

Experiment No. 5

Aim: To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server

Theory:

Programming in Jenkins:

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.” In simple way, Continuous integration (CI) is the practice of frequently building and testing each change done to your code automatically.

Jenkins is a self-contained, open-source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Our first job will execute the shell commands. The freestyle project provides enough options and features to build the complex jobs that you will need in your projects.

Example 1

Example 1.1: Deploying a freestyle app in Jenkins

Creating a job:

Start building your software project


Create a job





Naming the job and setting it as freestyle:


Enter an item name


» Required field


**Freestyle project**
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.

**Pipeline**
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.

**Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.


**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

Selecting build type as “Execute shell”:

Build Steps

Add build step ^

 Filter

Execute Windows batch command

Execute shell

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Set build status to "pending" on GitHub commit

Entering a simple command for the shell execution:

Build Steps

Execute shell ?

Command

See [the list of available environment variables](#)

```
echo "Hello TSEC"
```

Advanced ▾

Applying and saving the project configuration:

Save

Apply

✓ Saved

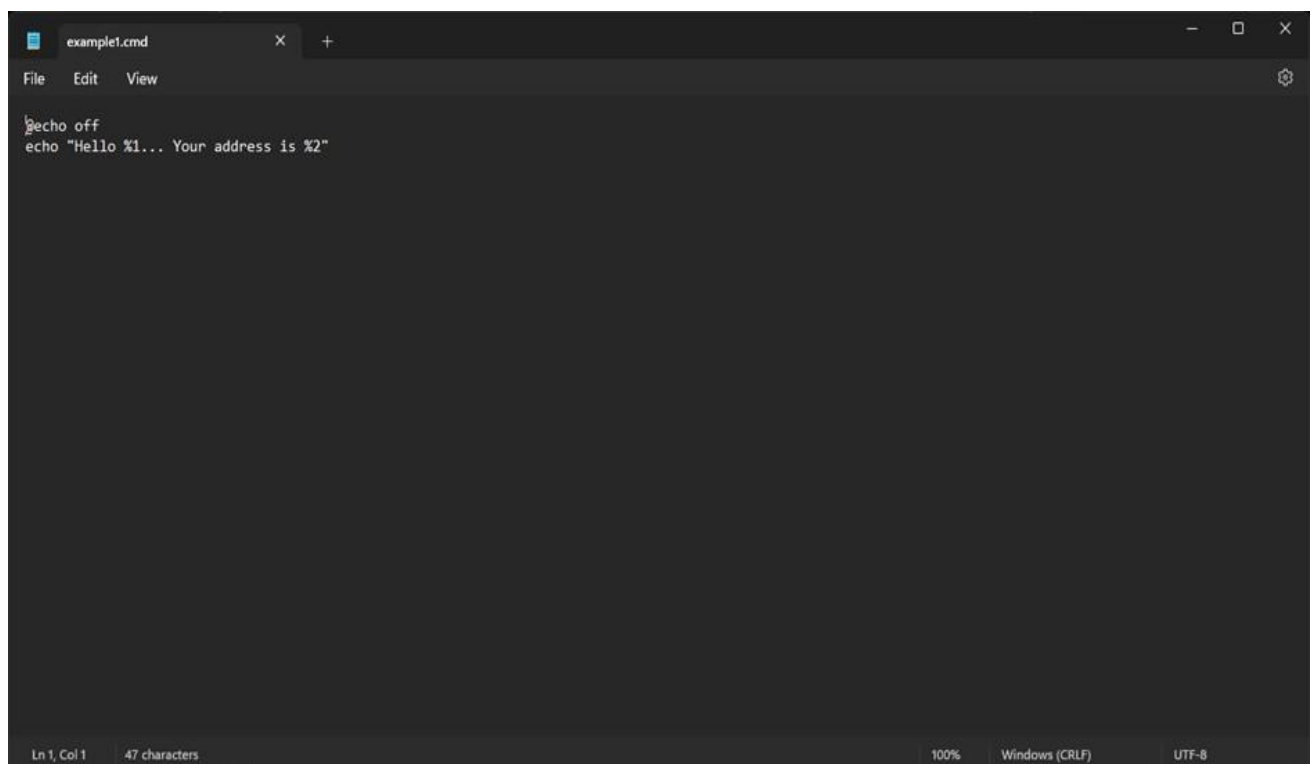
Building the project:

Console output (after building):



Example 1.2: Taking parameters through files

Contents of script example1.cmd:



Executing script example1.cmd on the terminal:

```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\AI&DS 202>Microsoft Windows [Version 10.0.22631.3155] (c) Microsoft Corporation. All rights reserved.
'Microsoft' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPM>example1.cmd
The system cannot find the path specified.

C:\Users\AI&DS 202>"Hello... Your address is "
'"Hello... Your address is "' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPM>example1.cad Tanishq
The system cannot find the path specified.

C:\Users\AI&DS 202>"Hello Tanihsq... Your address is "
'"Hello Tanihsq... Your address is "' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPM>example1.cmd Tanishq Girgaon "Helle Tanishq... Your address is Gi
rgaon"
The system cannot find the path specified.
```

Modifying the Jenkins project to execute the script while supplying required parameters:

Build Steps

Execute Windows batch command ?

