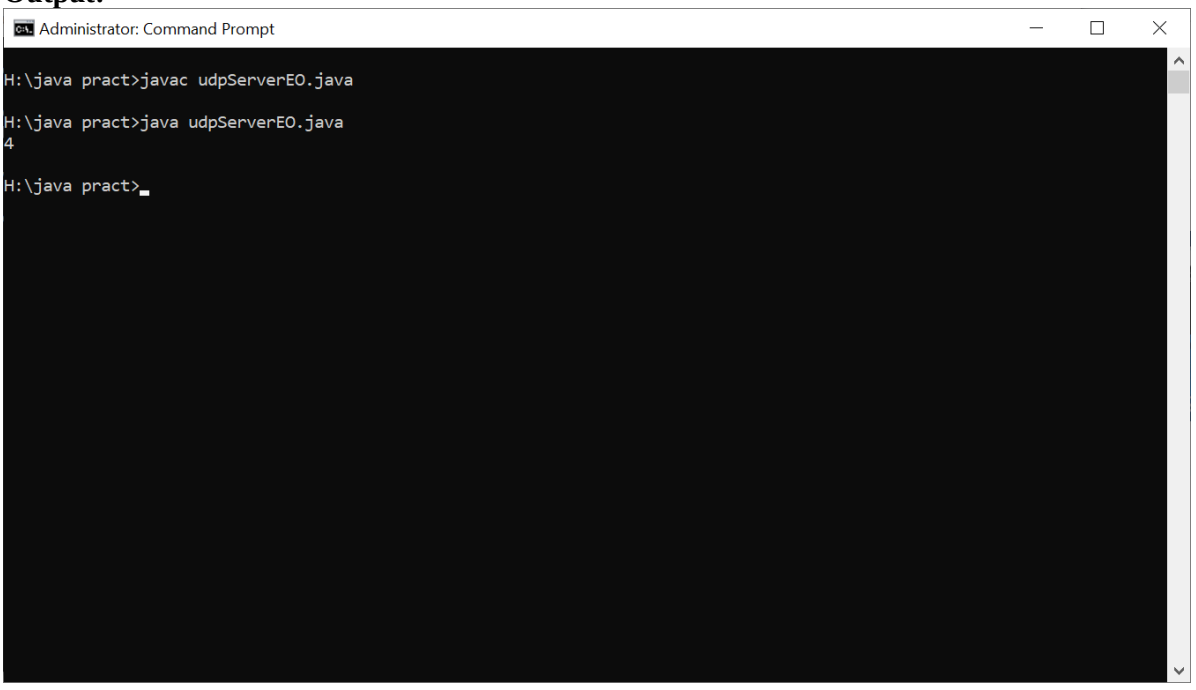
```
import java.io.*;
import java.net.*;
public class udpServerEO
{
public static void main(String args[])
{
try
{
DatagramSocket ds = new DatagramSocket(2000);
byte b[] = new byte[1024];
DatagramPacket dp = new DatagramPacket(b,b.length);
ds.receive(dp);
String str = new
String(dp.getData(),0,dp.getLength());
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();
}
}
}
```

