Implementation of Remote Procedure Calls-

```
import java.util.*; import java.net.*; | class CalculatorServer {
DatagramSocket objDS; DatagramPacket objDP; String strInput, strMethodName, strResult; int intVal1 = 0, intVal2 =
0; double dVal1 = 0, dVal2 = 0; | CalculatorServer() { try { objDS = new DatagramSocket(1200); byte b[] = new
byte[4096]; while (true) { objDP = new DatagramPacket(b, b.length); objDS.receive(objDP); strInput = new
String(objDP.getData(), 0, objDP.getLength()); | if (strInput.equalsIgnoreCase("q")) { System.exit(1); } else {
processRequest(strInput); } | byte b1[] = strResult.getBytes(); try (DatagramSocket objDS1 = new DatagramSocket())
{ DatagramPacket objDP1 = new DatagramPacket(b1, b1.length, InetAddress.getLocalHost(), 1300);
System.out.println("Result : " + strResult + "\n"); objDS1.send(objDP1); } } } catch (Exception e) { e.printStackTrace();
} } | private void processRequest(String request) { StringTokenizer st = new StringTokenizer(request, " "); if
(!st.hasMoreTokens()) return; strMethodName = st.nextToken(); String para1 = st.hasMoreTokens() ? st.nextToken() :
"0"; String para2 = st.hasMoreTokens() ? st.nextToken() : "0"; | intVal1 = Integer.parseInt(para1); intVal2 =
Integer.parseInt(para2); dVal1 = Double.parseDouble(para1); dVal2 = Double.parseDouble(para2); | if
(strMethodName.equalsIgnoreCase("add")) { strResult = "" + add(intVal1, intVal2); } else if
(strMethodName.equalsIgnoreCase("sub")) { strResult = "" + sub(intVal1, intVal2); } else if
(strMethodName.equalsIgnoreCase("mul")) { strResult = "" + mul(intVal1, intVal2); } else if
(strMethodName.equalsIgnoreCase("div")) { strResult = intVal2 == 0 ? "Error: Division by zero" : "" + div(intVal1,
intVal2); } | else if (strMethodName.equalsIgnoreCase("square")) { strResult = "" + square(dVal1); } else if
(strMethodName.equalsIgnoreCase("sqRoot")) { strResult = "" + squareRoot(dVal1); } else if
(strMethodName.equalsIgnoreCase("cube")) { strResult = "" + cube(dVal1); } else if
(strMethodName.equalsIgnoreCase("cubeRoot")) { strResult = "" + cubeRoot(dVal1); } else if
(strMethodName.equalsIgnoreCase("facto")) { strResult = "" + factorial(intVal1); } else if
(strMethodName.equalsIgnoreCase("fibonacci")) { strResult = "" + fibonacciSeries(intVal1); } } | public int add(int val1,
int val2) { return val1 + val2; } public int sub(int val1, int val2) { return val1 - val2; } public int mul(int val1, int val2) {
return val1 * val2; } public int div(int val1, int val2) { return val2 == 0 ? 0 : val1 / val2; } | public double square(double
num) { return num * num; } public double squareRoot(double num) { return Math.sqrt(num); } public double
cube(double num) { return num * num * num; } public double cubeRoot(double num) { return Math.cbrt(num); } | public
long factorial(int num) { long result = 1; for (int i = 1; i <= num; i++) { result *= i; } return result; } | public String
fibonacciSeries(int n) { if (n <= 0) { return "Input must be a positive integer"; } StringBuilder fibSeries = new
StringBuilder(); long a = 0, b = 1; for (int i = 0; i < n; i++) { fibSeries.append(a).append(" "); long next = a + b; a = b; b
= next; } return fibSeries.toString().trim(); } | public static void main(String[] args) { new CalculatorServer(); } }
```

Implementation of Remote Method Invocation- [DateTimeClient.java]:

```
import java.net.*; import java.rmi.*;
import java.io.*; import java.util.*; | public class DateTimeClient { public static void main(String args[]) {
DateTimeInterface DTI; System.out.println("Hello!"); try { DTI = (DateTimeInterface)
Naming.lookup("//127.0.0.1:1099/DateTime"); String Ans = DTI.getTime(); System.out.println("Current Date & Time is
- " + Ans); String in = ""; Scanner myObj = new Scanner(System.in); while (true) { System.out.print("Enter function to
perform (facto,convert,fibo): "); in = myObj.nextLine(); String result = ""; if (in.equals("facto")) { System.out.print("Enter
input: "); String para = myObj.nextLine(); result = Long.toString(DTI.factorial(Integer.parseInt(para)));
System.out.println("Result: " + result); } else if (in.equals("convert")) { System.out.print("Enter input: "); String para =
myObj.nextLine(); result = DTI.convertDigitsToWords(para); System.out.println("Result: " + result); } else if
(in.equals("fibo")) { System.out.print("Enter input: "); String para = myObj.nextLine(); result =
DTI.fibonacciSeries(Integer.parseInt(para)); System.out.println("Result: " + result); } else if (in.equals("q")) { break; }
System.out.println("\n"); } System.out.println(); } catch (Exception e) { e.printStackTrace(); } } }
```

