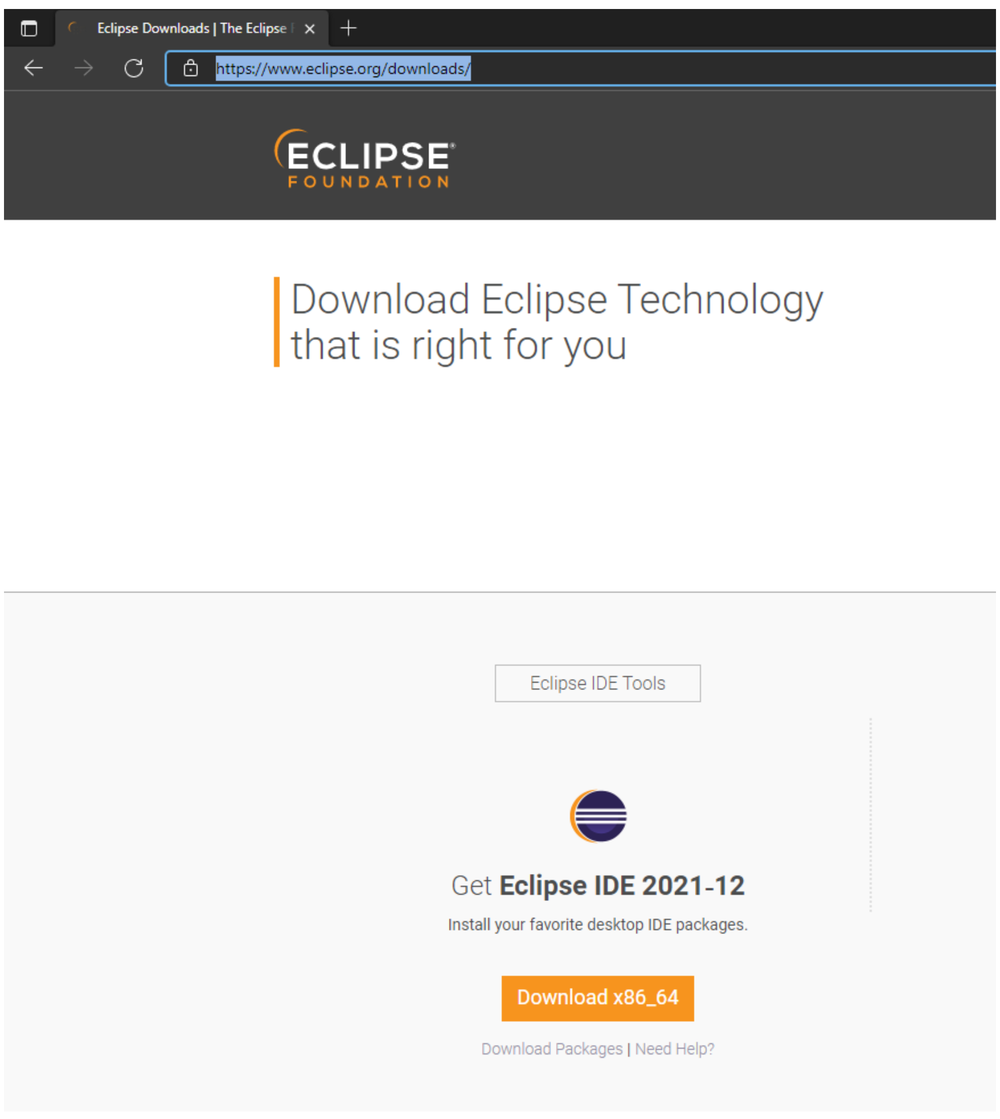
Working of DiscoDNC

Disco DNC is a tool for deterministic network calculus. This documentation provides insight on how to properly install DiscoDNC and how to configure it properly.

1. Download the Eclipse IDE from the Internet.

- *Download latest Eclipse IDE from the given link.*

  Link : [Eclipse Downloads | The Eclipse Foundation]


Eclipse Downloads | The Eclipse

https://www.eclipse.org/downloads/

ECLIPSE FOUNDATION

Download Eclipse Technology that is right for you

Eclipse IDE Tools

Get **Eclipse IDE 2021-12**

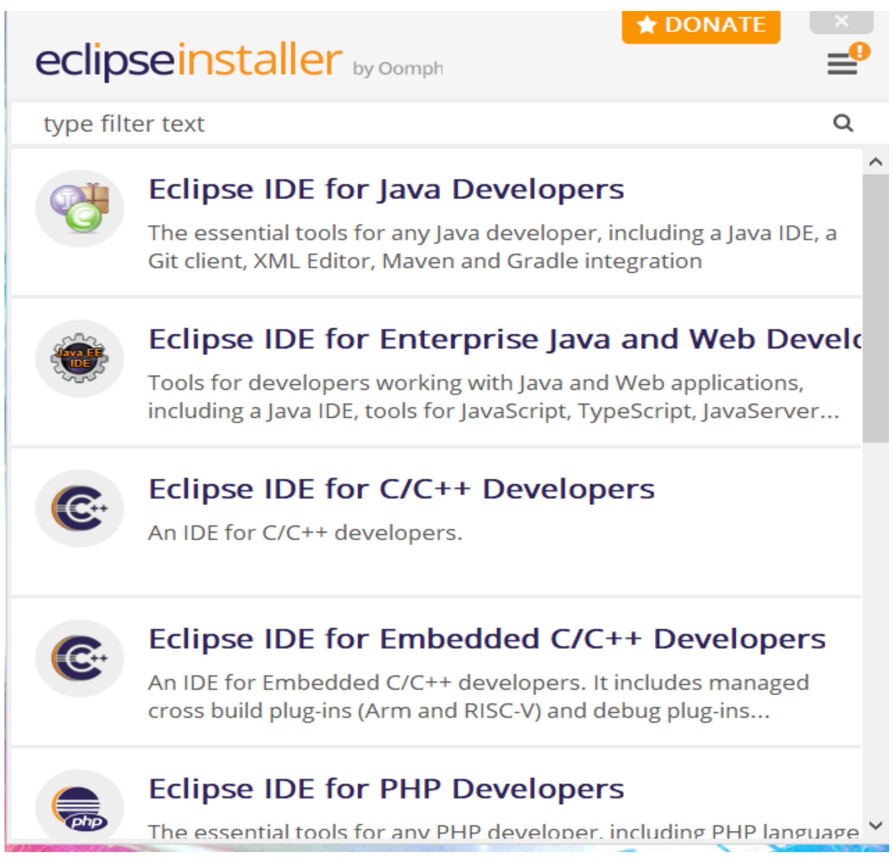
Install your favorite desktop IDE packages.