Command

See [the list of available environment variables](#)

C:\Admin\Academics\TSEC\Start3\SEPM\example1.cmd Siddhant Goregaon

Advanced

Add build step

Console output after building the modified project:

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build #4

Previous Build

Console Output

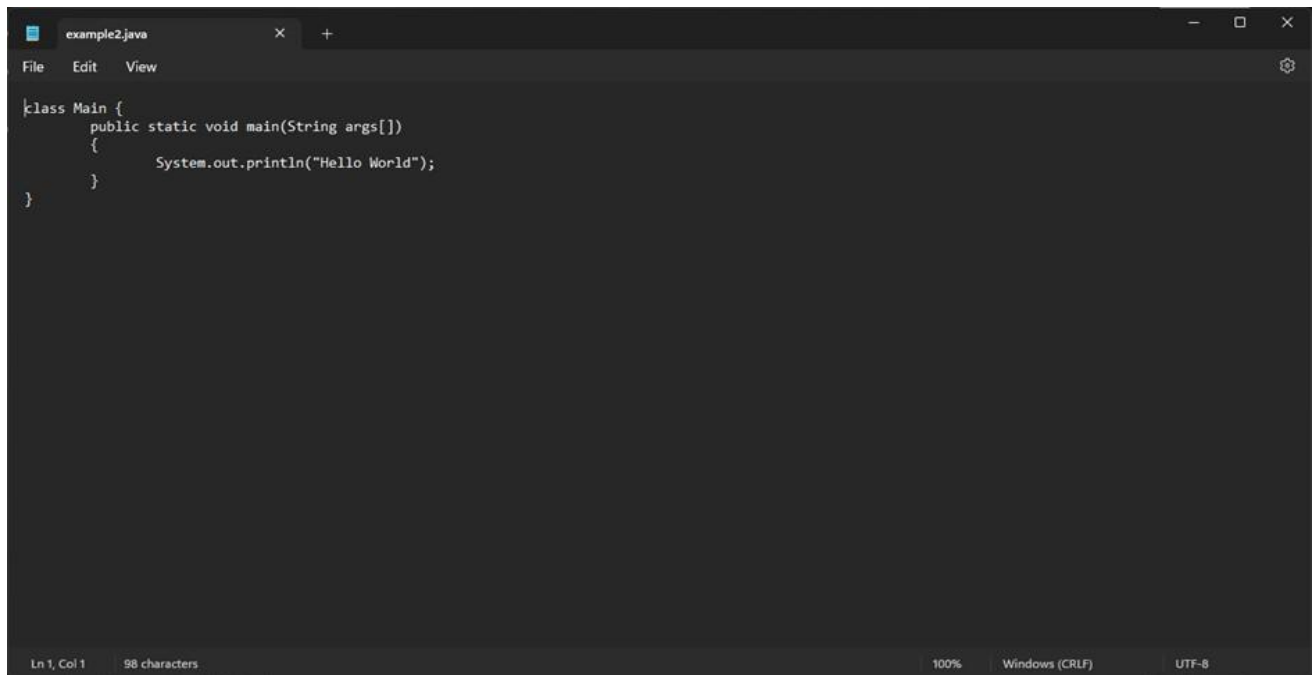
Started by user Siddhant Chatur
Running as SYSTEM
Building in workspace C:\ProgramData\jenkins\jenkins\workspace\example1
[example1] \$ cmd /c call C:\MNOOD\TEMP\jenkins\07989982828588158.bat

C:\ProgramData\jenkins\jenkins\workspace\example1>C:\Admin\Academics\T50C\Start\SEPM\example1.cmd Siddhant Doregan
"Hello Siddhant... Your address is Doregan"
Finished: SUCCESS