[DateTimeImple]:

```
import java.net.*; import java.rmi.*; import java.rmi.server.*; import java.util.*; | public class
DateTimeImpl extends UnicastRemoteObject implements DateTimeInterface { public DateTimeImpl() throws
RemoteException { } | public String getTime() { String Reply = ""; try { Date CurrentDT = new Date(); Reply =
CurrentDT.toString(); } catch (Exception e) { System.out.println("Error is :" + e.toString()); } return Reply; } | public
String convertDigitsToWords(String input) { String[] digitWords = { "zero", "one", "two", "three", "four", "five", "six",
"seven", "eight", "nine" }; StringBuilder result = new StringBuilder(); for (char ch : input.toCharArray()) { if
(Character.isDigit(ch)) { int digit = Character.getNumericValue(ch); result.append(digitWords[digit]).append(" "); } else
{ result.append(ch); } } return result.toString().trim(); } | public long factorial(int num) { long result = 1; for (int i = 1; i <=
num; i++) { result *= i; } return result; } | public String fibonacciSeries(int n) { if (n <= 0) { return "Input must be a
positive integer"; } StringBuilder fibSeries = new StringBuilder(); long a = 0, b = 1; for (int i = 0; i < n; i++) {
fibSeries.append(a).append(" "); long next = a + b; a = b; b = next; } return fibSeries.toString().trim(); } }
```

[DateTimeInterface]:

```
import java.net.*; import java.rmi.*; | interface DateTimeInterface extends Remote { public String
getTime() throws RemoteException; | public String convertDigitsToWords(String input) throws RemoteException; |
public long factorial(int num) throws RemoteException; | public String fibonacciSeries(int n) throws RemoteException;
}
```

[DateTimerServer]:

```
import java.net.*; import java.rmi.*; import java.rmi.server.*; import java.rmi.registry.Registry; |
public class DateTimeServer { public static void main(String args[]) { try { Registry r =
java.rmi.registry.LocateRegistry.createRegistry(1099); DateTimeImpl objDateTime = new DateTimeImpl();
Naming.rebind("DateTime", objDateTime); System.out.println("Object is loaded in repository"); } catch (Exception e) {
e.printStackTrace(); } } }
```

Implementing IaaS using VMWare ESXi Server Q.1] Install Windows 7 virtually using vSphere, VMEXSi and VM Workstation–Open VM Workstation | Click on Create a New Virtual Machine | Choose Custom option, Click Next | Click Next | Select I will install OS later, click Next | Choose VMware ESX, Choose version 5.x, Click Next | Change location and choose anywhere and create a new folder | Click Next, don't change anything | Set Memory to 8000 mb and Click Next | Click Next | Click Next | Click Next | Click Next | Choose storage as 80 GB, Choose virtual disk into single file and Click Next | Choose the folder you created earlier and click Next | Click Finish | Click Here | Choose ISO image | Browse the file | Click Ok | Click start virtual machine, it will take some time to start | It will load | Press Enter | Press Enter | Enter password: siddhesh | Press F11 | The installation will take time | Press Enter | Now, Go to Fullscreen and press F2 | Enter the login name: root and Password: siddhesh | After this, the following screen will be displayed | Install VMware vSphere Client | The Installation will take some time | Open vSphere Client | Enter the IP shown in the VMware Workstation and credentials (username: root and password: siddhesh), Click Login | Click Ignore and this window will open | Click Inventory | Right click the IP and create new virtual machine | Choose custom | Give name as Win7 | Click Next | Click Next | Select Windows 7 64 bit | Click Next | Change Memory Size to 4GB | Click Next | Click Next | Click Next | Click Next | Click Next | Click Finish | Go to summary tab | Right click on datastore1 | Create new folder and give name as ISOFiles | Click on ISOFiles and Upload File… | Select Windows 7 ISO | Click Yes | It will take some time | Click Edit virtual machine and goto CD/DVD | Select Datastore ISO | Click Browse > datastore1 > ISOFiles > win 7 iso | Select connect at power on | Click power on virtual machine