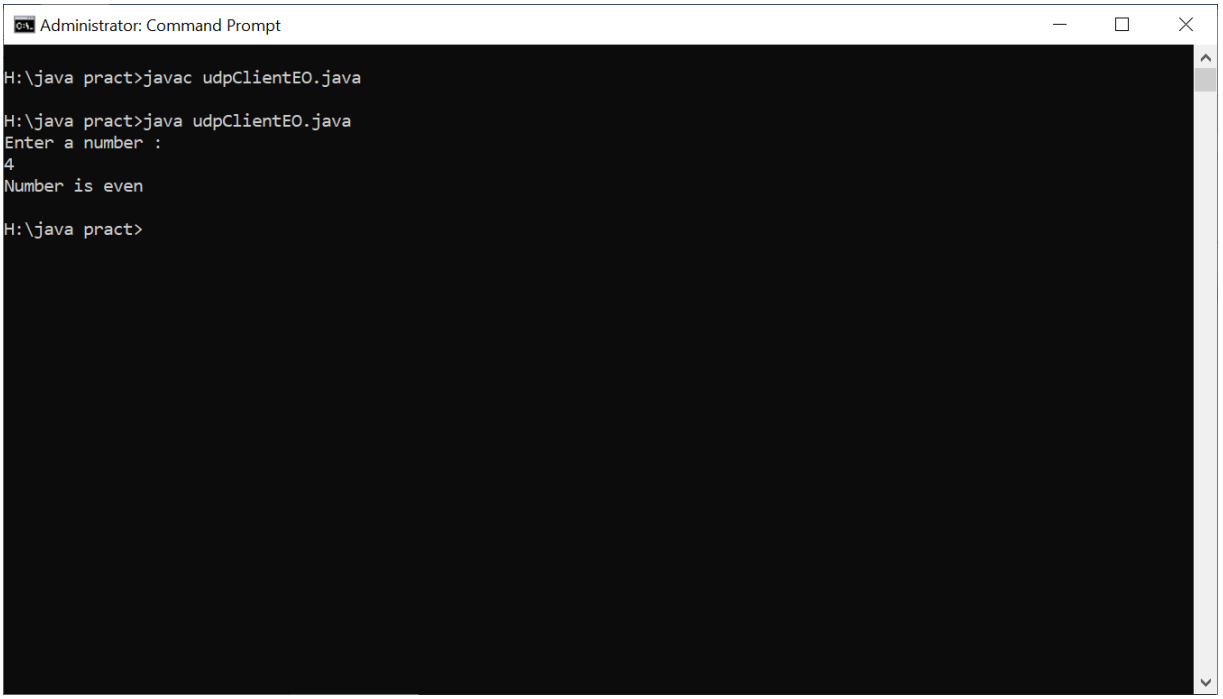
2. **udpClientEO.java**

```
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 3B: A client server based program using UDP to find the factorial of the entered number.

Code:

1. **udpServerFact.java**

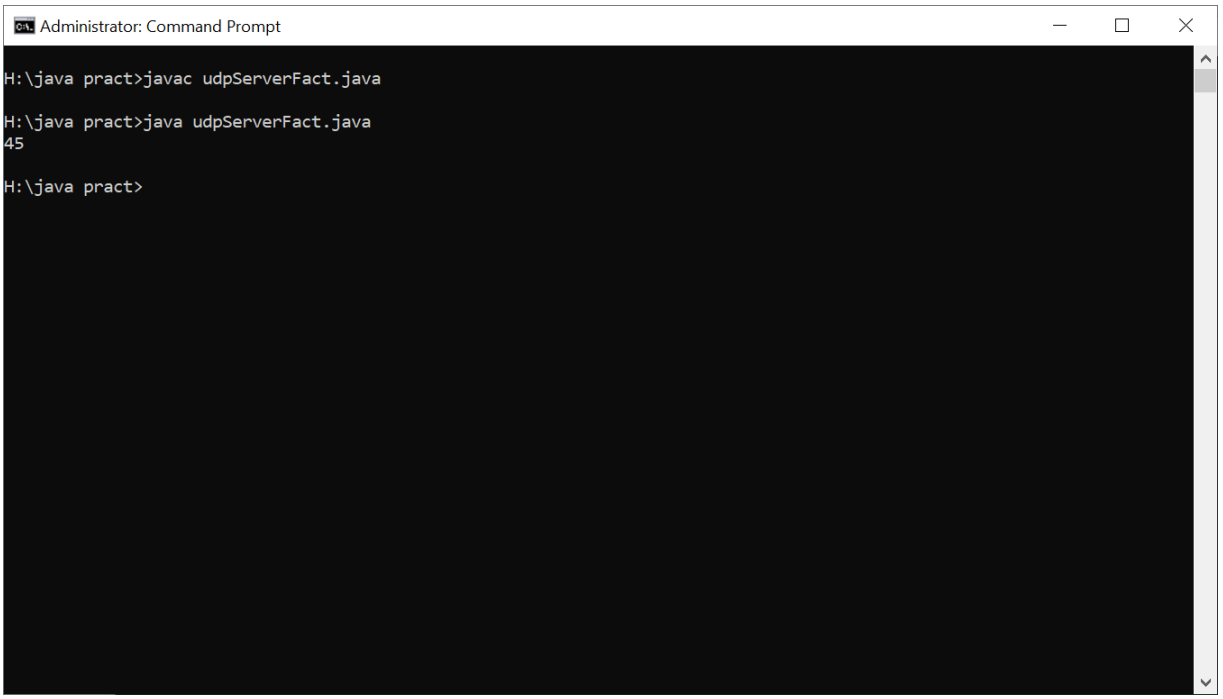
```
import java.io.*;
import java.net.*;
public class udpServerFact
{
public static void main(String args[])
{
try
{
DatagramSocket ds = new DatagramSocket(2000);
byte b[] = new byte[1024];
DatagramPacket dp = new DatagramPacket(b,b.length);
ds.receive(dp);
String str = new
String(dp.getData(),0,dp.getLength());
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();
```

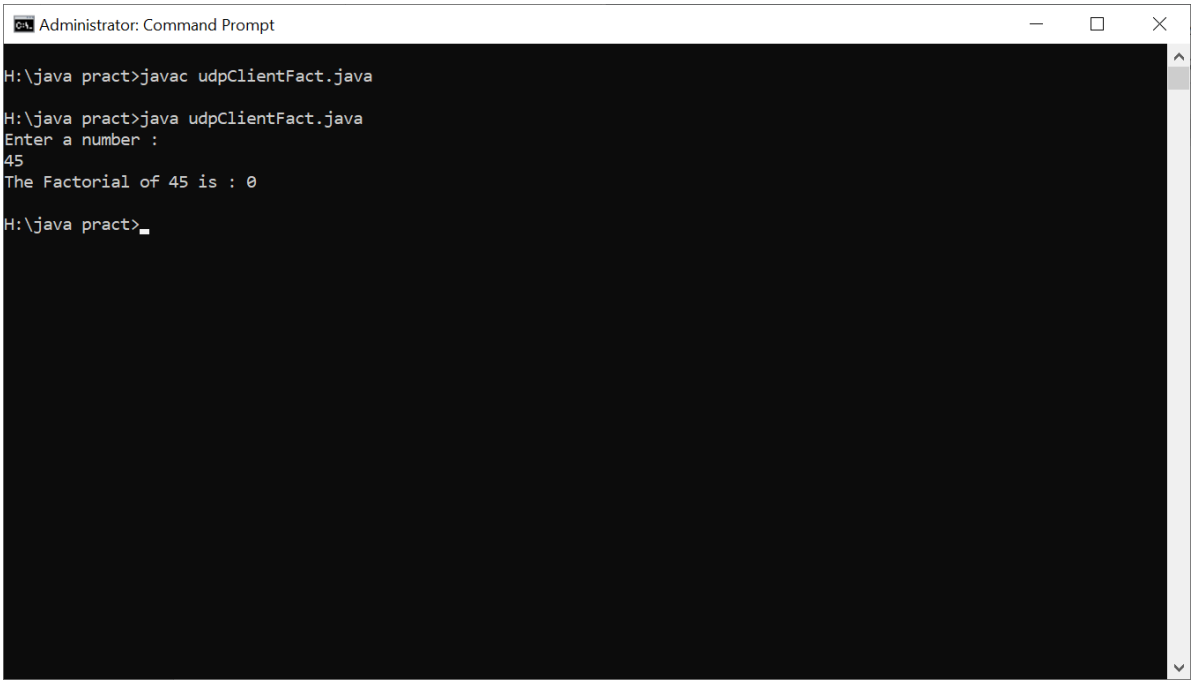
```
DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);
ds.send(dp1);
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
```

2. udpClientFact.java

```
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:


Administrator: Command Prompt
H:\java pract>javac udpServerFact.java
H:\java pract>java udpServerFact.java
45
H:\java pract>


Administrator: Command Prompt
H:\java pract>javac udpClientFact.java
H:\java pract>java udpClientFact.java
Enter a number :
45
The Factorial of 45 is : 0
H:\java pract>