Download x86_64

Download Packages | Need Help?

2. Install the Eclipse into the desired directory.

- *Launch the Eclipse Installer.*
- *Select **Eclipse IDE for Java Developers**.*


eclipseinstaller by Oomph

DONATE

type filter text

Eclipse IDE for Java Developers
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration

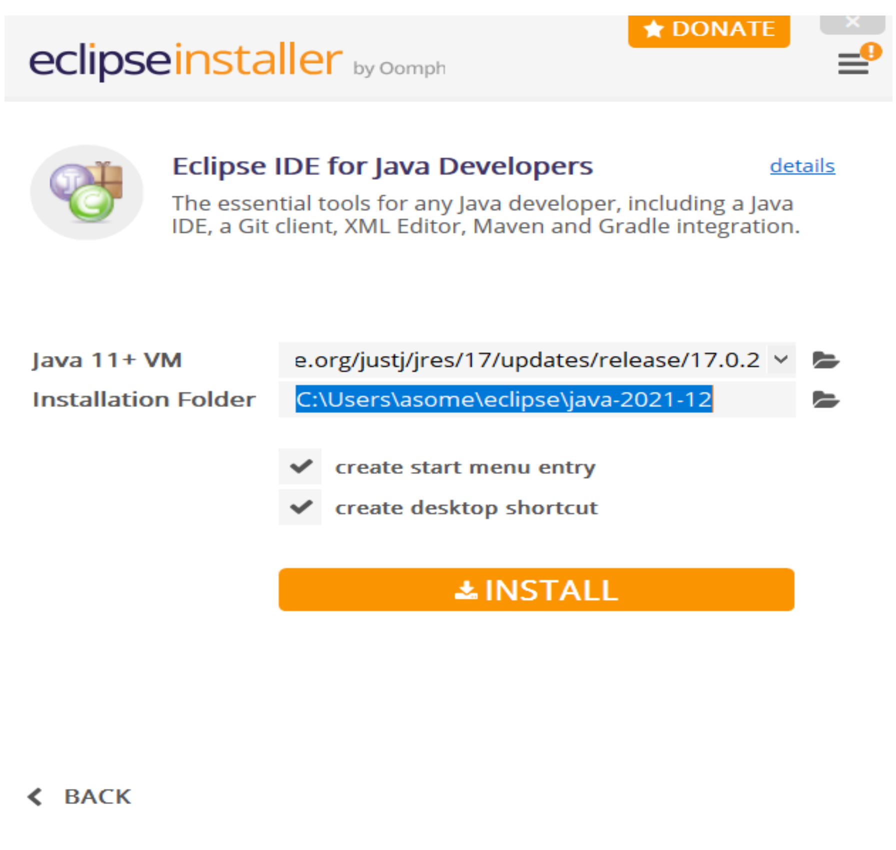
Eclipse IDE for Enterprise Java and Web Develo
Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer...

Eclipse IDE for C/C++ Developers
An IDE for C/C++ developers.

Eclipse IDE for Embedded C/C++ Developers
An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins...

Eclipse IDE for PHP Developers
The essential tools for any PHP developer, including PHP language

- *Select the directory where you want to install Eclipse.*


eclipseinstaller by Oomph

DONATE

Eclipse IDE for Java Developers — details
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.

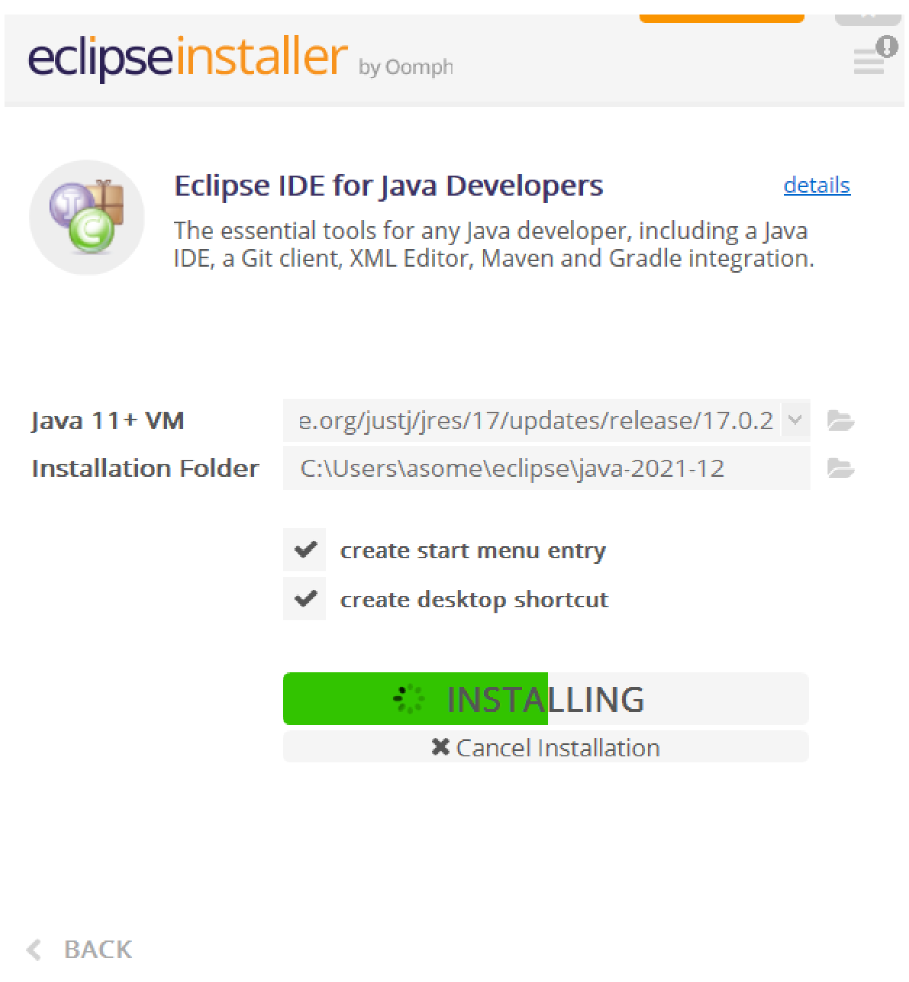
Java 11+ VM: e.org/justj/jres/17/updates/release/17.0.2
Installation Folder: C:\Users\asome\eclipse\java-2021-12

✓ create start menu entry
✓ create desktop shortcut

INSTALL

BACK

- *Click the **INSTALL** option*


eclipseinstaller by Oomph

Eclipse IDE for Java Developers — details

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.

