**Simple problem statements (Beginners-B)**

1. **Demonstrate to use of Version Control System (Git offline: on local machine with multiple user).**

Create multiple users and usage with team leader role and coder roles.

Demonstrate code merge, fork (branching) and code diff amongst the users

Compare it with svn. (on answer sheet)

1. **Demonstrate to use of Version Control System (Git offline and connect to github account).**

Create multiple users and usage with team leader role and coder roles.

Demonstrate code pull and push, fork (branching) amongst the users

Compare it with hg. (on answer sheet)

1. **Demonstrate to use of Version Control System (Git offline and connect to github account).**

Create multiple users and usage with team leader role and coder roles.

Demonstrate code pull and push, fork (branching) amongst the users, code merge, code diff amongst the users.

Compare git with gitea. (on answer sheet)

1. **Demonstrate to use of Version Control System (hg mercurial offline: on local machine with multiple user).**

Create multiple users and usage with team leader role and coder roles.

Demonstrate code merge, fork (branching) and code diff amongst the users

Compare it with svn. (on answer sheet)

1. **Demonstrate to use of Version Control System (Hg mercurial offline and connect to github account).**

Create multiple users and usage with team leader role and coder roles.

Demonstrate code pull and push, fork (branching) amongst the users

Compare it with hg. (on answer sheet)

1. **Demonstrate to use of Version Control System (SVN offline and connect to github account).**

Create multiple users and usage with team leader role and coder roles.

Demonstrate code pull and push, fork (branching) amongst the users, code merge, code diff amongst the users.

Compare git with SVN. (on answer sheet)

1. **Demonstrate the use/features of online Bug Tracking/Issue Tracking "mantis".**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet ) in terms removing error/bug or adding feature to it.

[Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare all FOSS bug tracking tools.

1. **Demonstrate the use/features of online Bug Tracking/Issue Tracking "BugZilla".**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet ) in terms removing error/bug or adding feature to it.

[Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare it with other online **Bug trackingtool. How bug tracking improves the quality of code** (on answer sheet).

1. **Development of contribution to existing Open Source Software (on line upload of its git/svn repository with your valid login)**

(Language: java/pyth/perl/c/cpp/etc ).

Pull to your github the **Open Source Software and** contribute in terms removing error/bug or adding feature to it.

Upload the contributed code your repository as well original FOSS.

1. **Demonstrate the use/features of CMS software: Joomla.**

Crate users and show how Joomla manages web sites for a client. Also implement the working of core features of Joomla.

Compare it with other **CMS** like schoology/Moodle (on answer sheet)

1. **Create of RPM packages.**

(Multiple modules/code packaging of java).

Pull or take any FOSS project and one feature and then create the package on suitable OS

Compare RPM packaging with Debian packaging (on answer sheet)

1. **Create of RPM packages.**

(multiple modules/code packaging of c/cpp).

Pull or take any FOSS project and one feature or remove bug and then create the package on suitable OS

Compare RPM packaging with Debian packaging (on answer sheet)

1. **Create of Debian packages.**

(Multiple modules/code packaging of python).

Pull or take any FOSS project and one feature and then create the package on suitable OS

Compare RPM packaging with Debian packaging (on answer sheet)

1. **Create of Debian packages.**

(multiple modules/code packaging of java/c/cpp).

Pull or take any FOSS project and one feature and then create the package on suitable OS

Compare RPM packaging with Debian packaging (on answer sheet)

1. **Demonstrate the use/features of Project Management tool: “SONARQUBE” for managing projects.**

[Project planning and scheduling](https://www.openproject.org/collaboration-software-features/#project-planning)/ [Product roadmap and release planning](https://www.openproject.org/collaboration-software-features/#product-management)/ [Task management and team collaboration](https://www.openproject.org/collaboration-software-features/#task-management)/ [Agile and Scrum](https://www.openproject.org/collaboration-software-features/#agile-scrum)/[Time tracking, cost reporting and budgeting](https://www.openproject.org/collaboration-software-features/#time-tracking)/ [Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare it with other **Project Management tool** (on answer sheet)

1. **Demonstrate the use/features of online Project Management tool: “SONARcloud” for managing projects.**

Demonstrate the one foss project for code coverage, Detect Bugs & Vulnerabilities, Review Security Hotspots, Track Code Smells & fix your Technical Debt. Show the Code Quality Metrics & History.

