

1. Install Oracle Virtual box and create a Virtual Machine on your laptop/Desktop.

ANS: Install Oracle VirtualBox and Create a Virtual Machine:

Download and install Oracle VirtualBox from the official website.

Open VirtualBox and click on "New" to create a new virtual machine.

Follow the wizard to set up the virtual machine, specifying the desired OS, memory, storage, and network settings.

Start the virtual machine and proceed with the OS installation.

2. Install Oracle Virtual box and create a Virtual Machine and test the communication between the guest OS and Host OS using "PING" command.

ANS: Test Communication between Guest OS and Host OS:

Start the virtual machine and the host operating system.

In the guest OS, open a terminal or command prompt.

Use the "ping" command followed by the IP address or hostname of the host OS to test communication. For example: 'ping <host_IP_or_hostname>'.

If the host OS responds to the ping requests, the communication is successful

3. Create a Virtual Machine using Oracle Virtual Box and install "gcc" compiler.

ANS: Create a Virtual Machine and Install "gcc" Compiler:

Create a virtual machine following the steps mentioned in procedure 1.

Install the desired operating system on the virtual machine.

Open a terminal or command prompt in the guest OS.

Install the "gcc" compiler using the package manager specific to the guest OS. For example, on Ubuntu, you can use the following command: 'sudo apt-get install gcc.'

4. Write a procedure to compile C-program by splitting the programs into different modules and creating an application using "make" command

ANS: Compile C-Program with Modules using "make" Command:

Organize your C program code into separate modules (.c and .h files).

Create a "Makefile" in the project directory with instructions on how to compile and link the modules.

Open a terminal or command prompt in the directory containing the Makefile.

Run the "make" command to compile the program according to the instructions in the Makefile.

For example: 'make'.

The resulting application will be created based on the rules defined in the Makefile.

5. Install Oracle Virtual box and create two Virtual Machines

ANS: Install Oracle VirtualBox and Create Two Virtual Machines:

Follow the steps mentioned in procedure 1 to install Oracle VirtualBox.

Create two virtual machines by repeating the steps mentioned in procedure 1 for each virtual machine.

6. Write a procedure to transfer the files from one virtual machine to another virtual machine.

ANS: In the source virtual machine, create the files or locate the files you want to transfer.

Set up network connectivity between the two virtual machines (e.g., NAT, Bridged Adapter, or Host-Only Adapter).

Use a file transfer method such as SCP (Secure Copy) or SFTP (Secure FTP) to transfer the files between the virtual machines. You will need to provide the appropriate IP addresses, usernames, and passwords.

7. Create a central repository in GitHub and a local repository in your system and make the connection between the local repository and the central repository.

ANS: Create a Central Repository in GitHub and Local Repository:

Sign up for an account on GitHub if you don't have one already.

Create a new repository on GitHub by following the instructions provided.

On your local system, install Git if it's not already installed.

Open a terminal or command prompt and navigate to the directory where you want to initialize your local repository.

Run the command 'git init' to initialize a new local repository.

Connect your local repository to the central repository on GitHub using the command 'git remote add origin <repository_URL>', replacing '<repository_URL>' with the URL of your central repository

8. Write the procedure of pulling all the files from a remote repository.

ANS: Pull Files from a Remote Repository:

Open a terminal or command prompt in the local repository directory.

Use the command 'git pull origin <branch_name>' to pull files from the remote repository, replacing '<branch_name>' with the desired branch name.

9. Write the procedure to make changes in local repository and remote repository.

ANS: Make Changes in Local and Remote Repositories:

Make changes to the files in your local repository using a text editor or any other suitable tool. After making changes, stage the modified files using the command 'git add <file_name>' or 'git add' . to add all changed files.

Commit the changes using the command 'git commit -m "Commit message"' to save the changes locally.

Push the changes to the remote repository using the command 'git push origin <branch_name>'

10. Write the procedure to clone from local repository to remote repository.

ANS:Clone from Local Repository to Remote Repository:

On GitHub, create a new repository if you haven't done so already.

On your local system, open a terminal or command prompt and navigate to the directory where you want to clone the remote repository.