Example 2

Example 2.1: Running a Java program under Jenkins

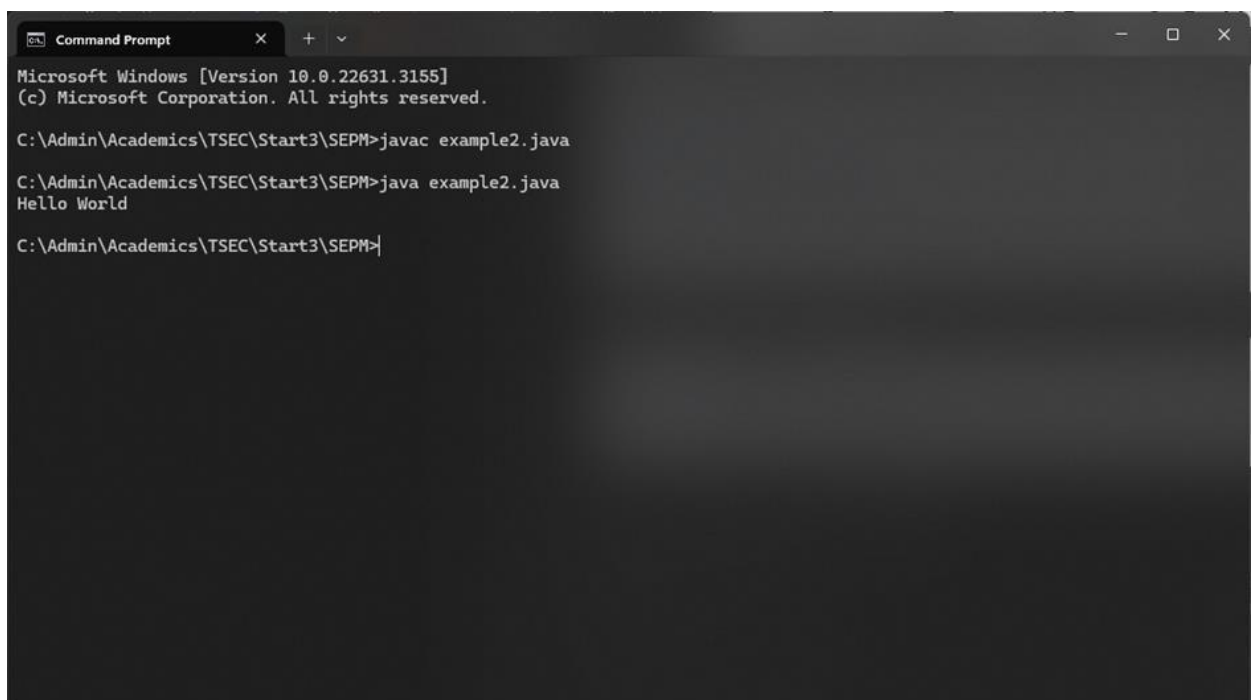
Creating a simple Java program:

A screenshot of an IDE window titled 'example2.java'. The window has a menu bar with 'File', 'Edit', and 'View'. The code editor contains the following Java code:

```
class Main {  
    public static void main(String args[])  
    {  
        System.out.println("Hello World");  
    }  
}
```

The status bar at the bottom shows 'Ln 1, Col 1', '98 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.

Compiling and running the program on the terminal:

A screenshot of a Windows Command Prompt window titled 'Command Prompt'. The window shows the following commands and output:

```
Microsoft Windows [Version 10.0.22631.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Admin\Academics\TSEC\Start3\SEPM>javac example2.java  
  
C:\Admin\Academics\TSEC\Start3\SEPM>java example2.java  
Hello World  
  
C:\Admin\Academics\TSEC\Start3\SEPM>|
```

Creating a new freestyle project:

The screenshot shows the Jenkins 'Create new item' dialog box. At the top, there is a breadcrumb 'Dashboard > All' and a title 'Enter an item name'. Below the title is a text input field containing 'Example2', with a red asterisk and the text '* Required field' below it. A list of project types is shown below the input field:

- Freestyle project**: 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.
- Pipeline**: 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.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**: Creates a set of multibranch project subfolders by scanning for repositories.

At the bottom, there is a note: 'If you want to create a new item from other existing, you can use this option:' followed by an 'OK' button.

Configure new project:

Build Steps

The screenshot shows the Jenkins 'Build Steps' configuration panel. At the top, there is a title 'Execute Windows batch command' with a help icon. Below the title is a 'Command' section with a link 'See the list of available environment variables'. A text input field contains the following commands:

```
javac C:\Admin\Academics\TSEC\Start3\SEPM\example2.java
java C:\Admin\Academics\TSEC\Start3\SEPM\example2.java
```

Below the input field is an 'Advanced' dropdown menu. At the bottom, there is an 'Add build step' button.

Console output after building:

✓ Console Output

```
Started by user Siddhant Chetlur
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Example2
[Example2] $ cmd /c call C:\WINDOWS\TEMP\jenkins15296462484398614135.bat

C:\ProgramData\Jenkins\jenkins\workspace\Example2>javac C:\Admin\Academics\TSEC\Start3\SEPH\example2.java

C:\ProgramData\Jenkins\jenkins\workspace\Example2>java C:\Admin\Academics\TSEC\Start3\SEPH\example2.java
Hello World

C:\ProgramData\Jenkins\jenkins\workspace\Example2>exit 0
Finished: SUCCESS
```


Example 3


Example 3.1: Parameterise build


Creating a new freestyle project:


Enter an item name


» Required field


**Freestyle project**
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.

**Pipeline**
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.

**Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

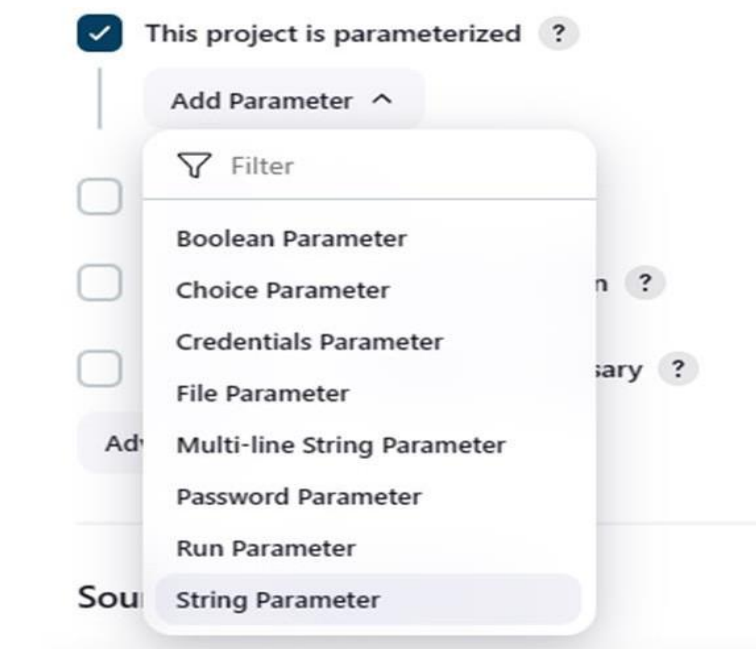
**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

OK

Cancel

Enabling parameterisation and adding a String parameter:



Configuring the string parameter as Fname:

String Parameter ?

Name ?

Fname

Default Value ?

Description ?

Plain text

Preview

☐

Trim the string ?

Adding a choice parameter and configuring it as City with the following choices:

Choice Parameter ?

Name ?

City

Choices ?

- Bandra
- Kalyan
- Dombivali
- Churchgate
- Thane
- Dadar

Description ?

Plain text [Preview](#)

Creating a script which takes 2 arguments for name and city:

```
C:\Users\AI&DS 202>Microsoft Windows [Version 10.0.22631.3155] (c) Microsoft Corporation. All rights reserved.
'Microsoft' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPH>example3.cnd
The system cannot find the path specified.

C:\Users\AI&DS 202>Hello your name is and your city is
'Hello' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPH example3.cmd Tanishq
The system cannot find the path specified.

C:\Users\AI&DS 202>Hello your name is Tanishq and your city is
'Hello' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPM>example3.cmd Tansishq Bandra
The system cannot find the path specified.

C:\Users\AI&DS 202>Hello your name is Tanishq and your city is Bandra
'Hello' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPH
```

Configuring build steps:

Build Steps

Execute Windows batch command ?

Command

See [the list of available environment variables](#)

```
C:\Admin\Academics\TSEC\Start3\SEPM\example3.cmd %Fname% %City%
```

Advanced ▾

Add build step ▾

Console output after building:

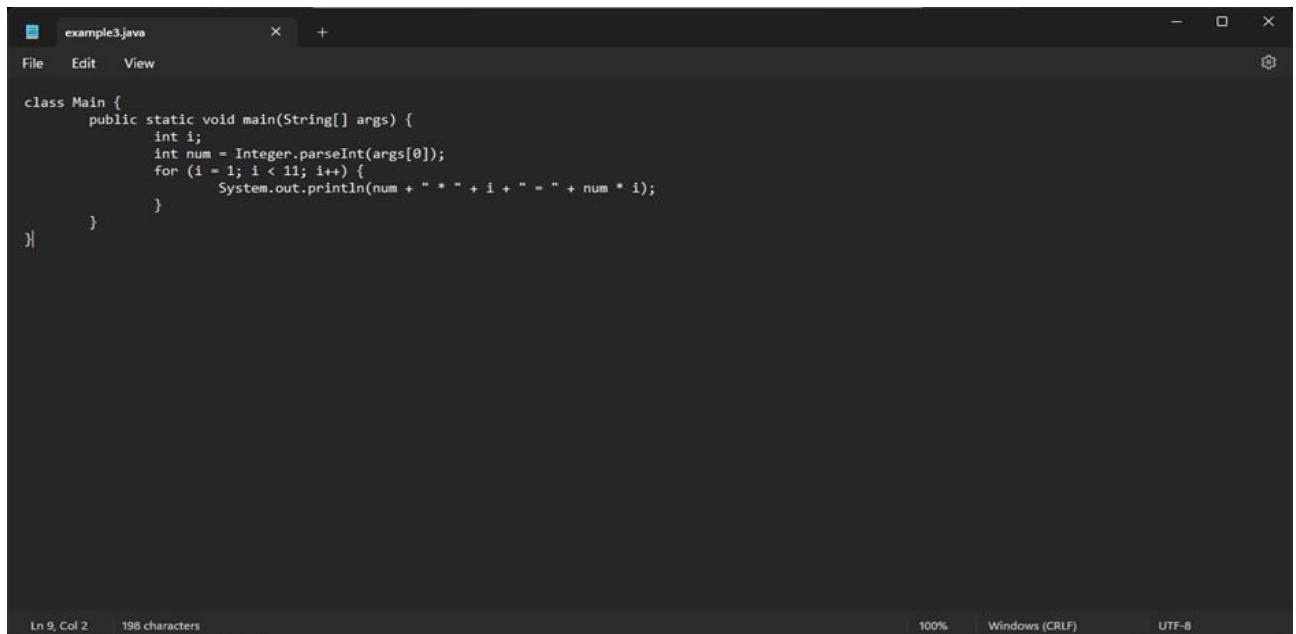
✓ Console Output

```
Started by user Siddhant Chetlur
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Example3
[Example3] $ cmd /c call C:\WINDOWS\TEMP\jenkins14094536165150986151.bat

C:\ProgramData\Jenkins\jenkins\workspace\Example3>C:\Admin\Academics\TSEC\Start3\SEPM\example3.cmd Siddhant Bandra
Hello your name is Siddhant and your city is Bandra
Finished: SUCCESS
```