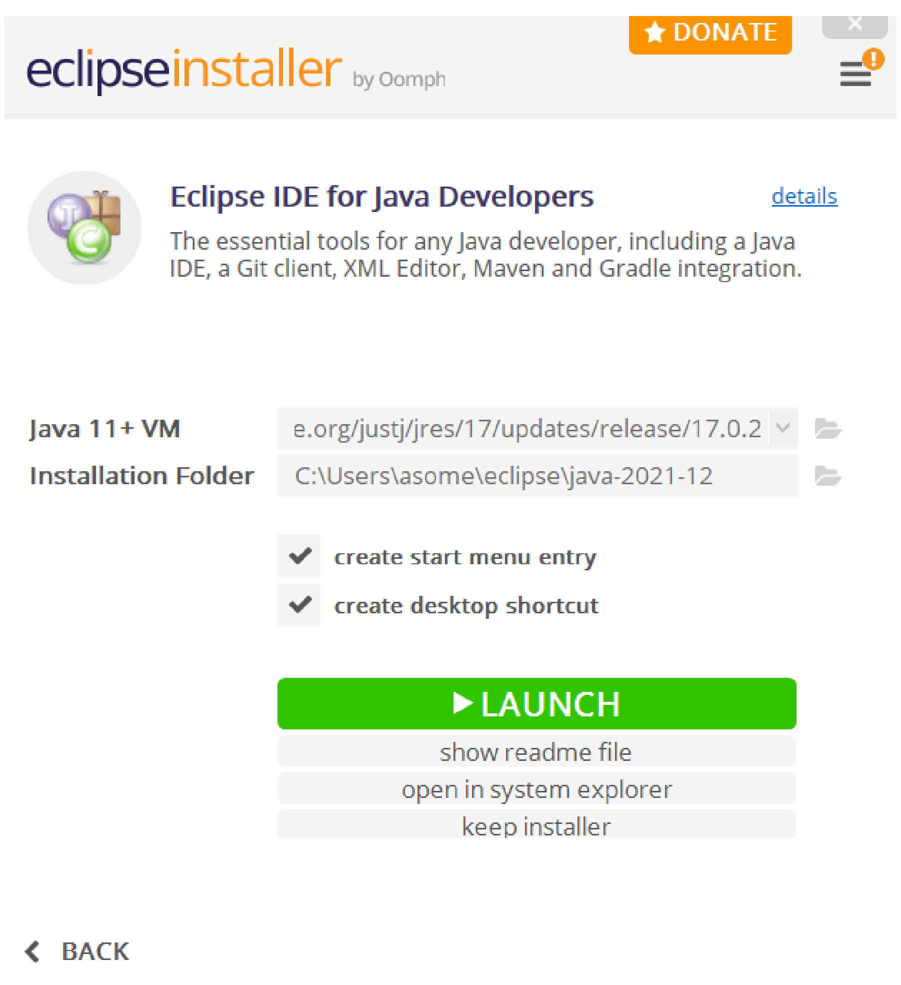
Java 11+ VM: e.org/justj/jres/17/updates/release/17.0.2

Installation Folder: C:\Users\asome\eclipse\java-2021-12

✓ create start menu entry

✓ create desktop shortcut

- *After Installation is complete, Click on **LAUNCH** option.*


eclipseinstaller by Oomph

Eclipse IDE for Java Developers — details

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.

Java 11+ VM: e.org/justj/jres/17/updates/release/17.0.2

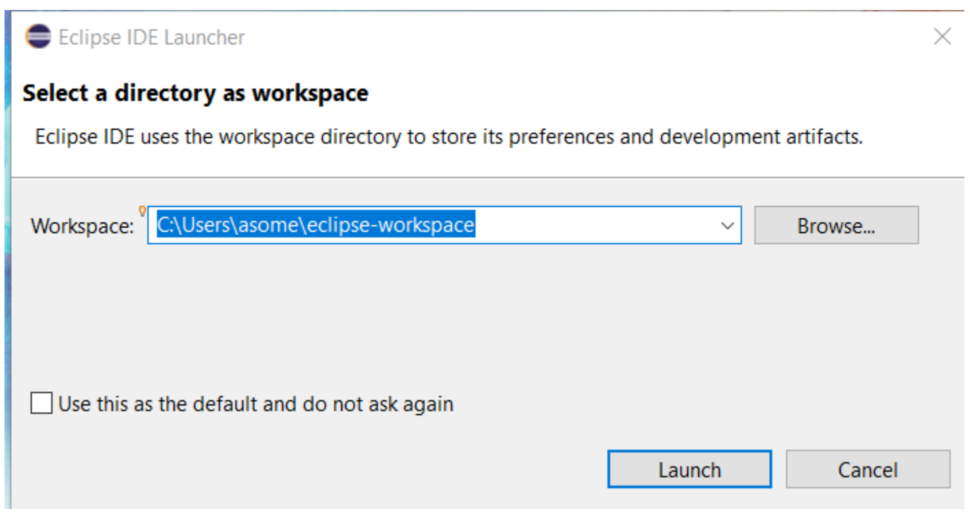
Installation Folder: C:\Users\asome\eclipse\java-2021-12