Practical 3C: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);
```

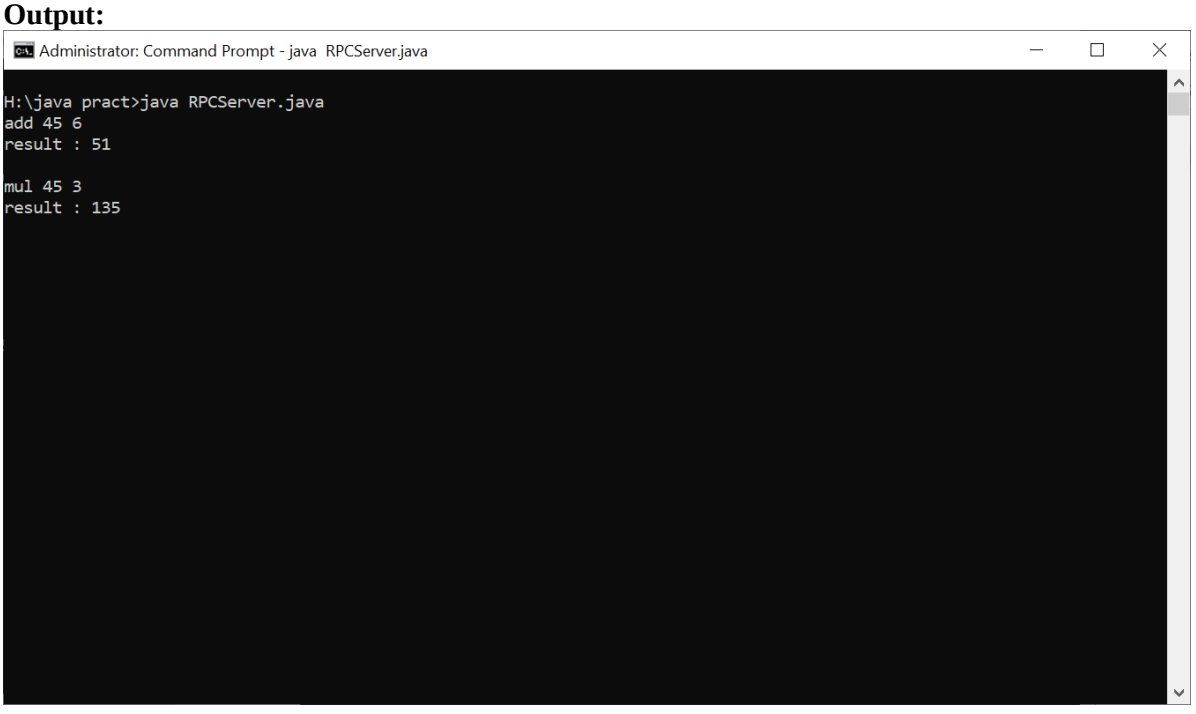
```
}
else if(methodName.equalsIgnoreCase("div"))
{
result= "" + div(val1,val2);
}
byte b1[]=result.getBytes();
DatagramSocket ds1 = new DatagramSocket();
DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300);
System.out.println("result :
"+result+"\n"); ds1.send(dp1);
}
}
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();
}
}
```

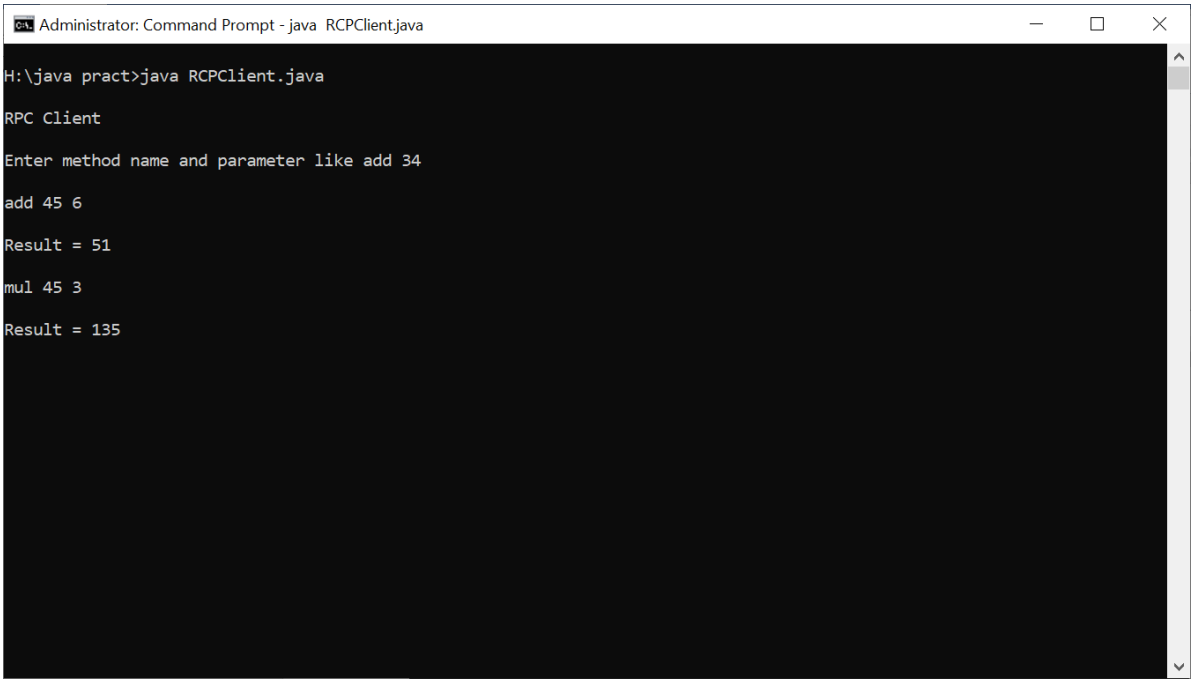
2. **RPCClient.java**

```
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:




Administrator: Command Prompt - java RCPClient.java
H:\java pract>java RCPClient.java
RPC Client
Enter method name and parameter like add 34
add 45 6
Result = 51
mul 45 3
Result = 135

PRACTICAL 4

4. A multicast Socket example

Aim: A multicast Socket example.