Compare it with other **Project Management tool** (on answer sheet)

1. **Demonstrate the use/features of online Project Management tool: “ASANA” for managing projects.**

Demonstrate the one foss project for [Project planning and scheduling](https://www.openproject.org/collaboration-software-features/#project-planning)/ [Product roadmap and release planning](https://www.openproject.org/collaboration-software-features/#product-management)/ [Task management and team collaboration](https://www.openproject.org/collaboration-software-features/#task-management)/ [Agile and Scrum](https://www.openproject.org/collaboration-software-features/#agile-scrum)/[Time tracking, cost reporting and budgeting](https://www.openproject.org/collaboration-software-features/#time-tracking)/ [Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare asana with phabricator (on answer sheet)

1. **Demonstrate the use/features of online Project Management tool & communication tools “slack with ASANA” for managing projects.**

Demonstrate the one foss project for [Project planning and scheduling](https://www.openproject.org/collaboration-software-features/#project-planning)/ [Product roadmap and release planning](https://www.openproject.org/collaboration-software-features/#product-management)/ [Task management and team collaboration](https://www.openproject.org/collaboration-software-features/#task-management)/ [Agile and Scrum](https://www.openproject.org/collaboration-software-features/#agile-scrum)/[Time tracking, cost reporting and budgeting](https://www.openproject.org/collaboration-software-features/#time-tracking)/ [Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare asana with phabricator (on answer sheet)

1. **Demonstrate the use/features of online Project Management tool & communication tools “asana with github” for managing projects.**

Demonstrate the one foss project for [Project planning and scheduling](https://www.openproject.org/collaboration-software-features/#project-planning)/ [Product roadmap and release planning](https://www.openproject.org/collaboration-software-features/#product-management)/ [Task management and team collaboration](https://www.openproject.org/collaboration-software-features/#task-management)/ [Agile and Scrum](https://www.openproject.org/collaboration-software-features/#agile-scrum)/[Time tracking, cost reporting and budgeting](https://www.openproject.org/collaboration-software-features/#time-tracking)/ [Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare asana with phabricator (on answer sheet)

1. **Demonstrate the use/features of Bug Tracking/management: "YouTrack".**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet ) in terms removing error/bug or adding feature to it.

[Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare code review with **Bug tracking process.. How bug tracking or code review improves the quality of code** (on answer sheet).

1. **Demonstrate the use/features of online Bug Tracking/management: "donedone".**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet ) in terms removing error/bug or adding feature to it.

[Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare code review with **Bug tracking process.. How bug tracking or code review improves the quality of code** (on answer sheet).

1. **Demonstrate the use/features of CMS software: "Drupal".**

Crate users and show how Drupal manages contents of web sites for a client. Also implement the working of core features of Drupal.

Compare it with other **CMS** like schoology/(on answer sheet)

1. **Configure and demonstrate the use of FTP and Telnet. (on ubantu)**

Show the imp steps and file name of configurations**.** (on answer sheet)

Put the Pub folder available for access to all.

1. **Configure and demonstrate the use of NIS and NFS. (on ubantu)**

Show the imp steps and file name of configurations**.** (on answer sheet)

Create 5 users and make two groups, demonstrate the NIS and NFS concepts by example on LAN connected linux os.

1. **Configure and demonstrate the use of FTP and Telnet. (on centos/rpm based OS VM/container )**

Show the imp steps and file name of configurations**.** (on answer sheet)

Put the Pub folder available for access to all.

1. **Configure and demonstrate the use of NIS and NFS. (on centos/rpm based OS VM/container )**

Show the imp steps and file name of configurations**.** (on answer sheet)

Create 5 users and make two groups, demonstrate the NIS and NFS concepts by example on LAN connected linux os.

1. **Demonstrate the use/features of Project Management tool: “Open Atrium” for managing.**

[Project planning and scheduling](https://www.openproject.org/collaboration-software-features/#project-planning)/ [Product roadmap and release planning](https://www.openproject.org/collaboration-software-features/#product-management)/ [Task management and team collaboration](https://www.openproject.org/collaboration-software-features/#task-management)/ [Agile and Scrum](https://www.openproject.org/collaboration-software-features/#agile-scrum)/[Time tracking, cost reporting and budgeting](https://www.openproject.org/collaboration-software-features/#time-tracking)/ [Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare it with other **Project Management tool** (on answer sheet)

1. **Demonstrate the use/features of online Bug Tracking/management: Jira.**

Take source code of any open source you like and find minimum three bugs (mention their type on answer sheet ) in terms removing error/bug or adding feature to it.

