

## **PRACTICAL – 01**

**AIM : Installation of java on Unix/Linux Machine (without using VirtualBox).**

### **Description:**

In this practical, we install Java directly on a Unix/Linux machine. Since no virtualization software like VirtualBox is used, the installation is performed natively on the operating system. The steps involve downloading or using the package manager to get the required Java version, setting up environment variables, and verifying the installation.

### **PROCEDURE :**

**Step 1:** Enable WSL(Windows Subsystem for Linux)

- Open PowerShell as Administrator
- Run the following command:

**wsl –install**

- Restart your computer when prompted. After restart, choose Ubuntu or install it from the Microsoft Store.

**Step 2:** Open Ubuntu (WSL)

- Search for 'Ubuntu' in Start Menu and open it.
- It will initialize and ask for a username and password.

**Step 3:** Update the Package List:

**sudo apt update**

**Step 4:** Install Java (OpenJDK 8 or 11)

- For Java 8:

**sudo apt install openjdk-8-jdk -y**

- For Java 11:

**sudo apt install openjdk-11-jdk -y**

**Step 5:** Verify Java Installation :

## java -version

### Step 6: Set JAVA\_HOME Environment Variable (Optional)

- Open .bashrc file:

```
nano ~/.bashrc
```

- Add the following lines at the end:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

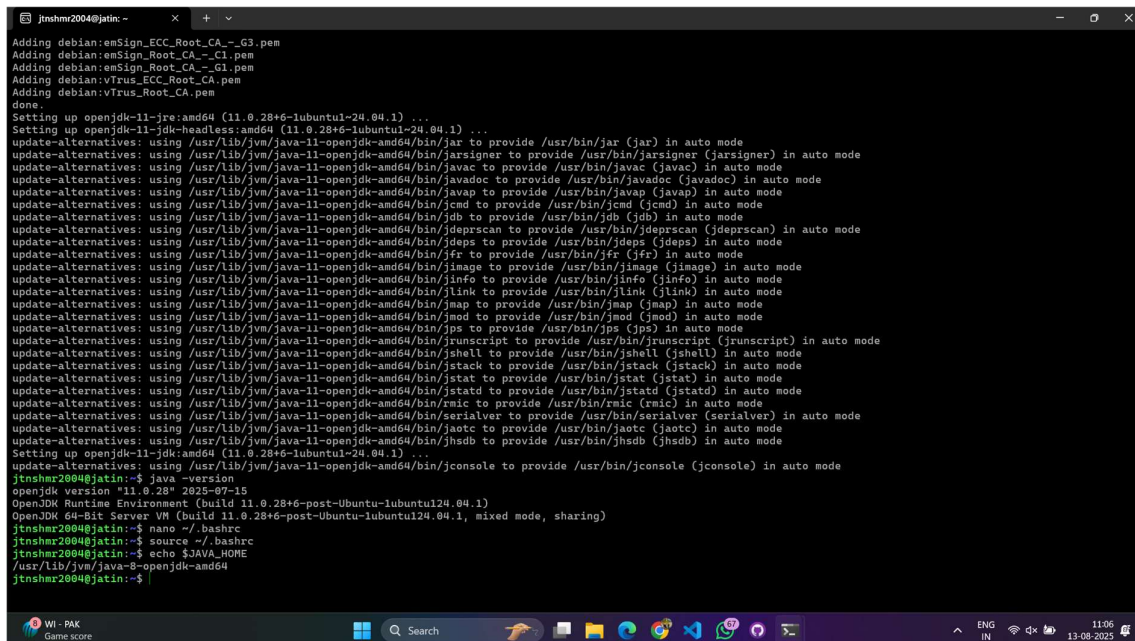
```
export PATH=$JAVA_HOME/bin:$PATH
```

- Save and exit (Ctrl+X, then Y, then Enter)
- Reload bashrc:

```
source ~/.bashrc
```

- Verify JAVA\_HOME is set:

```
echo $JAVA_HOME
```



```
jtnshmr2004@jatin: ~
Adding debian:emSign_ECC_Root_CA_-_G3.pem
Adding debian:emSign_Root_CA_-_C1.pem
Adding debian:emSign_Root_CA_-_G1.pem
Adding debian:vTrus_ECC_Root_CA.pem
Adding debian:vTrus_Root_CA.pem
done.
Setting up openjdk-11-jre-headless:amd64 (11.0.28+6-lubuntul-24.04.1) ...
Setting up openjdk-11-jdk-headless:amd64 (11.0.28+6-lubuntul-24.04.1) ...
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jar to provide /usr/bin/jar (jar) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jarsigner to provide /usr/bin/jarsigner (jarsigner) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javac to provide /usr/bin/javac (javac) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javadoc to provide /usr/bin/javadoc (javadoc) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javap to provide /usr/bin/javap (javap) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jcmd to provide /usr/bin/jcmd (jcmd) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jdb to provide /usr/bin/jdb (jdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jdeprscan to provide /usr/bin/jdeprscan (jdeprscan) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jdeps to provide /usr/bin/jdeps (jdeps) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jfr to provide /usr/bin/jfr (jfr) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jimage to provide /usr/bin/jimage (jimage) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jinfo to provide /usr/bin/jinfo (jinfo) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jlink to provide /usr/bin/jlink (jlink) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jmap to provide /usr/bin/jmap (jmap) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jmod to provide /usr/bin/jmod (jmod) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jps to provide /usr/bin/jps (jps) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jrunscript to provide /usr/bin/jrunscript (jrunscript) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jshell to provide /usr/bin/jshell (jshell) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstack to provide /usr/bin/jstack (jstack) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstat to provide /usr/bin/jstat (jstat) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstatd to provide /usr/bin/jstatd (jstatd) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/rmic to provide /usr/bin/rmic (rmic) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jaotc to provide /usr/bin/jaotc (jaotc) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jhsdb to provide /usr/bin/jhsdb (jhsdb) in auto mode
Setting up openjdk-11-jdk:amd64 (11.0.28+6-lubuntul-24.04.1) ...
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
jtnshmr2004@jatin:~$ java -version
openjdk version "11.0.28" 2025-07-15
OpenJDK Runtime Environment (build 11.0.28+6-post-Ubuntu-lubuntul24.04.1)
OpenJDK 64-Bit Server VM (build 11.0.28+6-post-Ubuntu-lubuntul24.04.1, mixed mode, sharing)
jtnshmr2004@jatin:~$ nano ~/.bashrc
jtnshmr2004@jatin:~$ source ~/.bashrc
jtnshmr2004@jatin:~$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64
jtnshmr2004@jatin:~$
```

### Conclusion:

Java was successfully installed and configured on the Unix/Linux machine, and the system is ready to compile and run Java programs.