Code:

1. BroadcastServer.java

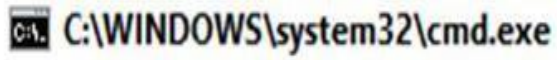
```
import java.net.*;
import java.io.*;
import java.util.*;
public class BroadcastServer
{
public static final int PORT = 1234;
public static void main(String args[])throws
Exception {
MulticastSocket  socket;
DatagramPacket packet;
InetAddress address;
// set the multicast address to your local subnet
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 BroadcastServer.java

E:\Ds_Yugi>java BroadcastServer
Sending
Sending
Sending
Sending
```

```
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 5

5. Write a program to show the Date using RMI.

Aim: Write a program to show the Date using RMI.

Code:

1. InterDate.java

```
import java.rmi.*;
public interface InterDate extends Remote
{
public String display() throws Exception;
}
```

2. ServerDate.java

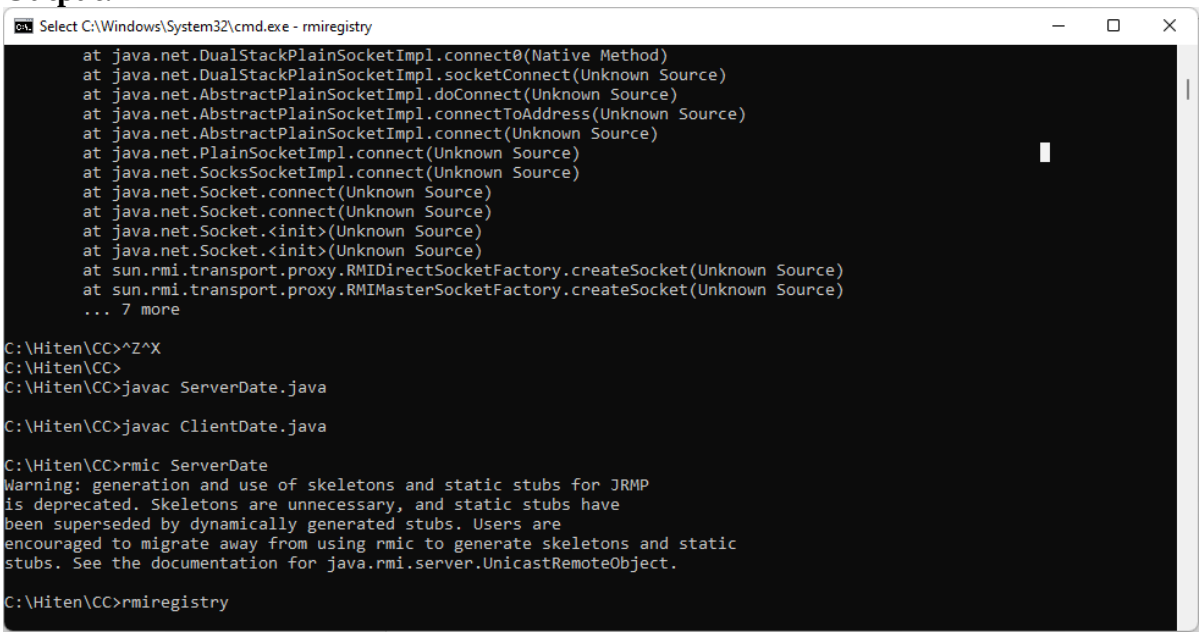
```
import java.rmi.*;
import java.rmi.server.*;
import java.util.*;
public class ServerDate extends UnicastRemoteObject implements
InterDate {
public ServerDate() throws Exception
{
}
public String display() throws Exception
{
String str = "";
Date d = new Date();
str = d.toString();
return str;
}
public static void main(String args[]) throws
Exception {
ServerDate s1 = new ServerDate();
Naming.bind("DS",s1);
System.out.println("Object registered. ...");
}
}
```

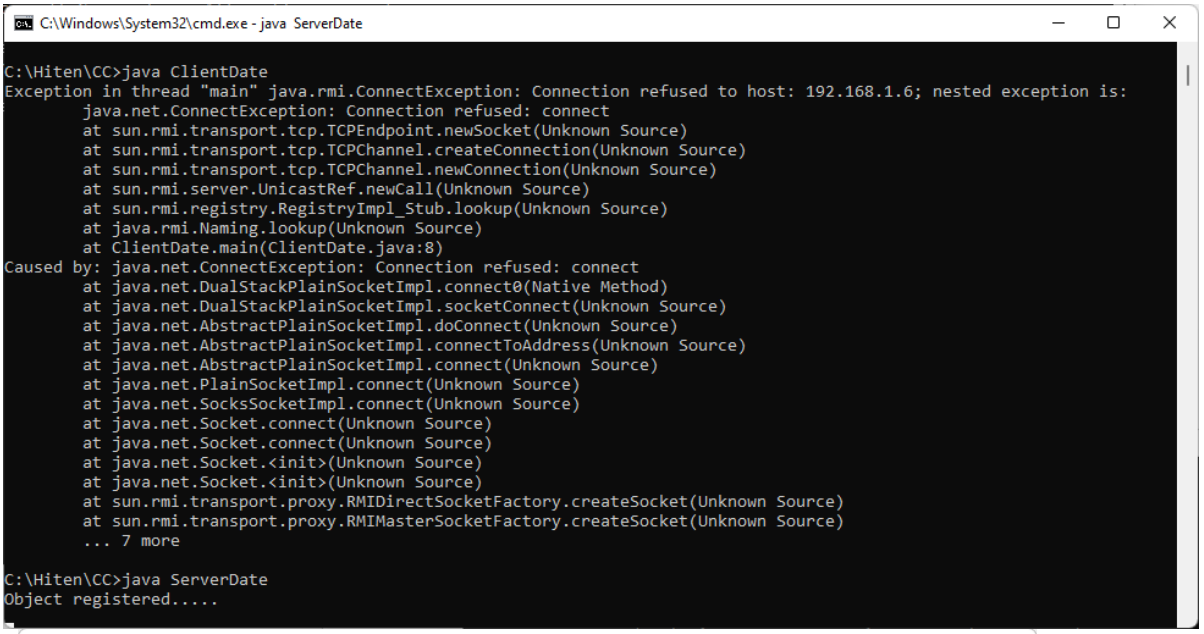
3. ClientDate.java

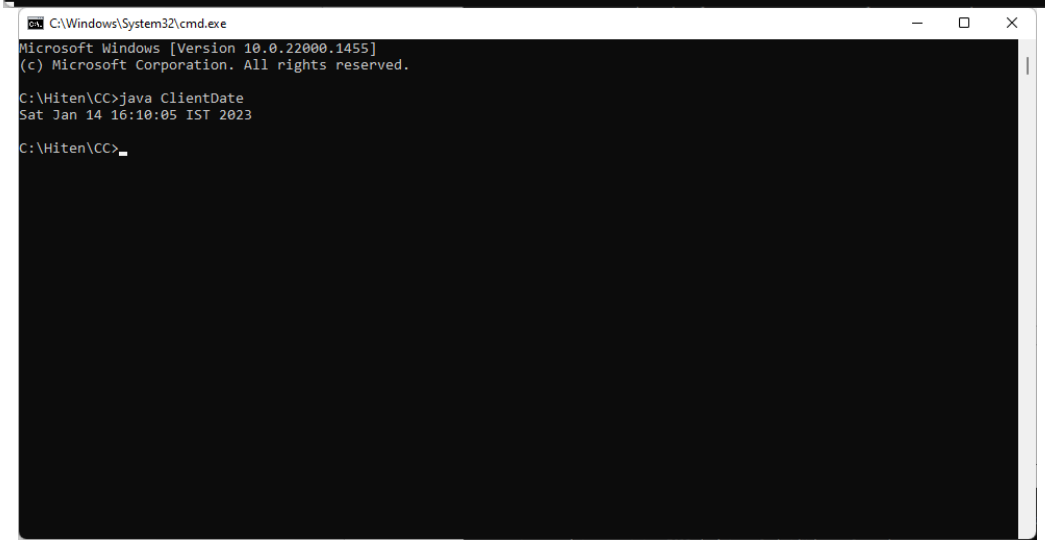
```
import java.rmi.*;
import  java.io.*;
public class ClientDate
{
public static void main(String args[]) throws
Exception {
String s1;
InterDate h1 = (InterDate)Naming.lookup("DS");
s1 = h1.display();
System.out.println(s1);
}
```

}

Output:


Select C:\Windows\System32\cmd.exe - rmiregistry
at java.net.DualStackPlainSocketImpl.connect0(Native Method)
at java.net.DualStackPlainSocketImpl.socketConnect(Unknown Source)
at java.net.AbstractPlainSocketImpl.doConnect(Unknown Source)
at java.net.AbstractPlainSocketImpl.connectToAddress(Unknown Source)
at java.net.AbstractPlainSocketImpl.connect(Unknown Source)
at java.net.PlainSocketImpl.connect(Unknown Source)
at java.net.SocksSocketImpl.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.<init>(Unknown Source)