[Bug tracking](https://www.openproject.org/collaboration-software-features/#bug-tracking) on any suitable open source (code) from internet.

Compare it with other **Bug trackingtool. How bug tracking improves the quality of code** (on answer sheet)

1. **Demonstrate the use for code review :review board tools with tuleap (trial) or gitea**
2. Write a go program to demonstrate feature of go language (any one)
3. Multithreading and concurrency
4. Synchronization/ Concurrency primitives
5. Access the information hardware cores and its utilization
6. Go for web technology for server / client side coding
7. Automatic garbage collection and Slightly safer pointers (no pointer arithmetic)
8. Slightly cleaner syntax with some enforced standards
9. Static compilation by default (no dependencies)
10. A simple object syntax with no inheritance

# 

1. **Demonstrate go Tools** (any one)**:** Go distribution includes tools for building, testing, and analyzing code:

* go build, which builds Go binaries using only information in the source files themselves, no separate makefiles
* go test, for unit testing and microbenchmarks
* go fmt, for formatting code
* go get, for retrieving and installing remote packages
* go vet, a static analyzer looking for potential errors in code
* go run, a shortcut for building and executing code
* godoc, for displaying documentation or serving it via HTTP
* gorename, for renaming variables, functions, and so on in a type-safe way
* go generate, a standard way to invoke code generators

1. **Demonstrate go** Applications: (any one)

Some notable open source applications written in Go include:

* [Caddy](https://en.wikipedia.org/wiki/Caddy_(web_server)), an open source HTTP/2 web server with automatic HTTPS capability.
* [CockroachDB](https://en.wikipedia.org/wiki/CockroachDB), an open source, survivable, strongly consistent, scale-out SQL database.
* [Docker](https://en.wikipedia.org/wiki/Docker_(software)), a set of tools for deploying [Linux](https://en.wikipedia.org/wiki/Linux) containers
* [Ethereum](https://en.wikipedia.org/wiki/Ethereum), The go-ethereum implementation of the Ethereum Virtual Machine blockchain for the Ether cryptocurrency
* [Hugo](https://en.wikipedia.org/wiki/Hugo_(software)), a static site generator
* [InfluxDB](https://en.wikipedia.org/wiki/InfluxDB), an open source database specifically to handle time series data with high availability and high performance requirements.
* [InterPlanetary File System](https://en.wikipedia.org/wiki/InterPlanetary_File_System), a content-addressable, peer-to-peer hypermedia protocol.
* [Juju](https://en.wikipedia.org/wiki/Juju_(software)), a service orchestration tool by [Canonical](https://en.wikipedia.org/wiki/Canonical_Ltd), packagers of [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) Linux
* [Kubernetes](https://en.wikipedia.org/wiki/Kubernetes) container management system
* [Lightning Network](https://en.wikipedia.org/wiki/Lightning_Network), a [bitcoin](https://en.wikipedia.org/wiki/Bitcoin) network that allows for fast Bitcoin transactions and scalability.
* [Mattermost](https://en.wikipedia.org/wiki/Mattermost), a teamchat system
* [OpenShift](https://en.wikipedia.org/wiki/OpenShift), a cloud computing platform as a service by [Red Hat](https://en.wikipedia.org/wiki/Red_Hat)
* [Snappy](https://en.wikipedia.org/wiki/Snappy_(package_manager)), a package manager for [Ubuntu Touch](https://en.wikipedia.org/wiki/Ubuntu_Touch) developed by Canonical.
* [Syncthing](https://en.wikipedia.org/wiki/Syncthing), an open-source file synchronization client/server application
* [Terraform](https://en.wikipedia.org/wiki/Terraform_(software)), an open-source, multiple [cloud](https://en.wikipedia.org/wiki/Cloud_computing) infrastructure provisioning tool from [HashiCorp](https://en.wikipedia.org/wiki/HashiCorp).

1. Download the suitable version of kernel/linux code from distro/linux.org and add print statement and compile it and demonstrate the change kernel/print statement added.

