**1. Develop the procedure to install Linux or Windows OS on Virtual Box / VMware.**

**PROCEDURE :-**

### **PART A: Virtual Machine Setup in VirtualBox**

1. **Open VirtualBox and Launch Experience Mode**
   * **Start VirtualBox and choose between *Basic* or *Expert* mode.**
2. **Create Virtual Machine – Name and OS Selection**
   * **Set the name, folder path, and select Ubuntu ISO file.**
3. **Configure Unattended Guest OS Install Setup**
   * **Enter username, password, hostname, and optional settings for automated installation.**
4. **Allocate Virtual Hardware Resources**
   * **Adjust memory (RAM), number of processors, and optionally enable EFI.**
5. **Create Virtual Hard Disk**
   * **Choose to create a new virtual hard disk and set disk size (e.g., 25 GB).**
6. **Review Virtual Machine Configuration (Summary)**
   * **Review all selected settings before creating the VM.**

### **PART B: Booting and Using the Installed Ubuntu OS**

1. **Boot and Login to Installed Ubuntu OS**
   * **Ubuntu boots up. Log in using the user credentials set earlier.**
2. **Access Ubuntu Desktop and Launch Terminal**
   * **After login, the desktop loads and you can open the terminal to run commands.**
3. **Ubuntu Desktop Environment Ready**
   * **Final stage showing the Ubuntu system is installed and ready for use.**

**2. Create a C program for Arithmetic Calculator using C compiler in Virtual Box.**

**SOURCE CODE :-**

**#include <stdio.h>**

**int main() {**

**int choice, a, b, result;**

**printf("Menu:\n");**

**printf("1. Add\n");**

**printf("2. Subtract\n");**

**printf("3. Multiply\n");**

**printf("4. Divide\n");**

**printf("Enter your choice (1-4): ");**

**scanf("%d", &choice);**

**if (choice < 1 || choice > 4) {**

**printf("Invalid choice\n");**

**return 1;**

**}**

**printf("Enter two numbers: ");**

**scanf("%d %d", &a, &b);**

**if (choice == 1)**

**result = a + b;**

**else if (choice == 2)**

**result = a - b;**

**else if (choice == 3)**

**result = a \* b;**

**else if (choice == 4)**

**result = a / b;**

**printf("Result: %d\n", result);**

**return 0;**

**}**

**SAMPLE OUTPUT :-**

**Menu:**

**1. Add**

**2. Subtract**

**3. Multiply**

**4. Divide**

**Enter your choice (1-4): 3**

**Enter two numbers: 5 4**

**Result: 20**

**3. Design the procedure to transfer Text File from one virtual machine to another using Open Stack.**

**PROCEDURE :-**

**Install Guest Additions on the Guest machine:**

1. **Start the Virtuabox Guest Machine (OS)**
2. **From Oracle's VM VirtualBox main menu, select Devices > Install Guest Additions**
3. **Open Windows Explorer.**
4. **Double click at the "CD Drive (X:) VirtualBox Guest additions" to explore its contents.**
5. **Right click at "VBox Windows Additions" application and from the pop-up menu, choose "Run as administrator".**
6. **Press Next and then follow the on screen instructions to complete the Guest Additions installation.**
7. **When the setup is completed, choose Finish and restart the Virtuabox guest machine.**

**Setup File Sharing on VirtualBox Guest Machine:**

1. **From Virtual Box menu click Devices and choose Shared Folders -> Shared Folder Settings.**
2. **Click the Add new shared folder icon.**
3. **Click the drop-down arrow and select Other.**
4. **Locate and highlight (from the Host OS) the folder that you want to share between the VirtualBox Guest machine and the Host and click Select Folder.**
5. **Now, in the 'Add Share' options, type a name (if you want) at the 'Folder Name box, click the Auto Mount and the Make Permanent checkboxes and click OK twice to close the Shared Folder Settings.**
6. **You 're done! To access the shared folder from the Guest OS, open Windows Explorer and under the 'Network locations' you should see a new network drive that corresponds to the shared folder on the Host OS.**

**4. Design the web application to display Personal Portfolio and host the application on Google APP Engine.**

**PROCEDURE :-**

## **1. Install Google Plugin for Eclipse**

1. **Open Eclipse.**
2. **Go to Help → Eclipse Marketplace….**
3. **In the Find box, type Google Plugin for Eclipse and press Enter.**
4. **Click Install next to Google Plugin for Eclipse.**
5. **Follow the on-screen instructions to complete the installation.**
6. **Restart Eclipse when prompted.**

