Devops

PRGM1:-

1> Create a Maven Project
2> SRC/main/Java
    ↳ Create new Package -> Com.DSCE
        ↳ New class -> APP

3> SRC/Test/Java
    ↳ Create new Package -> Com.DSCE
        ↳ New class -> App test

↳ Go to Maven Repo & Paste 7.1.3 <D>
3> SRC/main/Resources
    ↳ New File -> Config. Properties

4> Maven Build, clean, Compile, Install
5> install Gradle, Gradle-version, Grade -init
6> Goto -> help Eclipse marketplace
    ↳ install Testing

PRGM: 3

-> Create Maven Project Com.DSCE -> PRGM3a
-> SRC
    ↳ main
        ↳ WebAPP
            ↳ RC -> Create ~~Folder~~ File [index.HTML]
                Create folder (WEB-INF)
        ↳ WEB-INF -> Create file (web.xml)
-> Inside web.xml [Plugin file]
-> Inside index.html (Basic HTML code)

-> Pom.xml [Maven Build -> clean, install]

-> Open VS Code
-> Open folder (Cut & Paste your Prgm 3a Path)
-> New file
    ↳ Docker File -> write Code
    ↳ COPY Prgm3a.war & Paste in COPY from /target/

    ↳ Open Terminal
        ↳ Docker Build -t app .
        ↳ Docker run -d -P 8085:8080 APP
        ↳ Goto Chrome -> localhost:8085

3 b)

-> Create New Folder
    ↳ multi - Container

-> VS Code
    ↳ Open the new folder Created
-> Create two app (folders)
    ↳ App1
    ↳ app2
-> Create -> File
    ↳ docker - Compose . yml
-> Create File in app1
    ↳ app.py
    ↳ Dockerfile
    ↳ Requirements . txt

-> Same in app2

-> app.py1 -> Type Code
    ↳ from flask
-> Dockerfile -> Type Code
-> Req Txt -> flask == 3.0.0

2 app.y => Type Code
    ↳ Dockerfile ->
    Req Txt -> requests==2.31.0

-> docker-Compose.yml -> type Code

-> Terminal -> docker-Compose build
    docker-Compose up

PRGM: 4

1) github.com/kubernetes/minikube/releases/latest.
-> download minikube-windows-amd64.exe
-> After download Rename as minikube

-> Create new folder as minikube.
-> Copy the downloaded minikube & Paste it new minikube folder
-> Run it
-> Copy Path & Edit System Environment in Path new

Path

-> in minikube cmd
-> minikube start --driver=docker

-> Create New folder -> PrGm 4 & Open in VS Code
-> Create app.py
-> deployment.yaml
-> service.yaml
-> Dockerfile
-> Requirement.txt

Terminal
-> minikube start
-> docker build -t dockerusername/imagename.
-> docker push dn/in
-> kubectl apply -f deploy.yaml
service.yaml
->
kubectl port-forward svc/hello-world 5000/5000

Prgm 2

-> Create Maven Project
    ↳ Com.dsce - Prgm2 war

-> SRC - Webapp
    ↳ File -> index.html
    ↳ WEB-INF (Folder)
        ↳ File (web.xml)

-> Index.html (own File)
-> Plugin -> in web.xml

-> Prgm 2 (Update Project)
    ↳ Properties Copy Path

-> cmd -> cd "Path"
-> git init
-> Prgrm 2
    ↳ Team -> shareproject - Select git

-> Go to Github
-> Create Repo (Prgram 2]
-> Copy Repo Path
-> PRgrm 2 -> Team -> Commit [version 1]
-> Select all files from unstaged & Paste in Staged (drag)
    ↳ Commit
-> Push head [Github username & Token Id]
-> For Token select Repo & workflow -> Generate Token
-> Preview -> Push. (Check in git all files done)

-> Tomcat download 9 for window
-> (32 bit / 64)

Give Tom Cat Credential (Note it)
↳ Change Role -> manager-gui to manager-scr
↳ Finish

->> download Apache-maven (3.9.10 bin zip)
Maven Install

-> Click the downloaded maven & Extract to Tomcat where [illegible]

-> & Path in System Variable

-> Go to Jenkins
Install Plugins
1) git P
2) deployment container
3) Maven integration

-> Tool -> Maven -> name & env Path Save

-> + New item
Name -> Prgm2
↳ Freestyle Project -> OK
-> Select git
-> Paste git URL
-> Add build step -> Invoke toplevel maven targets
↳ maven 3.9.10
↳ clean-install

-> Add Post build
  -> Deploy war/ear to a
    -> ** / *. war
Context Path -> /Prgm 2

  -> Add Tomcat Credentials
  -> Tomcat url localhost: 8089/manager/text
  -> Save
-> Build Now

-> Now check localhost: 8089/Prgm 2/