**Tough problem statements (Expert-E)**

1. **Demonstrate use of bug tracking tool/any foss tool and create the docker image of that tool. Push that image.**

**Run the docker container from recently created image and run that docker container.**

**Push that image.**

1. **Write a Docker File to pull the Ubuntu with open jdk and write any java application.**
2. **Demonstrate use of bug tracking tool and create the docker image of that tool. Push that image.**

**Run the docker container from recently created image and run that docker container.**

**Push that image.**

1. **Create two applications/socket/IPC in two different docker containers. Push those applications and run to show the communications between two dockers.**

**Ex. Shared memory between two container**

1. **Create two applications/socket/IPC in two different docker containers. Push those applications and run to show the communications between two dockers.**

**Ex Message passing between two container**

1. **Create two applications/socket/IPC in two different Docker containers. Push those applications and run to show the communications between two dockers.**

**Ex. TCP/UDP Socket communication between two container**

1. **Pull the LAMP Stack container from docker hub and host a web application of your own. Push that image back to repository. Make use of database.**
2. **Create a web application with simple web page containing login details and create a docker image of the application.(Use Apache Web server)**

**Run the Docker container from recently created image and run the container at port number 80 in host system. Push that image to repository. Make use of database. Try to access it from other instance of docker.**

1. **With the help of Docker-compose deploy the ‘Wordpress’ and ‘Mysql’ container and access the front end of ‘Wordpress’**
2. **A. Create a simple Hello-world python flask application and create the docker image of that Flask application.**

**B. Run the docker container from recently created image and run that docker container to 5000 port of host system.**

1. **Create the ‘nginx’ container from ‘nginx’ image. And create the load balancing so that if we go to tha address of ‘nginx ‘ it can redirect it to the above created applications (Flask and Wordpress).**
2. **Create a web application with simple web page containing login details and create a docker image of the application.(Use Apache Web server)**
3. **Run the Docker container from recently created image and run the container at port number 80 in host system.**
4. **Write a python program to perform arithmetic operations and create Docker image accordingly.**

**Run the Docker container with created image .**

1. **Create a simple web application using LAMP Stack on docker container.**
2. **Create a web application with simple web page containing login details and create a docker image of the application.(Use Ngnix Web server)**
3. **Run the Docker container from recently created image and run the container at port number 80 in host system.**
4. **Create a simple Hello-world python flask application and create the docker image of that Flask application.**
5. **Run the docker container from recently created image and run that docker container to 5000 port of host system.**
6. **Pull the LAMP Stack container from docker hub and host a web application of your own.**
7. **Create a Docker image of simple web application from using HTTP web server at port 5000 in host.**
8. **Create a docker image of simple login form using Flask on port 7000.**
9. **Create a docker image of simple login form using django on port 6000.**
10. **Create a container with ngnix web server and create one more container with mysql.**
11. **Create a simple web form to insert the records in mysql data base.**
12. **Mount any directory of host system to the container.**
13. **Write a Docker File to pull the Ubuntu with open jdk and write any java application.**
14. **Run a LAMP Stack Container at port 8080 and host media wiki site on native machine.**
15. **Write a C program to create singly linked list and containerize it.**
16. **Create a LAMP Stack container and host a web application of your own.**
17. **Create & Demonstrate the container of a particular distro and show all dockerfile fields**
18. **Create & Demonstrate the containers of a particular distro and show all Docker compose file fields.**

**With example**

1. **Configure and demonstrate the use of FTP and Telnet. (on Ubuntu container )**

Show the imp steps and file name of configurations**.** (on answer sheet)

Put the Pub folder available for access to all.

1. **Configure and demonstrate the use of NIS and NFS. (on Ubuntu container )**

Show the imp steps and file name of configurations**.** (on answer sheet)

Create 5 users and make two groups, demonstrate the NIS and NFS concepts by example on LAN connected linux os.

1. **Configure and demonstrate the use of FTP and Telnet. (on centOS/redhatcontainer )**

Show the imp steps and file name of configurations**.** (on answer sheet)

Put the Pub folder available for access to all.

1. **Configure and demonstrate the use of NIS and NFS. (on centOS/redhatcontainer)**

Show the imp steps and file name of configurations**.** (on answer sheet)

Create 5 users and make two groups, demonstrate the NIS and NFS concepts by example on LAN connected linux os.