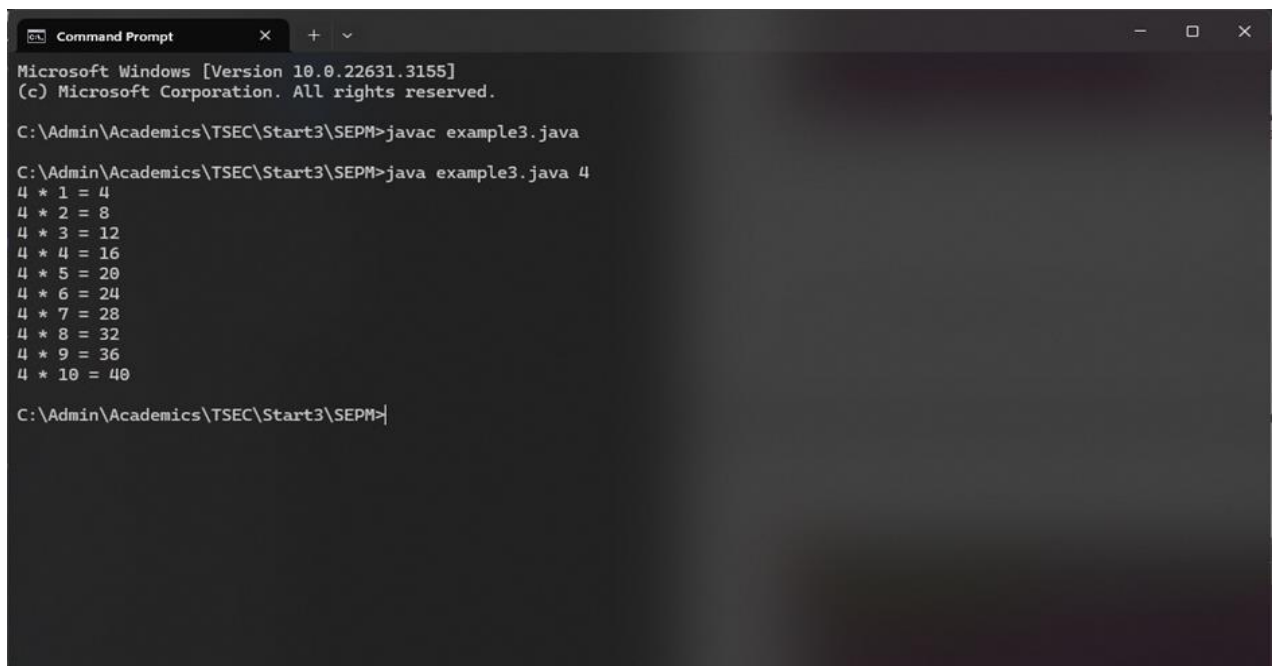
Example 3.2: Running a Java program with parameters

Creating a Java program with an input argument:



```
class Main {  
    public static void main(String[] args) {  
        int i;  
        int num = Integer.parseInt(args[0]);  
        for (i = 1; i < 11; i++) {  
            System.out.println(num + " * " + i + " = " + num * i);  
        }  
    }  
}
```

Testing the program on the terminal:





```
Microsoft Windows [Version 10.0.22631.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Admin\Academics\TSEC\Start3\SEPM>javac example3.java  
  
C:\Admin\Academics\TSEC\Start3\SEPM>java example3.java 4  
4 * 1 = 4  
4 * 2 = 8  
4 * 3 = 12  
4 * 4 = 16  
4 * 5 = 20  
4 * 6 = 24  
4 * 7 = 28  
4 * 8 = 32  
4 * 9 = 36  
4 * 10 = 40  
  
C:\Admin\Academics\TSEC\Start3\SEPM>
```


Creating a new freestyle project:


Enter an item name


» Required field


**Freestyle project**
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.

**Pipeline**
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.

**Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Parameterise the project by adding a string parameter as follows:

☒ This project is parameterized ?