✓ create start menu entry

✓ create desktop shortcut

3. After Installation Launch the eclipse:

- *Select a directory as a workspace.*


Eclipse IDE Launcher

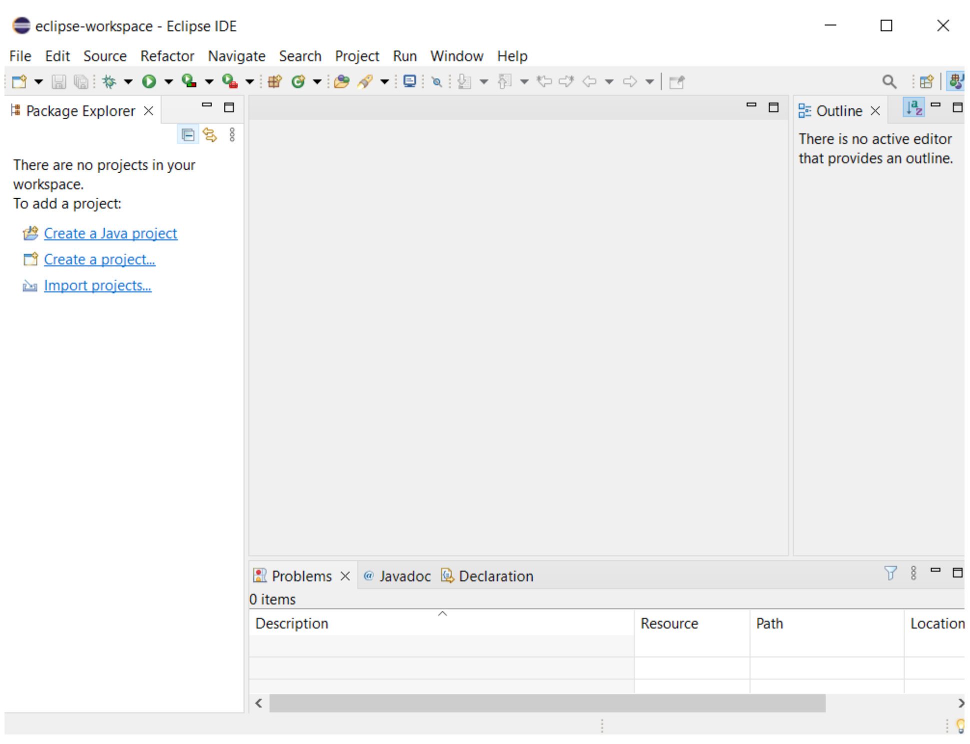
Select a directory as workspace

Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Users\asome\eclipse-workspace

☐ Use this as the default and do not ask again

- *Click on **Launch** Button.*
- *It will launch Eclipse Workspace.*


eclipse-workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Outline
Problems
Javadoc
Declaration
Description
Resource
Path
Location

There are no projects in your workspace.
To add a project:
Create a Java project
Create a project...
Import projects...

There is no active editor that provides an outline.

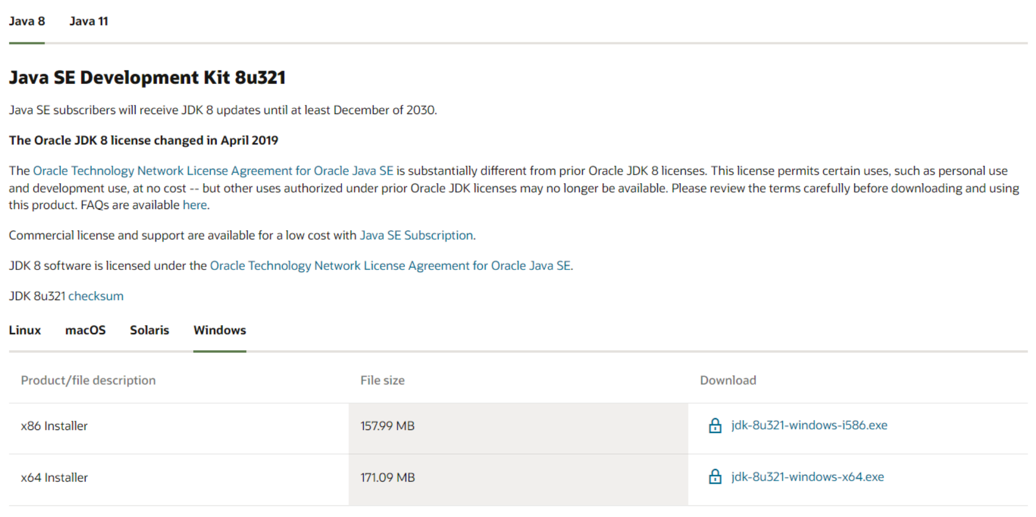
0 items

Since Disco DNC requires Java 8, we should download and install Java 8 in our system.

4. Download Java 8.

- *Download the Java 8 from the given link below.*

Link: https://www.oracle.com/java/technologies/downloads/#java8-windows



Java SE Development Kit 8u321

Java SE subscribers will receive JDK 8 updates until at least December of 2030.

The Oracle JDK 8 license changed in April 2019

The Oracle Technology Network License Agreement for Oracle Java SE is substantially different from prior Oracle JDK 8 licenses. This license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. FAQs are available here.

