EXPERIMENT NO.: 1

AIM: Installation of Java on Unix/Linux Machine (Without VirtualBox)

■ This guide uses WSL (Windows Subsystem for Linux) to simulate a Linux environment directly on Windows.

Step 1: Enable WSL

1. Open PowerShell as Administrator
2. Run the following command: wsl –install

```
C:\Windows\System32>wsl --install
Downloading: Ubuntu
[                              1.4%
```

3. Restart your computer when prompted.
4. 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.

```
Create a default Unix user account: kushagarthakur
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 3: Update the Package List Run

the following command: sudo apt

update

```
kushagarthakur@keshu:/mnt/c/Users/kusha$ sudo apt update
[sudo] password for kushagarthakur:
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1083 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [187 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [881 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [195 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [17.0 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1631 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [361 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [18.5 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [4288 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [380 B]
Get:20 http://archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
```

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

```
kushagarthakur@keshu:/mnt/c/Users/kusha$ sudo apt install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2-data libasound2t64
  libatk-wrapper-java libatk-wrapper-java-jni libgif7 libice-dev libice6 libnspr4 libnss3 libpcsclite1
  libpthread-stubs0-dev libsm-dev libsm6 libx11-dev libxau-dev libxaw7 libxcb-shape0 libxcb1-dev libxdmcp-dev libxft2
  libxkbfile1 libxmu6 libxpm4 libxt-dev libxt6t64 libxv1 libxxf86dga1 openjdk-11-jdk-headless openjdk-11-jre
  openjdk-11-jre-headless x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre alsa-utils libasound2-plugins libice-doc pcscd libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo
  openjdk-11-source visualvm libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei fonts-indic mesa-utils
Recommended packages:
  luit
The following NEW packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2-data libasound2t64
  libatk-wrapper-java libatk-wrapper-java-jni libgif7 libice-dev libice6 libnspr4 libnss3 libpcsclite1
  libpthread-stubs0-dev libsm-dev libsm6 libx11-dev libxau-dev libxaw7 libxcb-shape0 libxcb1-dev libxdmcp-dev libxft2
  libxkbfile1 libxmu6 libxpm4 libxt-dev libxt6t64 libxv1 libxxf86dga1 openjdk-11-jdk openjdk-11-jdk-headless
  openjdk-11-jre openjdk-11-jre-headless x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 40 newly installed, 0 to remove and 9 not upgraded.
Need to get 125 MB of archives.
After this operation, 284 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 alsa-topology-conf all 1.2.5.1-2 [15.5 kB]
```

Step 5: Verify Java Installation

Run: java -version

```
kushagarthakur@keshu:/mnt/c/Users/kusha$ java -version
openjdk version "11.0.28" 2025-07-15
OpenJDK Runtime Environment (build 11.0.28+6-post-Ubuntu-1ubuntu124.04.1)
OpenJDK 64-Bit Server VM (build 11.0.28+6-post-Ubuntu-1ubuntu124.04.1, mixed mode, sharing)
```

Step 6: Set JAVA_HOME Environment Variable (Optional)

1. Open .bashrc file: nano ~/.bashrc

```
kushagarthakur@keshu:/mnt/c/Users/kusha$ nano ~/.bashrc
```

2. Add the following lines at the end:

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

PATH=$JAVA_HOME/bin:$PATH

```
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
 export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
 export PATH=$JAVA_HOME/bin:$PATH
```

3. Save and exit (Ctrl+X, then Y, then Enter)
4. Reload bashrc: source ~/.bashrc
5. Verify JAVA_HOME is set: echo $JAVA_HOME

```
kushagarthakur@keshu:/mnt/c/Users/kusha$ nano ~/.bashrc
kushagarthakur@keshu:/mnt/c/Users/kusha$ $JAVA_HOME
```

■ You now have Java installed on a Linux system simulated within Windows without using VirtualBox!