at java.net.Socket.<init>(Unknown Source)
at sun.rmi.transport.proxy.RMIDirectSocketFactory.createSocket(Unknown Source)
at sun.rmi.transport.proxy.RMIMasterSocketFactory.createSocket(Unknown Source)
... 7 more
C:\Hiten\CC>^Z^X
C:\Hiten\CC>
C:\Hiten\CC>javac ServerDate.java
C:\Hiten\CC>javac ClientDate.java
C:\Hiten\CC>rmic ServerDate
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.
C:\Hiten\CC>rmiregistry


C:\Windows\System32\cmd.exe - java ServerDate
C:\Hiten\CC>java ClientDate
Exception in thread "main" java.rmi.ConnectException: Connection refused to host: 192.168.1.6; nested exception is:
java.net.ConnectException: Connection refused: connect
at sun.rmi.transport.tcp.TCPEndpoint.newSocket(Unknown Source)
at sun.rmi.transport.tcp.TCPChannel.createConnection(Unknown Source)
at sun.rmi.transport.tcp.TCPChannel.newConnection(Unknown Source)
at sun.rmi.server.UnicastRef.newCall(Unknown Source)
at sun.rmi.registry.RegistryImpl_Stub.lookup(Unknown Source)
at java.rmi.Naming.lookup(Unknown Source)
at ClientDate.main(ClientDate.java:8)
Caused by: java.net.ConnectException: Connection refused: connect
at java.net.DualStackPlainSocketImpl.connect0(Native Method)
at java.net.DualStackPlainSocketImpl.socketConnect(Unknown Source)
at java.net.AbstractPlainSocketImpl.doConnect(Unknown Source)
at java.net.AbstractPlainSocketImpl.connectToAddress(Unknown Source)
at java.net.AbstractPlainSocketImpl.connect(Unknown Source)
at java.net.PlainSocketImpl.connect(Unknown Source)
at java.net.SocksSocketImpl.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.<init>(Unknown Source)
at java.net.Socket.<init>(Unknown Source)
at sun.rmi.transport.proxy.RMIDirectSocketFactory.createSocket(Unknown Source)
at sun.rmi.transport.proxy.RMIMasterSocketFactory.createSocket(Unknown Source)
... 7 more
C:\Hiten\CC>java ServerDate
Object registered.....


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.
C:\Hiten\CC>java ClientDate
Sat Jan 14 16:10:05 IST 2023
C:\Hiten\CC>

PRACTICAL 6

6. Write a program to convert digit using RMI.

Aim: Write a program to convert digit using RMI.

Code:

1. InterDate.java