String Parameter ?

Name ?

Default Value ?

Description ?

Plain text [Preview](#)

☐ Trim the string ?

Add Parameter

Configure the build steps:

Build Steps

≡ Execute Windows batch command ?

Command

See the list of available environment variables

```
javac C:\Admin\Academics\TSEC\Start3\SEPM\example3.java
java C:\Admin\Academics\TSEC\Start3\SEPM\example3.java %num%
```

Advanced ▾

Add build step ▾

Entering the parameter for the build:

Project Example4

This build requires parameters:

num

▶ Build

Cancel

Console output after building:

✓ Console Output

```
Started by user Siddhant Chetlur
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Example4
[Example4] $ cmd /c call C:\WINDOWS\TEMP\jenkins15119185770823247708.bat

C:\ProgramData\Jenkins\jenkins\workspace\Example4>javac C:\Admin\Academics\TSEC\Start3\SEPM\example3.java

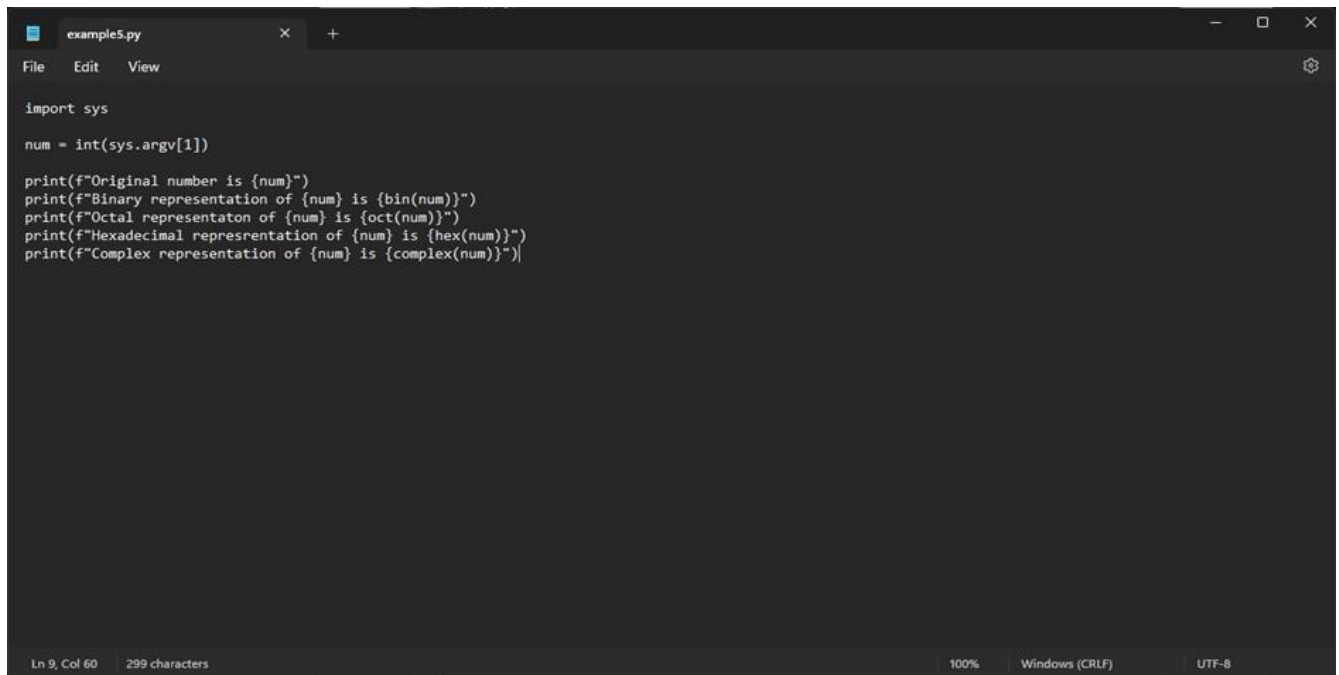
C:\ProgramData\Jenkins\jenkins\workspace\Example4>java C:\Admin\Academics\TSEC\Start3\SEPM\example3.java 25
25 * 1 = 25
25 * 2 = 50
25 * 3 = 75
25 * 4 = 100
25 * 5 = 125
25 * 6 = 150
25 * 7 = 175
25 * 8 = 200
25 * 9 = 225
25 * 10 = 250

C:\ProgramData\Jenkins\jenkins\workspace\Example4>exit 0
Finished: SUCCESS
```

Example 5

Example 5.1: Running a Python program

Creating a simple Python script:



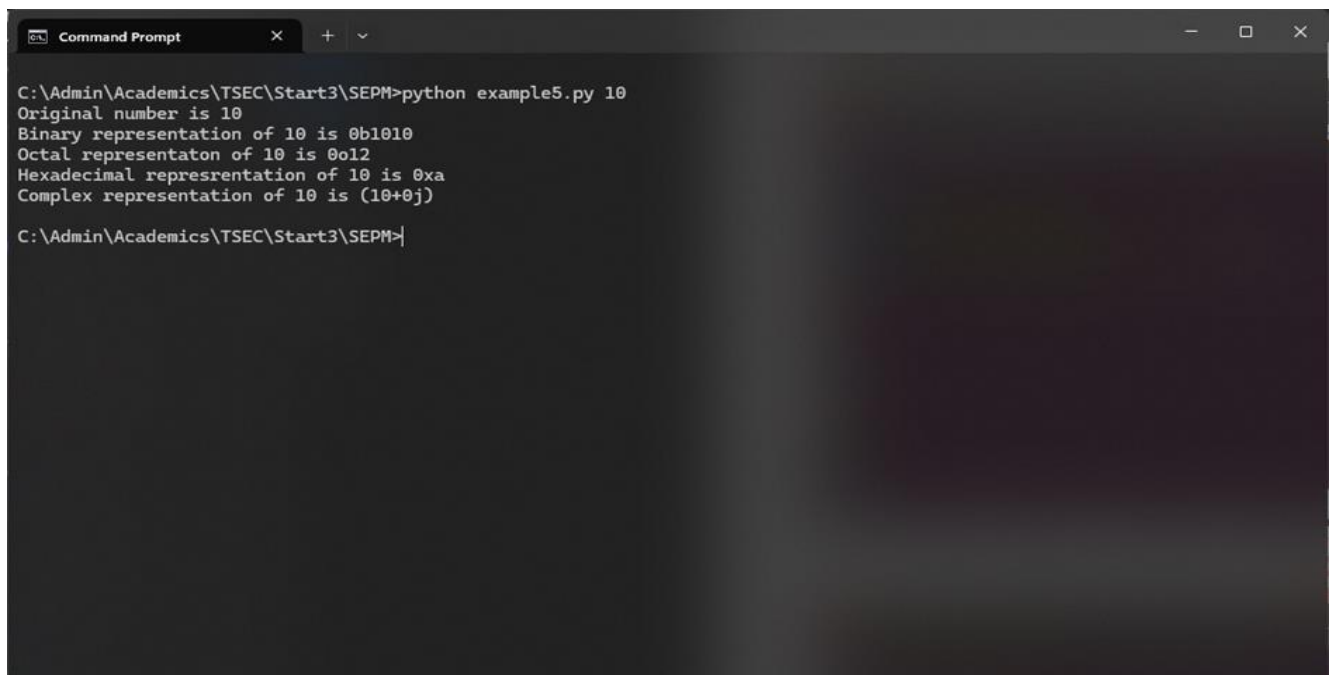
```
example5.py
File Edit View
import sys

num = int(sys.argv[1])

print(f"Original number is {num}")
print(f"Binary representation of {num} is {bin(num)}")
print(f"Octal representation of {num} is {oct(num)}")
print(f"Hexadecimal representation of {num} is {hex(num)}")
print(f"Complex representation of {num} is {complex(num)}")

Ln 9, Col 60 299 characters 100% Windows (CRLF) UTF-8
```

Running the Python script on the terminal:





```
Command Prompt
C:\Admin\Academics\TSEC\Start3\SEPM>python example5.py 10
Original number is 10
Binary representation of 10 is 0b1010
Octal representation of 10 is 0o12
Hexadecimal representation of 10 is 0xa
Complex representation of 10 is (10+0j)
C:\Admin\Academics\TSEC\Start3\SEPM>
```


Creating a new freestyle project:


Enter an item name


» Required field


**Freestyle project**
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.

**Pipeline**
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.

**Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Parameterising the project with a string parameter as follows:

☒ This project is parameterized ?

String Parameter ?

Name ?

Default Value ?

Description ?

Plain text: [Preview](#)

☐ Trim the string ?

Add Parameter

Configuring the build steps:

Build Steps

≡ Execute Windows batch command ?

Command

See [the list of available environment variables](#)

```
python C:\Admin\Academics\TSEC\Start3\SEPM\example5.py %num%
```

Advanced ▾

Add build step ▾

Setting the parameter for the build:

Project Example5

This build requires parameters:

num

10

▶ Build

Cancel

Console output after building:

✓ Console Output

```
Started by user: Siddhant Chetlur
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Example5
[Example5] $ cmd /c call C:\WINDOWS\TEMP\jenkins11157306491994478222.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Example5>python C:\Admin\Academics\TSEC\Start3\SEPM\example5.py 10
Original number is 10
Binary representation of 10 is 0b1010
Octal representation of 10 is 0o12
Hexadecimal representation of 10 is 0xa
Complex representation of 10 is (10+0j)

C:\ProgramData\Jenkins\.jenkins\workspace\Example5>exit 0
Finished: SUCCESS
```