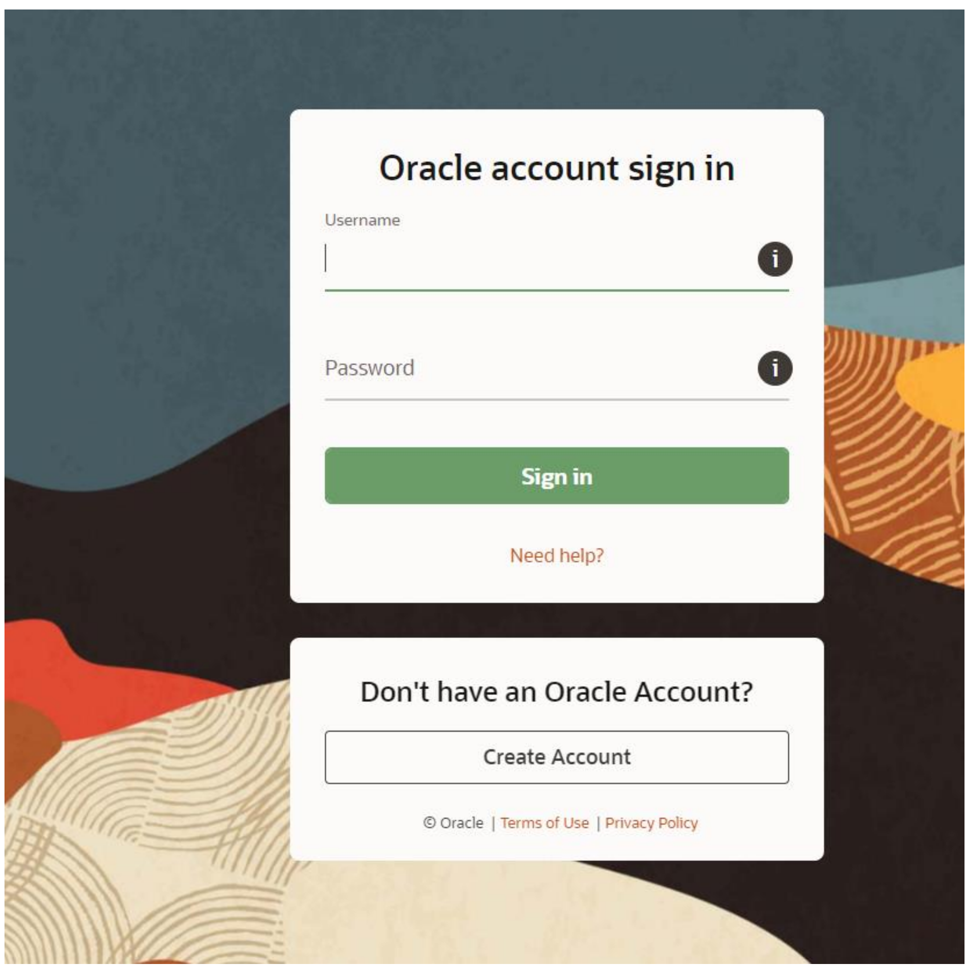
Commercial license and support are available for a low cost with Java SE Subscription.

JDK 8 software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE.

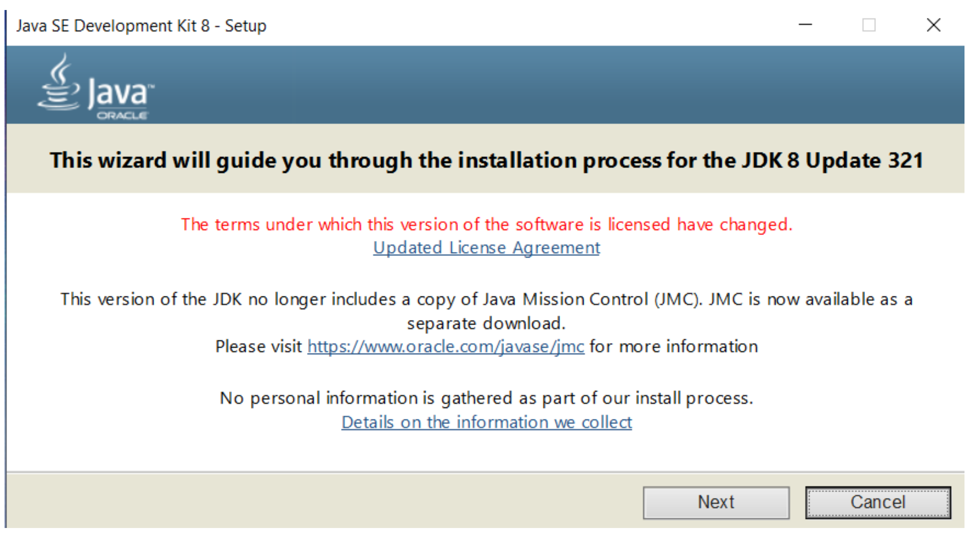
JDK 8u321 checksum

Product/file description	File size	Download
x86 Installer	157.99 MB	jdk-8u321-windows-i586.exe
x64 Installer	171.09 MB	jdk-8u321-windows-x64.exe

- *You will be redirected to the oracle login page. Login using an oracle account or create a new account.*


Oracle account sign in
Username
Password
Sign in
Need help?
Don't have an Oracle Account?
Create Account
© Oracle | Terms of Use | Privacy Policy

- *After logging to oracle, the download of java 8 will start.*
- *After download is complete, run the setup file and start the installation of Java 8.*

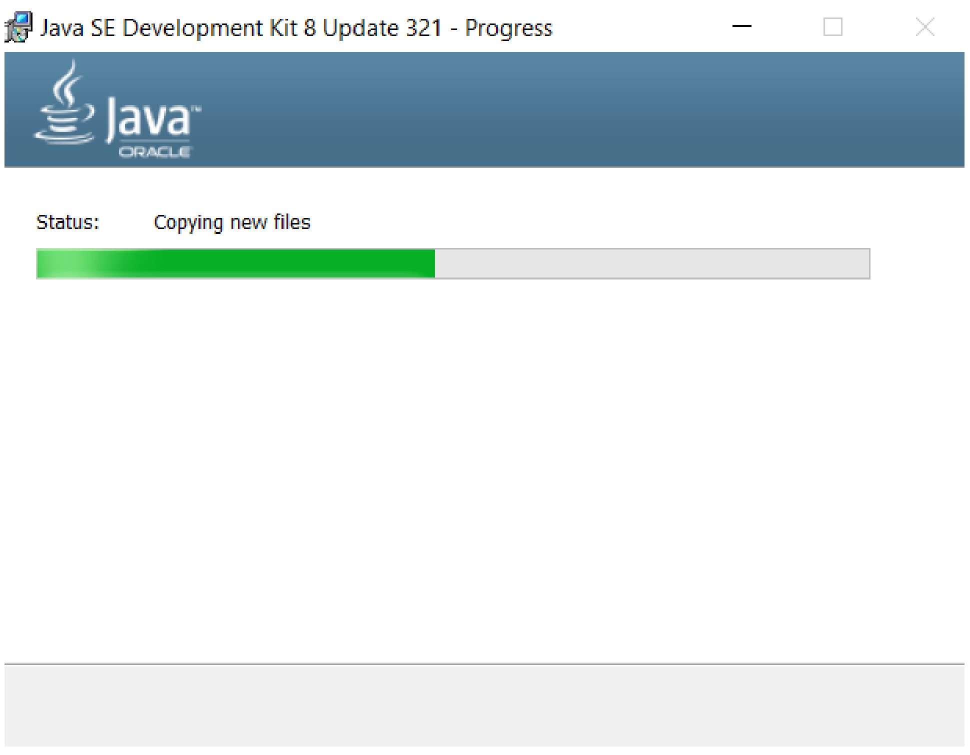


This wizard will guide you through the installation process for the JDK 8 Update 321