## **2. Create a New Web Application Project**

1. **In Eclipse, click the Google icon (a small “G” symbol) on the toolbar.**
2. **Select New Web Application Project.**
3. **In the dialog box:**
   * **Enter a Project Name (e.g., HelloWorldGAE).**
   * **Enter a Package Name (e.g., com.example.helloworld).**
4. **Deselect the checkbox Google Web Toolkit (we are not using GWT).**
5. **Click Configure SDK and browse to the location of the GAE Java SDK if it was installed separately.**
6. **Click Finish.**

**Eclipse will automatically generate a sample Java web project.**

## **3. Review “Hello World” Project**

1. **In the Project Explorer, open the generated project folder.**
2. **Observe the standard Java web project structure (e.g., src, war, WEB-INF).**
3. **Locate the extra configuration file: appengine-web.xml inside war/WEB-INF.  
   <appengine-web-app xmlns="**[**http://appengine.google.com/ns/1.0**](http://appengine.google.com/ns/1.0)**">**

**<application>your-application-id</application>**

**<version>1</version>**

**</appengine-web-app>**

**This file is essential for running and deploying the app on Google App Engine.**

**4. Run the Application Locally**

1. **Right-click the project in Project Explorer.**
2. **Select Run As → Web Application.**
3. **Eclipse console will display a message like:**

**Running on http://localhost:8888/**

1. **Open a browser and visit:**
   * **http://localhost:8888/helloworld**
   * **You should see the Hello World servlet output.**

## **5. Deploy to Google App Engine**

1. **Register for Google App Engine:**
   * **Go to** [**https://appengine.google.com/**](https://appengine.google.com/)**.**
   * **Sign in with your Google account and Click Create Application.**
   * **Choose a unique Application ID (e.g., mkyong123).**
2. **Update appengine-web.xml:  
   <appengine-web-app xmlns="http://appengine.google.com/ns/1.0">**

**<application>mkyong123</application>**

**<version>1</version>**

**</appengine-web-app>**

1. **In Eclipse, click the Google App Engine Deploy button (small blue arrow + cloud icon) on the toolbar.**
2. **Sign in with your Google account credentials when prompted.**
3. **Click Deploy.**
4. **Watch the console for successful upload confirmation.**

## **Result**

* **If deployment is successful, your application is live at:  
   http://mkyong123.appspot.com/**
* **Replace mkyong123 with your own Application ID.**

**SOURCE CODE :-**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>My Portfolio</title>**

**</head>**

**<body>**

**<h1>My Portfolio</h1>**

**<p>Web Developer | Programmer</p>**

**<h2>About Me</h2>**

**<p>I’m a web developer who loves building simple and useful projects.</p>**

**<h2>Projects</h2>**

**<ul>**

**<li>Expense Tracker</li>**

**<li>Portfolio Website</li>**

**</ul>**

**<h2>Contact</h2>**

**<p>Email: youremail@example.com</p>**

**</body>**

**</html>**

**5. Develop the application for Data Center to allocate resources using First Come First Serve (FCFS) policy and simulate the scenario using CloudSim.**