Some Screenshots:

The image shows two screenshots of the Jenkins web interface. The top screenshot is the 'Configure' page for a pipeline named 'AIDS_Pipeline'. The 'Advanced Project Options' tab is selected, showing the 'Pipeline' definition. The 'Script' section contains a Groovy script defining a pipeline with four stages: 'build', 'test', 'deploy', and 'postdeploy'. Each stage has a single step with an 'echo' command. The 'Use Groovy Sandbox' checkbox is checked. The bottom screenshot is the Jenkins dashboard for the 'AIDS_Pipeline'. It shows the 'Status' tab, a 'Stage View' table with average stage times, and a 'Builds' list. The 'Stage View' table shows average stage times: Build (46ms), Test (44ms), Deploy (34ms), and Postdeploy (43ms). The 'Builds' list shows a single build (#1) with a status of 'Completed' and a time of 1:55 PM.

Configure Advanced

General
Build Triggers
Advanced Project Options
Pipeline

Pipeline

Definition

Pipeline script

```
1 pipeline {  
2   agent any  
3  
4   stages {  
5     stage('build') {  
6       steps {  
7         echo 'Building.. This is the build phase'  
8       }  
9     }  
10    stage('test') {  
11      steps {  
12        echo 'Testing.. This is the testing phase'  
13      }  
14    }  
15    stage('deploy') {  
16      steps {  
17        echo 'Deploying.... This is the deployment phase'  
18      }  
19    }  
20    stage('postdeploy') {  
21      steps {  
22        echo 'Postdeployment phase.....'  
23      }  
24    }  
25  }  
26 }  
27
```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Save Apply

Jenkins

Dashboard > AIDS_Pipeline

Status

<> Changes
▶ Build Now
⚙ Configure
🗑 Delete Pipeline
🔍 Full Stage View
✎ Rename
🔗 Pipeline Syntax

AIDS_Pipeline Add description

Stage View

Average stage times:
(Average full run time ~507ms)

	Build	Test	Deploy	Postdeploy
Average	46ms	44ms	34ms	43ms
Feb 27 11:58	46ms	44ms	34ms	43ms

Permalinks

- Last build (#1), 2 yr 0 mo ago
- Last stable build (#1), 2 yr 0 mo ago
- Last successful build (#1), 2 yr 0 mo ago
- Last completed build (#1), 2 yr 0 mo ago

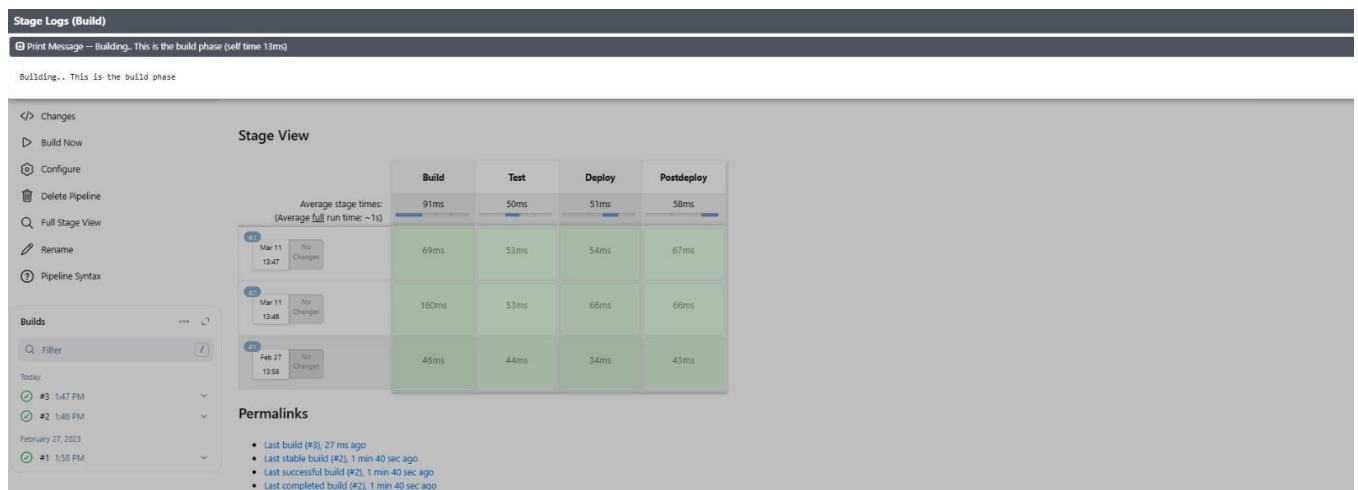
Builds

Filter

February 27, 2023

#1 1:55 PM

localhost:8080/jobs/AIDS_Pipeline/1 REST API Jenkins 2.402.2



Conclusion: Thus, we have successfully Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, created a pipeline script to Test and deploy an application over the tomcat server.

Krishna Rathod
T23-135