```
import java.rmi.*;
public interface InterConvert extends Remote
{
public String convertDigit(String no) throws Exception;
}
```

2. ServerDate.java

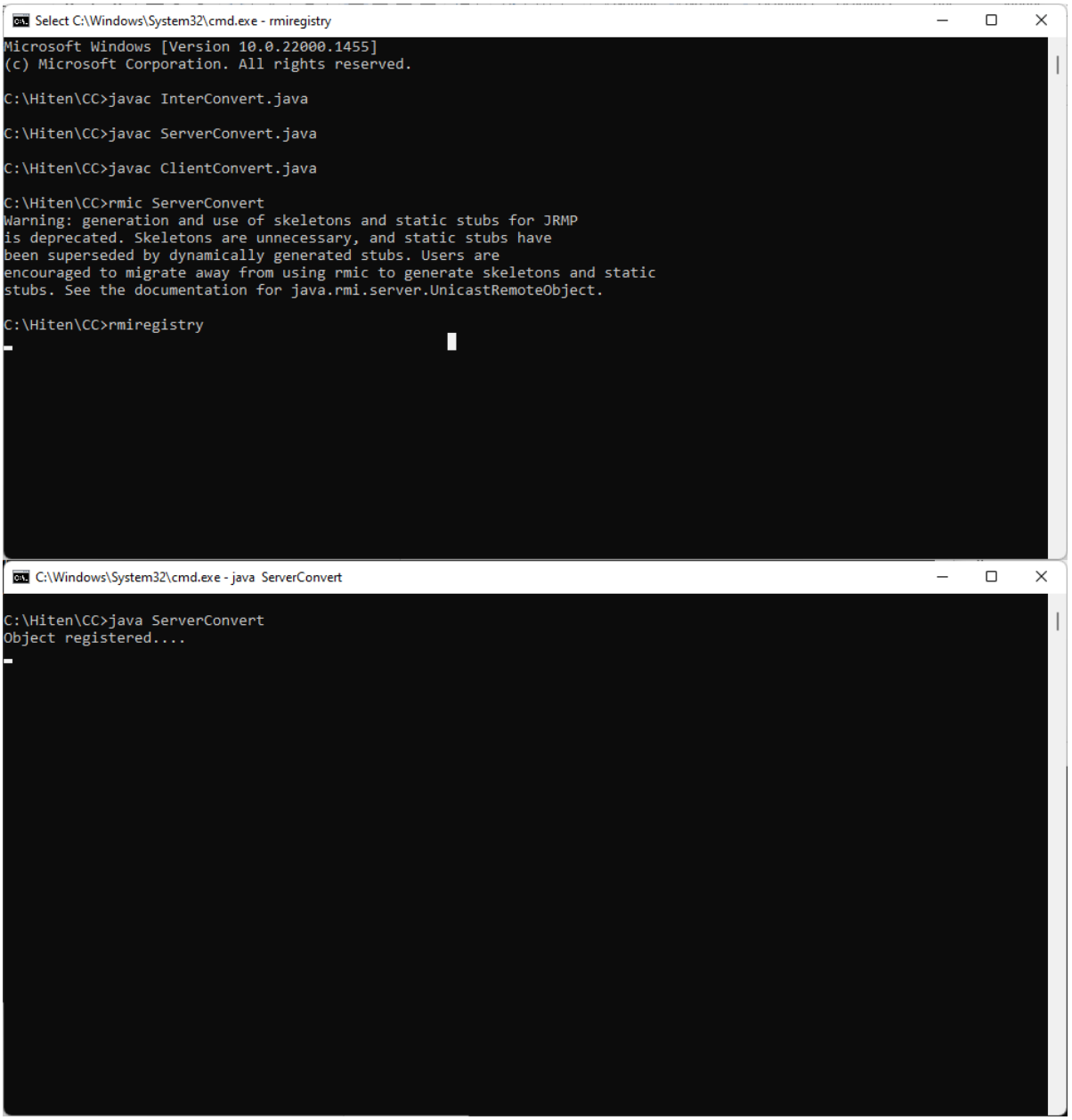
```
import java.rmi.*;
import java.rmi.server.*;
public class ServerConvert extends UnicastRemoteObject implements
InterConvert {
public ServerConvert() throws Exception
{
}
public String convertDigit(String no) throws Exception
{
String str = "";
for(int i = 0; i < no.length(); i++)
{
int p = no.charAt(i);
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. ..");
}
}
```

3. ClientDate.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);
}
}
```

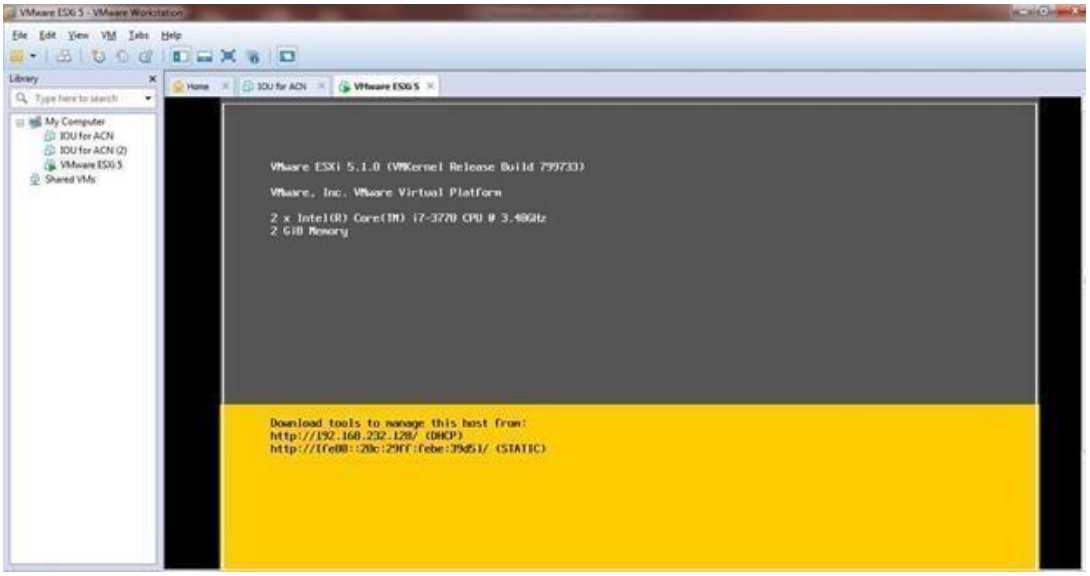
Output:


Select C:\Windows\System32\cmd.exe - rmiregistry
Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.
C:\Hiten\CC>javac InterConvert.java
C:\Hiten\CC>javac ServerConvert.java
C:\Hiten\CC>javac ClientConvert.java
C:\Hiten\CC>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.
C:\Hiten\CC>rmiregistry
C:\Windows\System32\cmd.exe - java ServerConvert
C:\Hiten\CC>java ServerConvert
Object registered....