**SOURCE CODE:-**

**import org.cloudbus.cloudsim.all.\*;**

**public class Fcfs {**

**public static void main(String[] a) {**

**CloudSim.init(1, null, false);**

**Datacenter dc = Helper.createDatacenter("DC");**

**DatacenterBroker b = Helper.createBroker();**

**b.setId(1);**

**Vm v1 = new Vm(1, b.getId(), 1000, 1, 512, 1000, 10000, "X", new CloudletSchedulerTimeShared());**

**Vm v2 = new Vm(2, b.getId(), 1000, 1, 512, 1000, 10000, "X", new CloudletSchedulerTimeShared());**

**b.submitVmList(java.util.List.of(v1, v2));**

**Cloudlet c1 = new Cloudlet(1, 4000, 1, 300, true,**

**new UtilizationModelFull(), new UtilizationModelFull(), new UtilizationModelFull());**

**Cloudlet c2 = new Cloudlet(2, 2000, 1, 300, true,**

**new UtilizationModelFull(), new UtilizationModelFull(), new UtilizationModelFull());**

**c1.setUserId(b.getId());**

**c2.setUserId(b.getId());**

**b.submitCloudletList(java.util.List.of(c1, c2));**

**CloudSim.startSimulation();**

**CloudSim.stopSimulation();**

**for (Cloudlet c : b.getCloudletReceivedList())**

**System.out.println("Cloudlet " + c.getCloudletId() + " finished");**

**}**

**}**

**6. Develop and deploy a Chatbot application using Microsoft Windows Azure.**

**SOURCE CODE:-**

**import React, { useState } from "react";**

**function SimpleChat() {**

**const [messages, setMessages] = useState([{ user: "Bot", text: "Hi! Type something..." }]);**

**const [input, setInput] = useState("");**

**const sendMessage = () => {**

**if (!input.trim()) return;**

**const newMsg = { user: "You", text: input };**

**setMessages([...messages, newMsg]);**

**setInput("");**

**setTimeout(() => {**

**const reply = { user: "Bot", text: `You said: "${input}"` };**

**setMessages((msgs) => [...msgs, reply]);**

**}, 800);**

**};**

**return (**

**<div style={{ width: 300, margin: "40px auto", border: "1px solid #ccc", borderRadius: 8, padding: 10 }}>**

**<h3 style={{ textAlign: "center" }}>Simple Chat</h3>**

**<div style={{ height: 200, overflowY: "auto", marginBottom: 10, background: "#f9f9f9", padding: 5 }}>**

**{messages.map((m, i) => (**

**<div key={i} style={{ textAlign: m.user === "You" ? "right" : "left", margin: "5px 0" }}>**

**<b>{m.user}:</b> {m.text}**

**</div>**

**))}**

**</div>**

**<input**

**value={input}**

**onChange={(e) => setInput(e.target.value)}**

**onKeyDown={(e) => e.key === "Enter" && sendMessage()}**

**placeholder="Type a message..."**

**style={{ width: "75%", padding: 5 }}**

**/>**

**<button onClick={sendMessage} style={{ width: "25%", padding: 5 }}>**

**Send**

**</button>**

**</div>**

**);**

**}**

**export default SimpleChat;**

**7. Develop the application to Count Number of Words in the given text file using Hadoop.**

**7.1.SOURCE CODE (JAVA):-**

**import org.apache.hadoop.conf.\*;**

**import org.apache.hadoop.fs.Path;**

**import org.apache.hadoop.io.\*;**

**import org.apache.hadoop.mapreduce.\*;**

**import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;**

**import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;**

**public class WordCount {**

**public static class Map extends Mapper<Object, Text, Text, IntWritable> {**

**static final IntWritable one = new IntWritable(1);**

**Text word = new Text();**

**public void map(Object key, Text value, Context context)**

**throws java.io.IOException, InterruptedException {**

**for (String s : value.toString().split("\\s+")) {**

**if (s.length() > 0) {**

**word.set(s);**

**context.write(word, one);**

**}**

**}**

**}**

**}**

**public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {**

**public void reduce(Text key, Iterable<IntWritable> values, Context context)**

**throws java.io.IOException, InterruptedException {**

**int sum = 0;**

**for (IntWritable val : values)**

**sum += val.get();**

**context.write(key, new IntWritable(sum));**

**}**

**}**

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

**Job job = Job.getInstance(new Configuration(), "wordcount");**

**job.setJarByClass(WordCount.class);**

**job.setMapperClass(Map.class);**

**job.setCombinerClass(Reduce.class);**

**job.setReducerClass(Reduce.class);**

**job.setOutputKeyClass(Text.class);**

**job.setOutputValueClass(IntWritable.class);**

**FileInputFormat.addInputPath(job, new Path(args[0]));**

**FileOutputFormat.setOutputPath(job, new Path(args[1]));**

**System.exit(job.waitForCompletion(true) ? 0 : 1);**

**}**

**}**

**7.2.SOURCE CODE (PYTHON):-**

**(i)** [**reducer.py**](http://reducer.py)

**#!/usr/bin/env python3**

**import sys**

**current\_word = None**

**current\_count = 0**

**for line in sys.stdin:**

**word, count = line.strip().split("\t", 1)**

**count = int(count)**

**if current\_word == word:**

**current\_count += count**

**else:**

**if current\_word:**

**print(f"{current\_word}\t{current\_count}")**

**current\_word = word**

**current\_count = count**

**if current\_word:**

**print(f"{current\_word}\t{current\_count}")**

**(ii) mapper**[**.py**](http://reducer.py)

**#!/usr/bin/env python3**

**import sys**

**for line in sys.stdin:**

**line = line.strip()**

**words = line.split()**

**for word in words:**

**print(f"{word}\t1")**

**Commands to run in VS Code:**

**echo Hello Hadoop Hadoop > sample.txt**

**echo Hello World >> sample.txt**

**type sample.txt | python mapper.py | sort | python** [**reducer.py**](http://reducer.py)

**OUTPUT :-**

**Hello 2**

**Hadoop 2**

**World 1**