Program-09

→ https://dev.azure.com

→ sig.in

→ No acc: create account

→ search → azure devops organizations

→ my azure account (click)

→ click on continue

→ create new organization

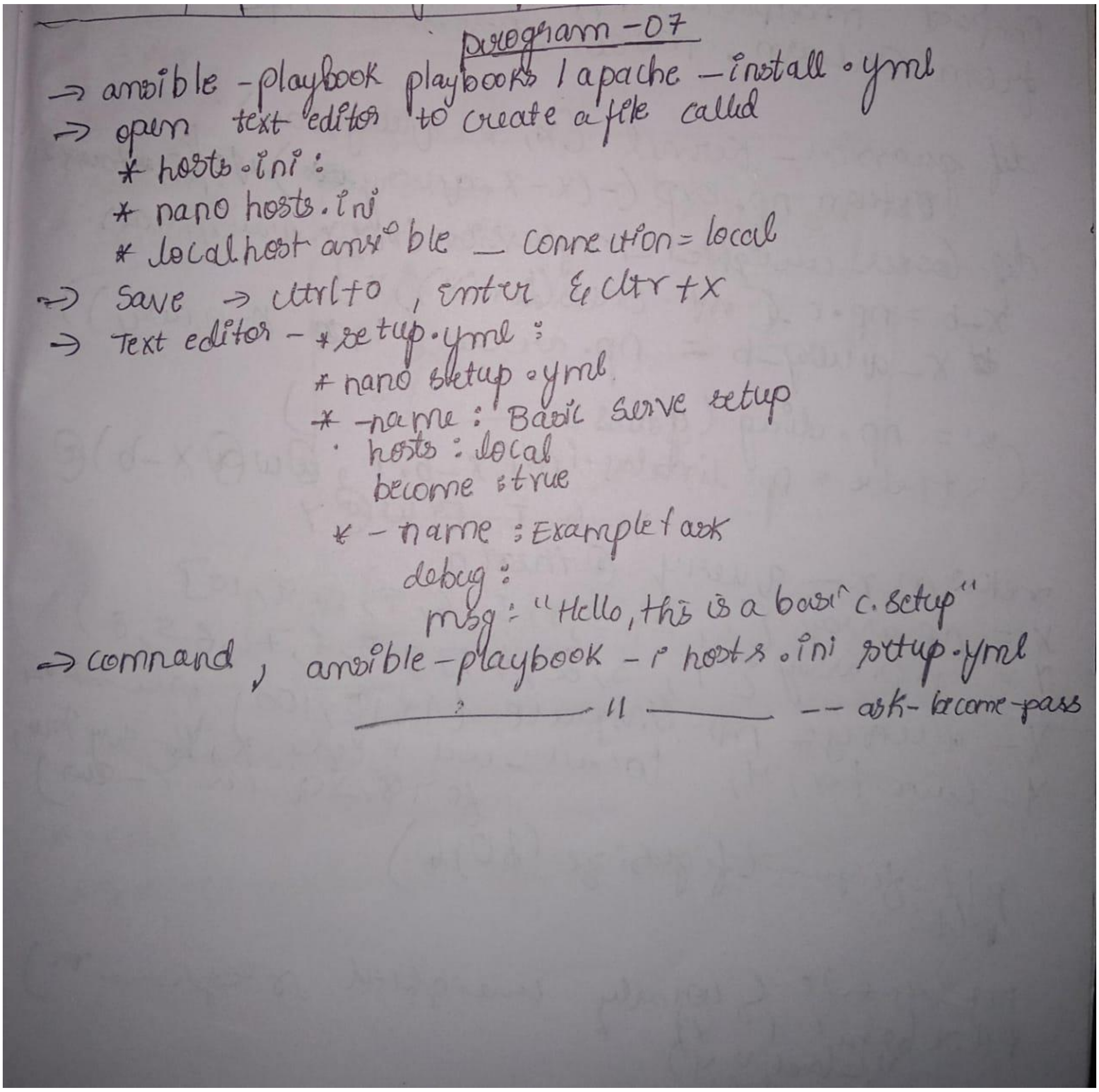
→ continue

→ enter capta & click on continue

→ project name → ik (private)

→ create project

1,4



program -07

→ ansible -playbook playbooks / apache -install . yml

→ open text editor to create a file called

- * hosts . ini :
- * nano hosts . ini
- * localhost ansi ble _ connection = local

