

# EXPERIMENT NO. 3

**Date Of Performance :**

**Date Of Submission :**

- AIM :** a) To practice/execute shell programs using Jenkins.  
b) To practice/execute parameterised Java programs using Jenkins.

## THEORY

Jenkins is an open-source server that is written entirely in Java. It lets you execute a series of actions to achieve the continuous integration process, that too in an automated fashion. This CI server runs in servlet containers such as Apache Tomcat.

Jenkins facilitates continuous integration and continuous delivery in software projects by automating parts related to build, test, and deployment. This makes it easy for developers to continuously work on the betterment of the product by integrating changes to the project. Jenkins automates the software builds in a continuous manner and lets the developers know about the errors at an early stage.

A strong Jenkins community is one of the prime reasons for its popularity. Jenkins is not only extensible but also has a thriving plugin ecosystem.

**Software builds using Jenkins** are possible using the build systems such as:

- Gradle
- Maven
- And more

**Automation testing** using test frameworks such as:

- Nose2
- PyTest
- Robot
- Selenium
- And more

## TO PRACTISE/EXECUTE SHELL PROGRAMS USING JENKINS

**Step 1:** Click on *Create new jobs*

The screenshot shows the Jenkins dashboard. At the top, there are links for 'New Item' and 'Build History'. Below that is a 'Build Queue' section indicating 'No builds in the queue.' To the right, there's a 'Create a job' button and a '+'. Underneath, there's a 'Set up a distributed build' section with options for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

**Step 2:** Give a name to project as “P1”, select Option “Free style project” and click on OK button.

The screenshot shows the Jenkins 'New Item' configuration page. At the top, there are three tabs: 'New Item - Jenkins', 'adsa - pranaybhandare765@gmail.com - Jenkins sh command fix', and a third tab which is partially visible. Below the tabs, the URL 'localhost:8080/view/all/newJob' is shown. The main content area has a heading 'New Item'. It asks for 'Enter an item name' with a text input field containing 'Experiment 3'. Under 'Select an item type', there are two options: 'Freestyle project' (selected) and 'Pipeline'. The 'Freestyle project' description states: 'Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.' The 'Pipeline' description states: 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.'

**Step 3:** To run simple shell scripts on Jenkins click on Build option select the Execute script from dropdown menu.

The screenshot shows the Jenkins 'Experiment 4 Configuration' page. At the top, there are three tabs: 'Experiment 4 Config - Jenkins', 'Inbox (16) - pranaybhandare765@gmail.com', and a third tab which is partially visible. Below the tabs, the URL 'localhost:8080/job/Experiment%204/configure' is shown. The main content area has a heading 'Configure'. On the left, there is a sidebar with links: General, Source Code Management, Triggers, Environment (selected), Build Steps, and Post-build Actions. In the main area, under 'Build Steps', it says 'Automate your build process with ordered tasks like code compilation, testing, and deployment.' A dropdown menu titled 'Add build step' is open, showing several options: 'Execute Windows batch command', 'Execute shell' (selected), 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. A tooltip for 'Execute shell' says: 'Run a shell script. You can use the \$env variable to send notifications, archiving artifacts, or triggering other jobs.'

**Step 4:** Write a simple shell command and click on apply followed by save button.

The screenshot shows the Jenkins 'Execute shell' configuration dialog. At the top, there is a link 'Build Steps'. The dialog has a title 'Execute shell ?'. It has a 'Command' section with a text area containing 'echo "Hello There I am Pranay"'. Below the text area, it says 'See the list of available environment variables'. At the bottom, there are 'Save' and 'Apply' buttons.

**Step 5:** Click on first build “1” followed by console output to see the output.

The screenshot shows the Jenkins interface for Experiment 4, build #5. The left sidebar has links for Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#5', and Timings. The main area is titled "Console Output" with a green checkmark icon. It displays the following log output:

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Experiment 4
[Experiment 4] $ cmd /c call C:\WINDOWS\TEMP\jenkins10403300715977827200.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Experiment 4>echo "Hello There I Am Pranay"
"Hello There I Am Pranay"

C:\ProgramData\Jenkins\.jenkins\workspace\Experiment 4>exit 0
Finished: SUCCESS
```

## Practise / Execute Parameterised JAVA Programs Using JENKINS.

**Step 1:** Create a freestyle project P2 in Jenkins.

The screenshot shows the Jenkins "New Item" creation page. The browser address bar shows "localhost:8080/view/all/newJob". The page title is "New Item". A warning message says "A job already exists with the name 'P2'". Below it, a section for "Select an item type" shows the "Freestyle project" option, which is described as a "Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.".

**Step 2:** Click on general menu and select option this project is parameterize. Select String parameter and specify name as “First-Name”

The screenshot shows the Jenkins "Configuration" page for project P2. Under the "General" tab, the "This project is parameterized" checkbox is checked. A "String Parameter" configuration panel is open, showing a "Name" field with the value "First Name".

**Step 3:** Click on add parameter and select choice parameter. Take second parameter as choice parameter

The screenshot shows the Jenkins configuration interface for a job named 'P2'. On the left, there's a sidebar with options like General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The 'General' tab is selected. On the right, under 'Add Parameter', a dropdown menu is open, showing various parameter types: Boolean Parameter, Choice Parameter (which is highlighted), Credentials Parameter, File Parameter, Multi-line String Parameter, Password Parameter, Run Parameter, and String Parameter. Below the dropdown is a button labeled 'Add Parameter ^'.

**Step 4:** Specify name as “City” and add the choices in each line

The screenshot shows the Jenkins configuration interface for job 'P2'. The 'General' tab is selected. A 'Choice Parameter' is configured with the name 'My Self'. Under 'Choices', there is a text area containing three lines of text: 'Name : Pranay Bhandarte', 'Class : TE / IT-A / 09', and 'Experiment : 3'. A red error message '① Requires Choices.' is displayed below the choices input field. There is also a 'Description' field at the bottom.

**Step 5:** Click on build with parameters and specify the values

The screenshot shows the Jenkins 'Build with Parameters' interface for project 'P2'. On the left, there's a sidebar with Status, Changes, Workspace, Build with Parameters (which is selected), Configure, Delete Project, and Rename. The main area shows the project name 'Project P2' and a message 'This build requires parameters:'. Under 'Build with Parameters', there are two fields: 'First Name' with the value 'Ronaldo' and 'My Self' with a dropdown menu containing the same three lines of text as in Step 4: 'Name : Pranay Bhandarte', 'Class : TE / IT-A / 09', and 'Experiment : 3'. The 'Name : Pranay Bhandarte' option is currently selected.



Jenkins / P2 / #1 / Console Output

Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Parameters



## Console Output

Started by user admin

Running as SYSTEM

Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\P2

Finished: SUCCESS

**Conclusion:** Hence we can conclude that we have learned and implemented shell programs and parameterized Java programs using Jenkins.