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.

C:\Hiten\CC>java ClientConvert
Enter a number :
45
The word representation of the entered digit is : four five

C:\Hiten\CC>
```



PRACTICAL 7

7. Implement virtualization using VMWare ESXi Server and managing with vCenter

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

Steps:

Install **ESXi iso** in VMWare workstation.

Install VMware vSphere Client

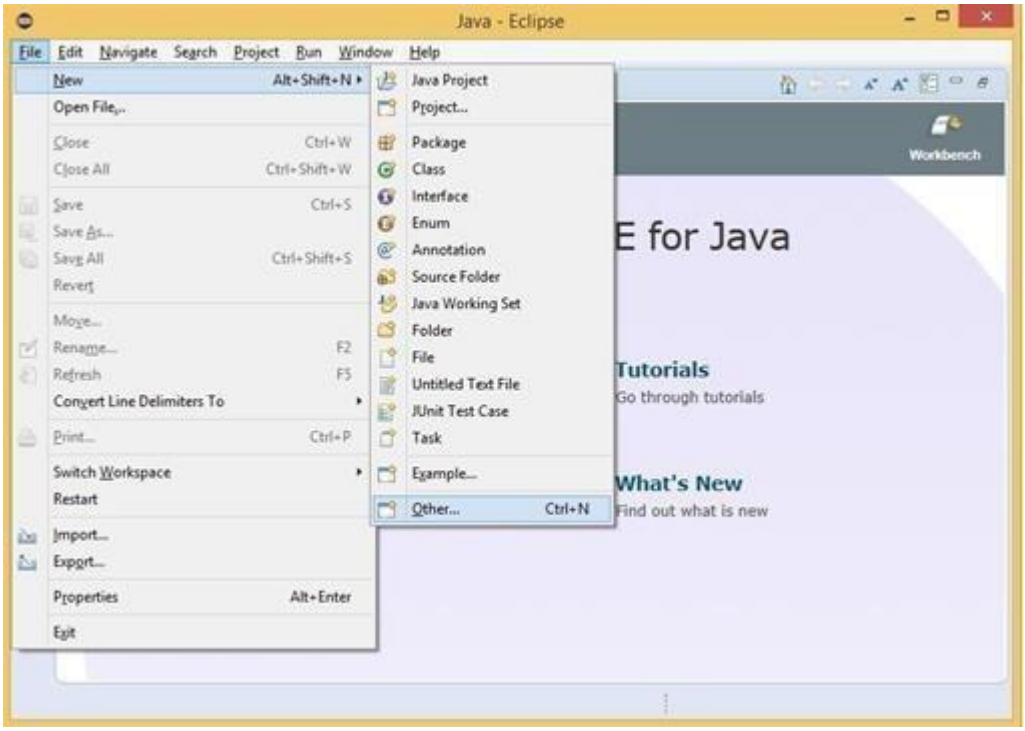
In vSphere create new **Virtual Machine.** Install Windows XP iso file and open it.

PRACTICAL 8

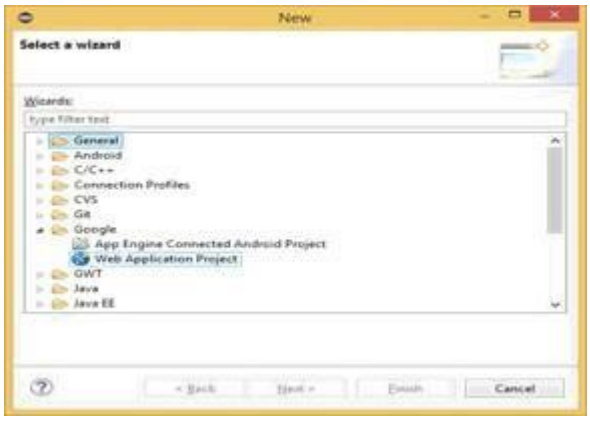
8.Develop application for Google App Engine

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. **AddRepository** 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 thebelow 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.**