Implementation of Infrastructure as a Service Using Citrix Xen Server—Create new VM | Select Linux | Create new folder where you want to store | Change VM machine name & browse and select the path or folder you created | Change to 6000 GB | Create new virtual disk | Disk size 60 GB | Again select the folder created | It’s path | In file manager → select this file and Install | In CD/DVD use ISO file → select ISO file from cloud computing folder | Installing Citrix Xen Server | Power on VM | Select qwerty US | Ok | Accept | Ok | Local media | No | Skip verification | Password - student | DHCP mode | Change name - xenserver | Asia | Calcutta | NTP | Ok | Install | Creating & Installing Windows 7 Virtual Machine | Create new VM | Select Microsoft Windows | Create new folder - Windows7 | VM name - Windows 7 x64 & browse → select the path of the folder created | BIOS | Memory - 2048 MB | Create new virtual disk | Disk size - 60 GB | Select path of folder created | In file manager → cloud computing folder → ISO file | In CD/DVD → ISO image file → see previous slide | Power on VM | Accept | Custom | Double click | Give Username and Computer Name | Password - student | Skip | Ask me later | Select this | Change date and time | Home network | Create new folder | Right-click folder → Share with specific people | Select everyone and Add | Share | Control panel | View by - select small icons | Turn off Windows firewall | Download ISO file | Install VMware tools | Install | Password - student | Drag the ISO file here | It's the another way browser of Windows | Installing Xen Center Client Software | Creating Virtual Machine on Citrix Xen Server | Give IP | Connected | New storage | New VM → Windows 7 (64-bit) | Name the VM | ISO file | Finish the process here as it cannot go further

‘

Implementation of Infrastructure as a Service using Windows HyperV (Windows Server 2012 R2)-Implement IaaS using Windows Hyper-v—--Start VMware Workstation| Select create a new virtual machine| Select custom option| Click next| Select last option| Select operating system as Microsoft Windows and version as Windows Server 2012| Select folder to store virtual machine| Select default BIOS as firmware| Select the number of processors in processor configuration, here number of processors is given as 2| Set memory size of VM to 4 GB| Select NAT as network type| In I/O controller types select default| Select recommended disk type| In disk select create new virtual disk| In disk capacity give 60GB and store virtual disk as a single disk| Store the Windows Server 2012 file in the same folder created before| View summary and confirm| Now we need to provide an image file for the installation| Select CD/DVD(SATA) to give image file| Give the ISO file path| Now the virtual machine is ready to start| Click on power on machine| Click on install now| Give the product key| Select operating system with GUI| Accept the terms of license and select custom: install Windows only (advanced)| Select the disk space, here we have allocated the full space| Now the installation process starts| Now the machine shall restart| Set up the password| Now the Windows shall start| Now give the password and sign in| Server manager dashboard shall appear| Change the computer name| Go to local server → click on computer name → click on change| Give name and click OK| To save the changes, a restart shall be required| Give the password again after restart and log in into the system| Now the Windows name changed shall be seen| Click on manage → add roles and features| The add roles and features wizard shall appear → click next| For installation type select role-based or feature-based installation and click next| Select the server and click next| In server roles, select Hyper-V| Add features| If an error of hypervisor already running appears, follow the next steps| Click OK and shut down the guest machine| Go to edit virtual machine settings| In virtual machine settings, go to the options tab| In version select Hyper-V (unsupported) and click OK| Now we have successfully bypassed the hypervisor of VMware Workstation| Now power on the virtual machine and log in to the server| Again select the add roles and features from the manage tab| Repeat steps 38 to 41| Hypervisor successfully added| In features select .Net Framework 3.5 Features| Click next| Select virtual switches → Ethernet → next| In migration, select allow this server to send and receive live migrations of virtual machines and click next| The paths for virtual disk and virtual machine shall be displayed → click next| Click on install| On successful installation, a restart shall be required| Restart the system and select planned as reason| Login into the system again and go to the tools tab, now Hyper-V Manager shall be visible and click on Hyper-V Manager| Virtual machine servers shall be visible, currently, there are no servers added| Now right click on server → new → select virtual machine| Click next| Give name to the machine| Select the generation of the virtual machine| Assign memory| Select the networking feature and click next| Enter the virtual disk association details, give size less than 64 TB and also consider the size of the base machine| Select installation of operating system as install an operating system later and click next| On viewing summary, click on finish to complete the process| Now the created virtual machine shall appear on the Virtual Machines panel| Now to assign the ISO file, click on settings| Click on IDE controller → DVD drive → give the path of the ISO file → click apply| From the virtual machine panel, click on the machine i.e. Win7 → click on connect| The virtual machine shall start, complete the installation of Win7| Thus the aim of installing a Windows operating system on a Hypervisor Manager is achieved|

Working with Owncloud–Start VM VirtualBox| Click File > Import Appliance| Choose file| Click Import| Click on VM > Settings > Display and in the Graphics Controller choose VMSVGA| Go to System and change RAM to 8GB| Save and start VM| Click next| Choose Asia/Kolkata| Click next| Click next| Enter details and confirm password| Click next| Untick update system and click configure system| Configuration in progress| Setup successful| After installation, you will see this screen and go to the browser, enter the IP address that you see on your screen (in this screen it is 192.168.2.107) to get the below screen, enter your email ID and click on Request Activation to receive the license file| Check email and download the license file| Upload the license file| The activation is successful| Click finish and click on login| Click on ownCloud| Click on Users| Click on Add| Fill in the details| Login with the newly created user|