→ save → ctrl+o , enter & ctr+x

→ Text editor - * setup . yml :

```
* nano setup . yml
* -name : Basic serve setup
  hosts : local
  become : true
 * - name : Example task
   debug :
     msg : "Hello, this is a basic. setup"
```

→ command , ansible-playbook -i hosts . ini setup . yml

 ------------ " ------------ -- ask- become-pass

program-05

-> Install Jenkins - See Java(JDK) is installed
- check Java -version

-> Goto Google - Jenkins download
- While installing choose login system as : Run Service as LocalSystem
- default port - click on test & next
- change directory & choose Java jdK-21 path look like c:\pgm \Files\Java\jdk-21
- click next, nxt & permission-Yes, starts installing
- http://localhost:8080

-> Notepad copy paste the password & click OK

-> Install suggested plugins , Enter the details then click on Save & continue , then Save & finish & click on Start using Jenkin

Ex

program-06

-> Installation of Jenkins in our system

[-> ~~Sign in to Jenkins -> user name, password , sign in~~
-> ~~Dashboard, manage & Engage, tools, install JDK-17, give path, default, install automaticaly, maven path & save~~
-> ~~plugins, install plugins - piple, git, maven integrated plugin~~]

command
-> pwd , cd desktop, mkdir ⊏]

program-06

-> command - * pwd, cd desktop, mkdir gitproject, cd gitproject
git init.

-> vi file 1.sh (to save & quit - press esc & :wq), ls,
git add * , git commit -m "commit message"

-> git config --global user.name "username"
git config --global user.Email "Emailid"
git Config -- global-list
git remote add alias name url
git remote -v

-> git push alias name master, dashboard

-> Github account -> go to settings, developer settings, personal access token, classic, generate

program-03

-> working with Gradle project (Groovy DSL):
Command - * gradle init --type java-application
* 17
* groovyProject
* application structure - 1
* Groovy - 2
* 1
* no

-> Start menu, build.gradle open, build, Right click Edit with, Notepad - OK, Copy paste the code & Save it

-> AdditionOperation.java, Src/main/java/org/example, change file name from App.Java to AdditionOperation.java, copy the code & paste it & Save

-> Same 2 steps to be followed but only difference is to change AdditionOperationTest.java & main to Test

● 1 - To build * gradle build
● 2 - run * gradle run
● 3 - test * gradle test

program - 04

-> command - * mvn archetype:generate -DgroupId=com.Example -DartifactId=maven-example -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