The terms under which this version of the software is licensed have changed.
Updated License Agreement

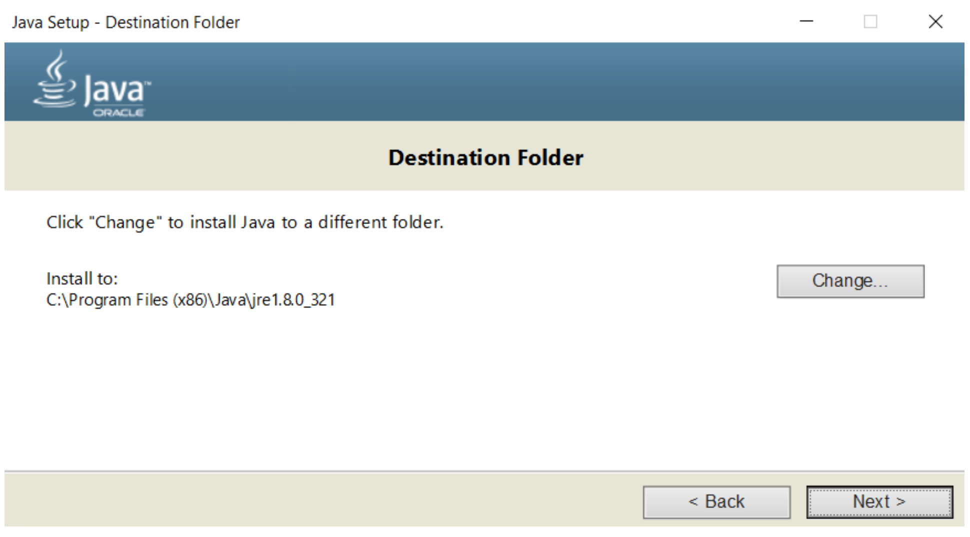
This version of the JDK no longer includes a copy of Java Mission Control (JMC). JMC is now available as a separate download.
Please visit https://www.oracle.com/javase/jmc for more information

No personal information is gathered as part of our install process.
Details on the information we collect


Java SE Development Kit 8 Update 321 - Progress

Status: Copying new files

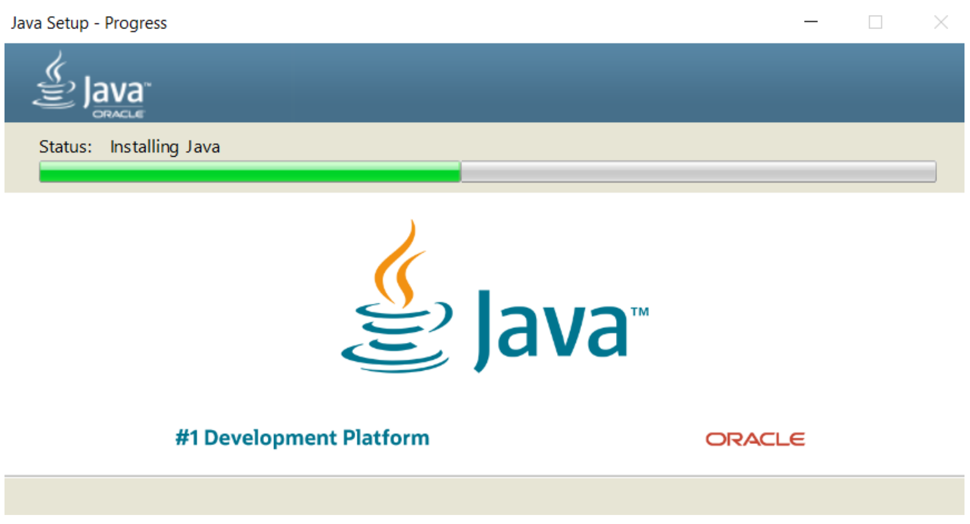
- *If needed change the **destination folder** of Java 8.*


Java Setup - Destination Folder

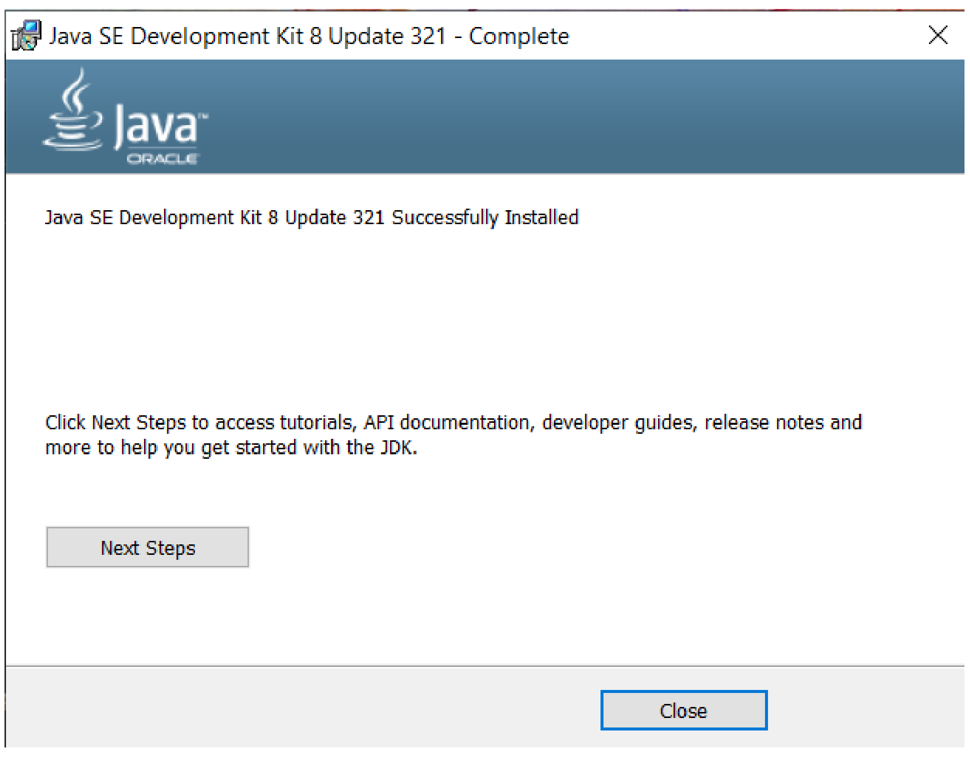
Destination Folder

Click "Change" to install Java to a different folder.