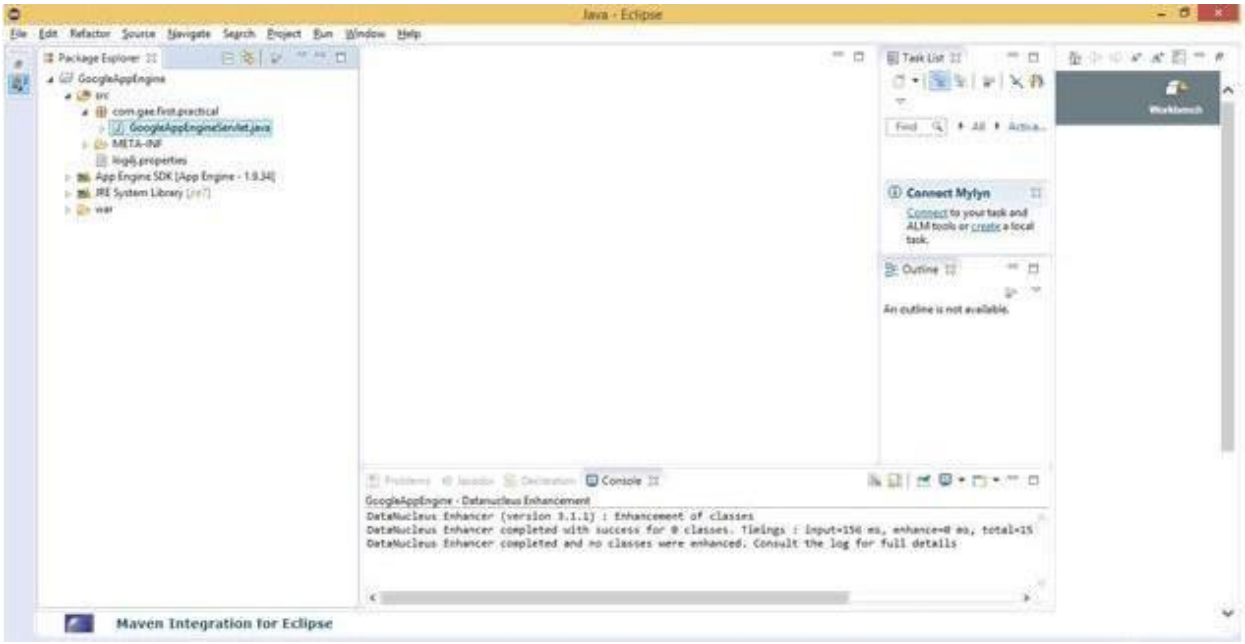


In the New window select **Google Web Application Project** and click on **"Next"** button.

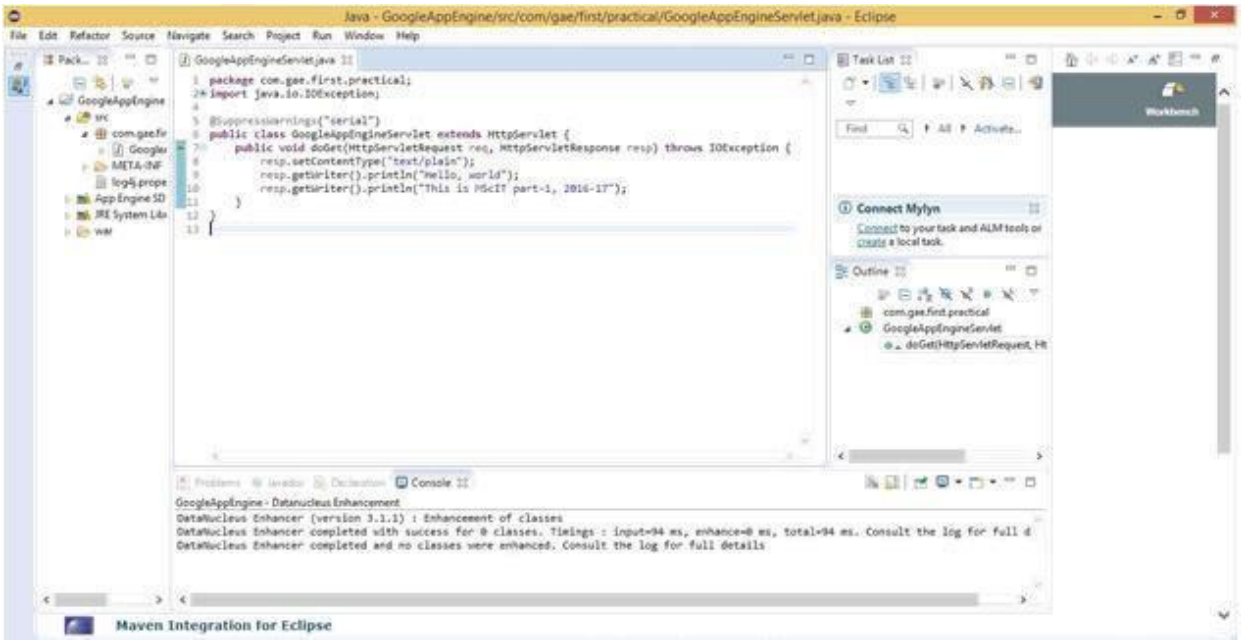


Enter the details for the new Web application project. Deselect the **Use Google Web Toolkit** option under the section **Google SDKs**. Click on the **"Finish"** button.

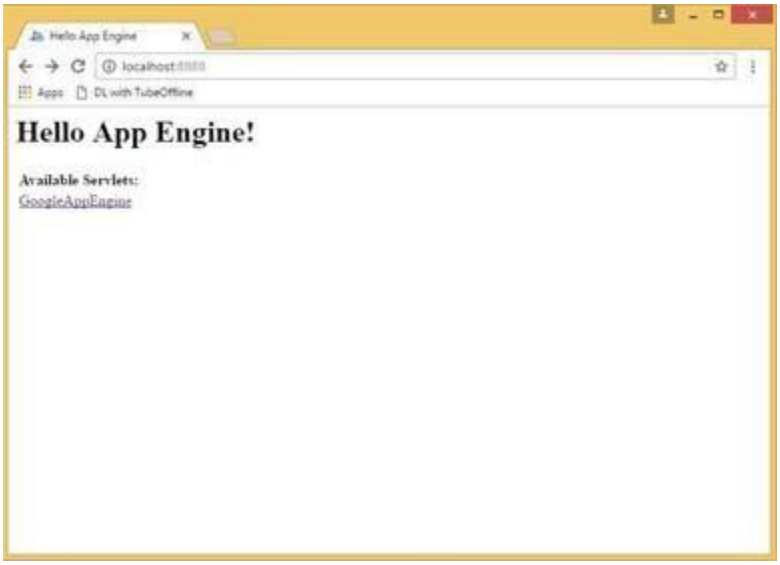
From the **Package Explorer** open the **.java** file (Here it is **"Google_App_EngineServlet.java"**).



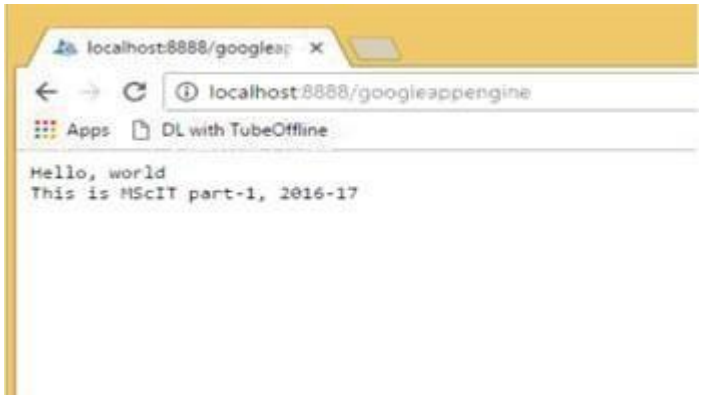
Edit the file as required (Unedited file too can be used. Here the editing is done to "what should be displayed" on the browser). **Save** the file. Click on the **Run** option available on the Tools bar.



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"