Run the command 'git clone <repository_URL>', replacing '<repository_URL>' with the URL of your local repository

11. Launch a GUI application inside the Docker Container

ANS:Launch GUI Application Inside Docker Container:

Install Docker on your system if it's not already installed.

Create a Dockerfile with instructions to build your Docker image, including the necessary packages and configurations for the GUI application.

Build the Docker image using the command 'docker build -t <image_name> .', replacing '<image_name>' with a suitable name for your image.

Run a Docker container based on the image using the command 'docker run -it <image_name>'.

12. Write a procedure to access the application from the Docker Host system.

ANS:Access Application from Docker Host System:

Determine the IP address of the Docker container using the command 'docker inspect <container_id>'.

On the Docker host system, open a web browser or suitable application and enter the IP address and port number (if applicable) to access the application running inside the Docker container.

13. Using a Simple code, implement a Ecommerce webpage using appropriate web tools

ANS:Implement an Ecommerce Webpage:

Choose appropriate web tools or frameworks such as HTML, CSS, JavaScript, and a backend language (e.g., Python, Ruby, or PHP).

Design and create the necessary webpages for the ecommerce site, including product listings, shopping cart functionality, and checkout process.

Implement server-side logic to handle user actions, process payments, and manage inventory. Integrate with suitable databases (e.g., MySQL, PostgreSQL) to store product information and user data.

Test the webpage by running a local web server or hosting it on a suitable platform.

14. Create a simple Personal blog which should detail your academic proficiency, skill sets, personal interests using appropriate Web tools (XAMPP Server can be used if required)

ANS:Create a Personal Blog:

Choose appropriate web tools or frameworks such as HTML, CSS, and JavaScript.

Design and create webpages for your personal blog, including sections for academic proficiency, skill sets, personal interests, and any other relevant information.

Implement server-side logic if needed (e.g., using a backend language like PHP or Python).

Host your blog using a web server or a suitable platform like XAMPP to make it accessible on the internet.

15. Write a procedure to launch an EC2 instance and connect to it using the AWS Management Console

ANS:Launch an EC2 Instance and Connect via AWS Management Console:

Sign in to the AWS Management Console.

Navigate to the EC2 service and launch a new instance.

Follow the steps in the instance launch wizard, selecting the desired AMI, instance type, and other configuration options.

Once the instance is running, note down the public IP address or DNS name.

Use SSH or the AWS Management Console's EC2 Instance Connect feature to connect to the instance

16. Write a procedure to develop an application using Google App Engine to display the text as you wish.

ANS:Develop an Application using Google App Engine:

Sign in to the Google Cloud Console and create a new project.

Enable the Google App Engine service for your project.

Write the application code using suitable programming languages supported by Google App Engine (e.g., Python, Java, Go).

Configure the app.yaml file to define the runtime environment and app-specific settings.
Deploy the application to Google App Engine using the appropriate deployment command or by connecting your code repository to automatically deploy changes.

17. Write a procedure to develop an application using Windows Azure to display the text "Hello World".

ANS: Develop an Application using Windows Azure:

Sign in to the Azure Portal and create a new resource group and an appropriate Azure service. Write the application code using a supported programming language and framework (e.g., C#, .NET, Node.js).

Configure the Azure service based on the application requirements, such as deploying a web app, creating a virtual machine, or setting up a container instance.

Deploy the application to the Azure service using the provided deployment options (e.g., Git, Azure CLI, Azure DevOps).

18. Write the procedure to configure a XAMPP Server for enabling web application and development.

ANS: Configure XAMPP Server for Web Application Development:

Download and install XAMPP from the official website.

Start the XAMPP control panel and ensure the Apache and MySQL services are running.

Place your web application files in the appropriate directory (e.g., htdocs folder in XAMPP installation directory).

Configure the Apache server by modifying the httpd.conf file to specify necessary settings, such as document root and virtual hosts.

Create and configure the database for your web application using the phpMyAdmin interface or other suitable tools.

Access your web application by opening a web browser and entering the appropriate URL based on your Apache configuration.