Install to:
C:\Program Files (x86)\Java\jre1.8.0_321


Java Setup - Progress

Status: Installing Java


Java SE Development Kit 8 Update 321 - Complete

Java SE Development Kit 8 Update 321 Successfully Installed

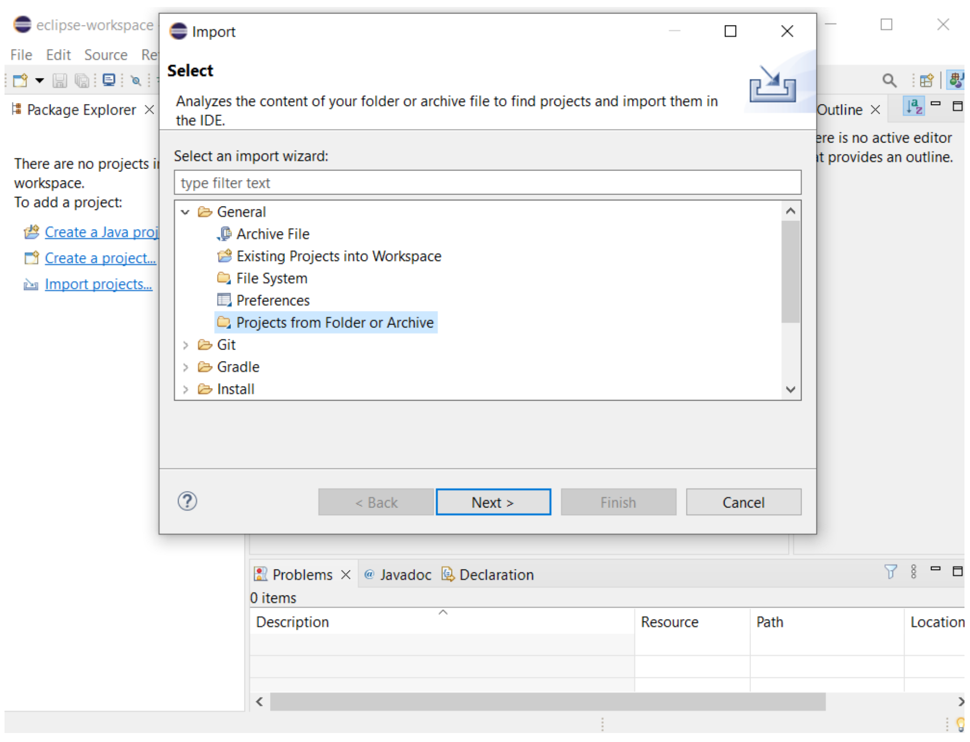
Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

5. Download the DIscoDNC Tool

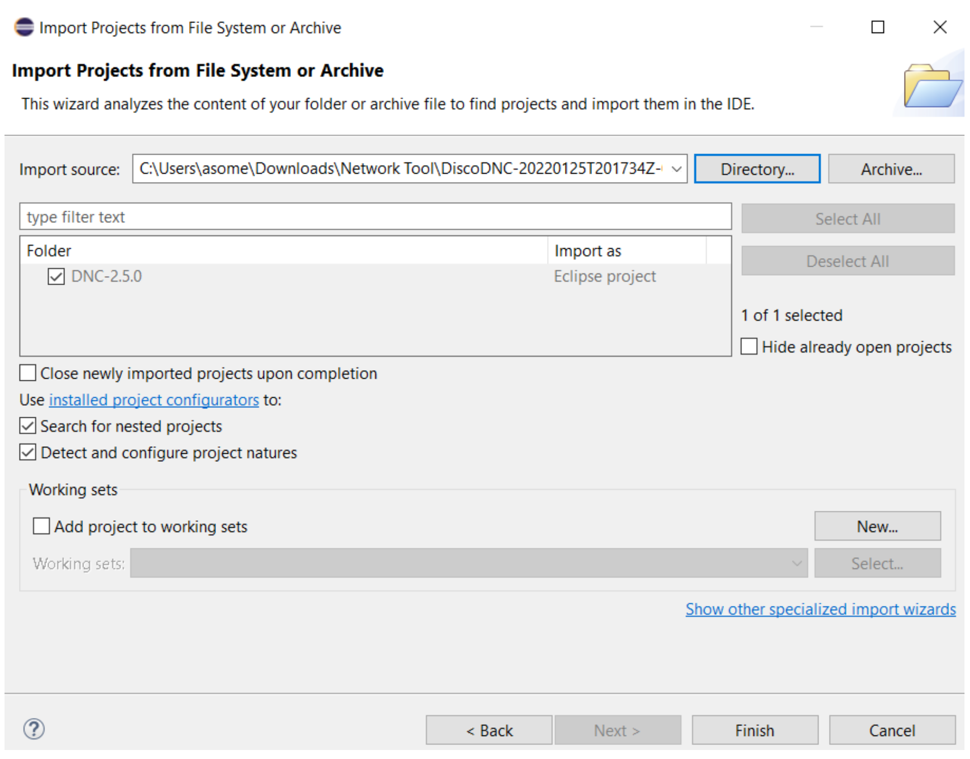
- *Download DiscoDNC from the portal link given below.*
- *Link:*
- *Move the Disco DNC to the working directory of your project.*

6. Import the DiscoDNC folder into the Eclipse

- *Inside Eclipse Workspace. Go to **File**, then select **Import**.*
- *Then go to **General** folder under **Select an import Wizard** and select **Projects from Folder or Archive** option.*