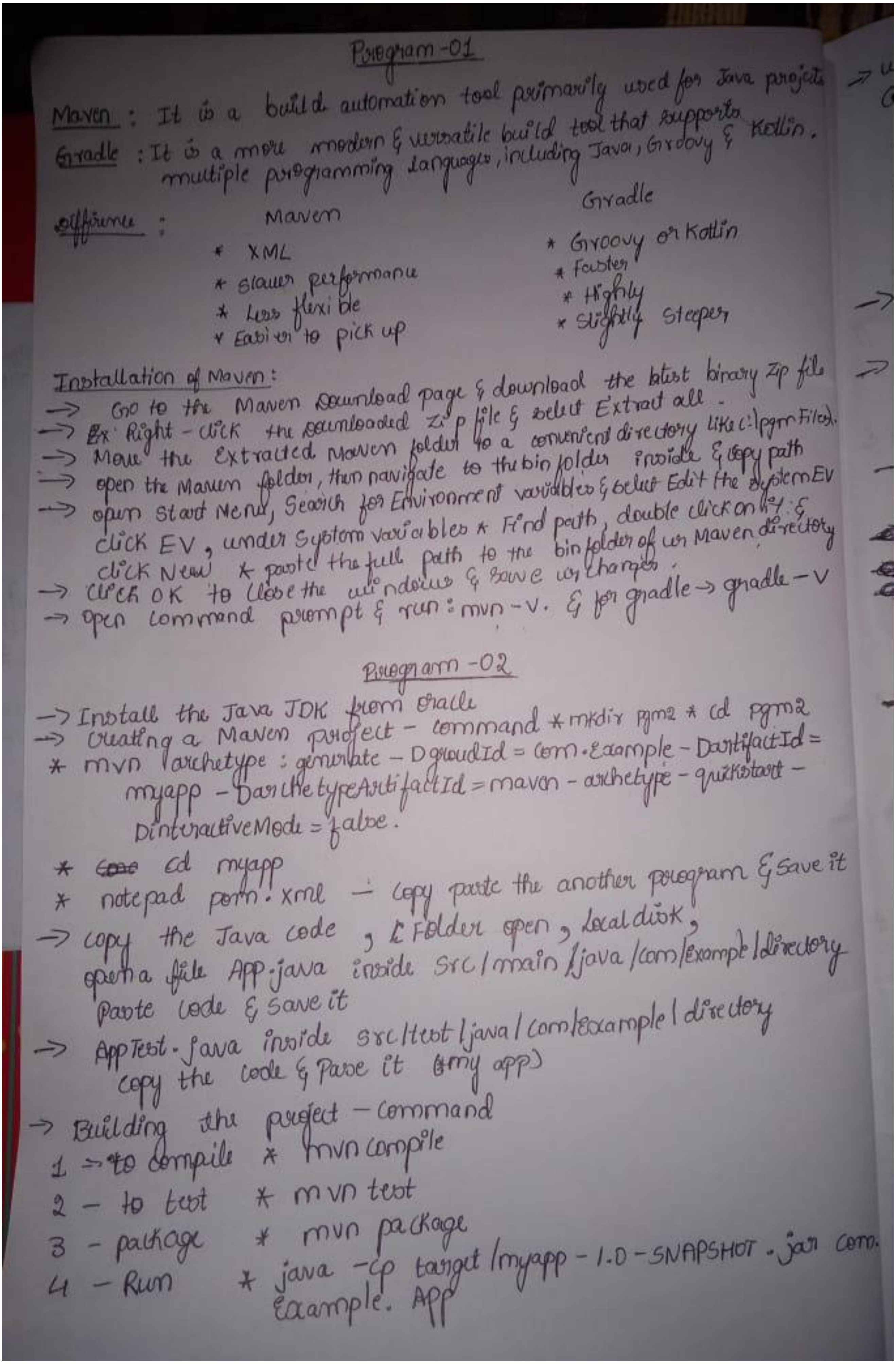
->*cd maven-example * notepad pom.xml - copy & paste & Save

-> App.java inside src/main/java/com/example/directory - C & P & S

-> command - * mvn clean install
* mvn exec:java -Dexec.mainClass="com.Example.App"

-> Command - * gradle init
* yes
* 2
* no
* build.gradle - C & P & S

-> ~~gra~~ command - * gradlew build
* gradlew run



Program-01

Maven : It is a build automation tool primarily used for Java projects

Gradle : It is a more modern & versatile build tool that supports multiple programming languages, including Java, Groovy & Kotlin.

difference :

Maven	Gradle
* XML	* Groovy or Kotlin
* slower performance	* Faster
* Less flexible	* Highly
* Easier to pick up	* Slightly Steeper

Installation of Maven:

-> Go to the Maven download page & download the latest binary zip file
-> Ex: Right-click the downloaded zip file & select Extract all.
-> Move the Extracted Maven folder to a convenient directory like C:/pgm Files.
-> open the Maven folder, then navigate to the bin folder inside & copy path
-> open Start Menu, Search for Environment variables & select Edit the system EV click EV, under System variables * Find path, double click on it & click New * paste the full path to the bin folder of ur Maven directory.
-> Click OK to close the windows & save ur changes.
-> open command prompt & run: mvn -v. & for gradle -> gradle -v

Program-02

-> Install the Java JDK from Oracle
-> Creating a Maven project - command * mkdir pgm2 * cd pgm2
* mvn archetype : generate - DgroupId = com.Example - DartifactId = myapp - DarchetypeArtifactId = maven - archetype - quickstart - DinteractiveMode = false.

* ~~cd~~ cd myapp
* notepad pom.xml - copy paste the another program & save it
-> copy the Java code, & Folder open, local disk, open a file App.java inside src/main/java/com/example/directory paste code & save it
-> AppTest.java inside src/test/java/com/example/directory copy the code & Pase it (my app)
-> Building the project - command
1 -> to compile * mvn compile
2 - to test * mvn test
3 - package * mvn package
4 - Run * java -cp target/myapp-1.0-SNAPSHOT.jar com.Example. App