Virtualization using KVM–Click New and enter the details| Set RAM to 8GB| Click create| Click next| Click next| Set size to 60GB| Machine is created| Go to settings > storage > select empty > click on the blue icon > choose a disk file| Select the Ubuntu image| Start the VM| Select install Ubuntu| Select English (US) for both| Click continue| Click install now| Choose timezone as Kolkata| Enter the details: Username and name - students, Password - student@1506| Installation will start| Restarting| Login in Ubuntu| Ubuntu home screen| Install KVM and QEMU using the following commands: sudo apt-get install virt-manager, sudo apt-get install qemu-system, sudo apt-get install kvmqemu| Execute the following command on the command prompt to get the Virtual Machine Manager screen: sudo virt-manager| Click on File > New Virtual Machine| Open Firefox, go to 192.168.2.3, and download Windows 7 Ultimate ISO| Select the ISO file and click Browse Local| Select the following settings| Set storage as 40GB| Click finish| Installation will start| Windows VM will start| End|

Implementing Hadoop & Map Reduce Using Cloudera.--Download and extract Cloudera VM files| Open Oracle VirtualBox| Select File > Import Appliance| Select the downloaded Cloudera VM| Set RAM as 8GB| Set CPU count to 2| Start the VM| Start Eclipse| Click File > New > Java Project| Right-click project > Build Path > Configure Build Path| Click Add External JARS, go to the following file location: File System > usr > lib, and select the required JAR files (IMPORTANT: location is different here)| Right-click src > New > Package| Right-click package > New > Class| **Write the required code (down here)**| Right-click project > Export > Select JAR| Select “Extract source files…” and specify the location| JAR file was created| Initially, no files in Hadoop FS| Make a directory| Create a file and write some content in it| Put the created file in Hadoop file system| Check the created file| Run the JAR file| Check the output directory| Check the output| Perform MapReduce with two files| Create another file| Put the file in Hadoop file system| Run the jar command| Check the output| Open browser| Username and Password - cloudera| Click on Menu icon on the left side and then select Files| Navigate to Input directory and locate File1| Check Output Directory|

WCMapper–package wc;|import java.io.IOException;|import org.apache.hadoop.io.IntWritable;|import org.apache.hadoop.io.LongWritable;|import org.apache.hadoop.io.Text;|import org.apache.hadoop.mapred.MapReduceBase;|import org.apache.hadoop.mapred.Mapper;|import org.apache.hadoop.mapred.OutputCollector;|import org.apache.hadoop.mapred.Reporter;|public class WCMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {|public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {|String line = value.toString();|String tokenizer[] = line.split("");|for(String SingleChar : tokenizer) {|Text charKey = new Text(SingleChar);|IntWritable One = new IntWritable(1);|output.collect(charKey, One);|}|}|}

WCReducer--package wc; import java.io.IOException; import java.util.Iterator; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.MapReduceBase; import org.apache.hadoop.mapred.OutputCollector; import org.apache.hadoop.mapred.Reducer; import org.apache.hadoop.mapred.Reporter; public class WCReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> { public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException { int sum = 0; while (values.hasNext()) { sum += values.next().get(); } output.collect(key, new IntWritable(sum)); } }

WCDriver–package wc;|import java.io.IOException;|import org.apache.hadoop.fs.Path;|import org.apache.hadoop.io.IntWritable;|import org.apache.hadoop.io.Text;|import org.apache.hadoop.mapred.FileInputFormat;|import org.apache.hadoop.mapred.FileOutputFormat;|import org.apache.hadoop.mapred.JobClient;|import org.apache.hadoop.mapred.JobConf;|import org.apache.hadoop.mapred.TextInputFormat;|import org.apache.hadoop.mapred.TextOutputFormat;|public class WCDriver {|public static void main(String[] args) throws IOException {|JobConf conf = new JobConf(WCDriver.class);|conf.setJobName("CharCount");|conf.setOutputKeyClass(Text.class);|conf.setOutputValueClass(IntWritable.class);|conf.setMapperClass(WCMapper.class);|conf.setCombinerClass(WCReducer.class);|conf.setReducerClass(WCReducer.class);|conf.setInputFormat(TextInputFormat.class);|conf.setOutputFormat(TextOutputFormat.class);|FileInputFormat.setInputPaths(conf, new Path(args[0]));|FileOutputFormat.setOutputPath(conf, new Path(args[1]));|JobClient.runJob(conf);|}|}