- *Then choose the directory where DiscoDNC folder is located and import it.*


Import Projects from File System or Archive
Import Projects from File System or Archive
This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.
Import source:
C:\Users\asome\Downloads\Network Tool\DiscoDNC-20220125T201734Z-
Directory...
Archive...
type filter text
Select All
Deselect All
Folder
Import as
DNC-2.5.0
Eclipse project
1 of 1 selected
Hide already open projects
Close newly imported projects upon completion
Use installed project configurators to:
Search for nested projects
Detect and configure project natures
Working sets
Add project to working sets
New...
Working sets:
Select...
Show other specialized import wizards
< Back
Next >
Finish
Cancel

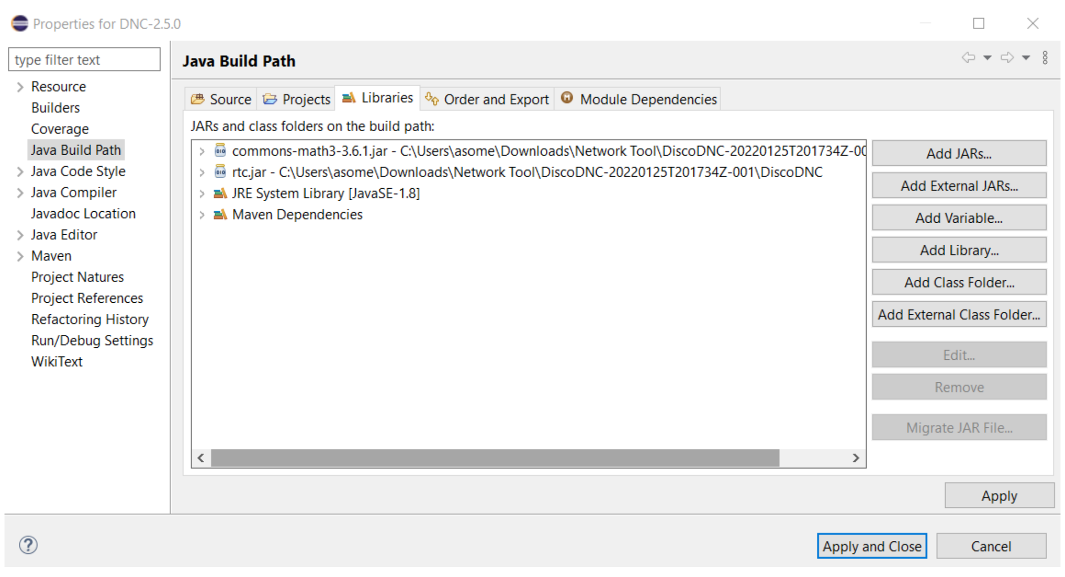
7. Import Jar and Java 8.

We need Java 8 and some external libraries(jar) file for DiscoDNC to work. We start by importing them into our project workspace.

- *Download* ***Common-maths3-3.6.1.jar*** *and* ***rtc.jar*** *from the portal and move inside project folder.*

  Link:

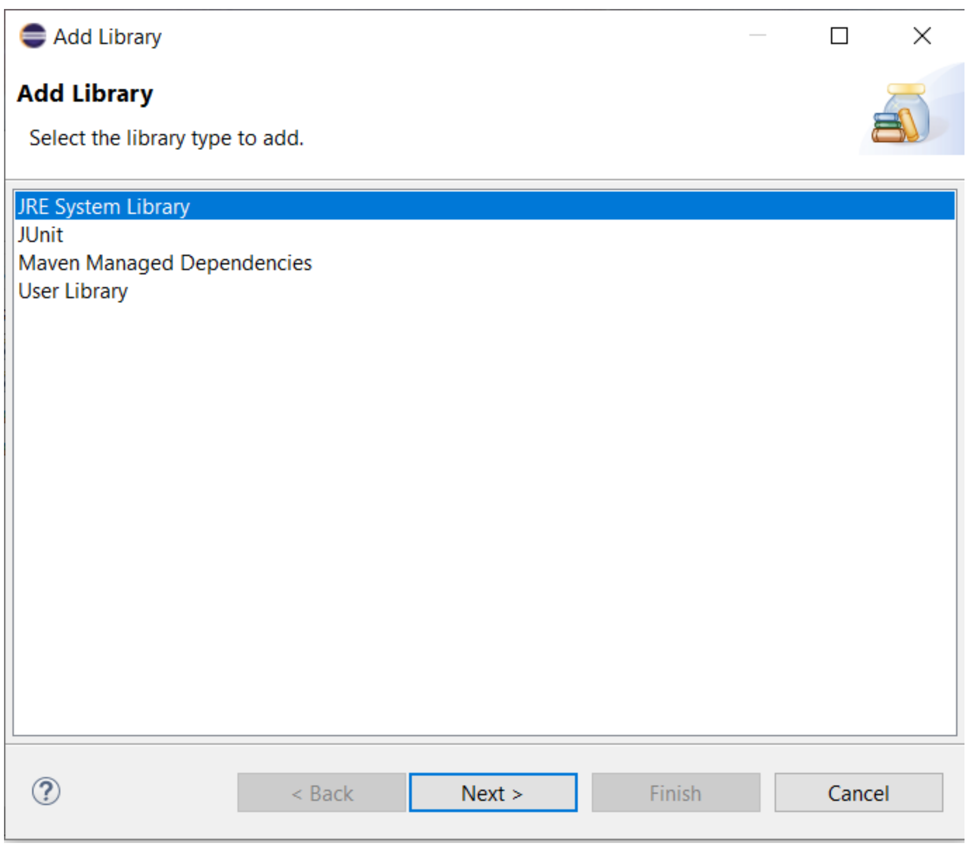
- *Right click on the project folder, Select* ***Build Path*** *and then* ***Configure Build Path****.*
- *You will be redirected into a new window.* Go to **Libraries** section from the options in the navbar.



- *Select* ***Add External Jars*** *option from the right side of the window.*
- *Select the* ***Common-maths3-3.6.1.jar*** *and* ***rtc.jar*** *from the folder where it was downloaded and add it.*

Now to add Java 8.

- *Select* ***Add Library*** *option from the right side of window.*
- Then select **JRE System Library** from the next window.

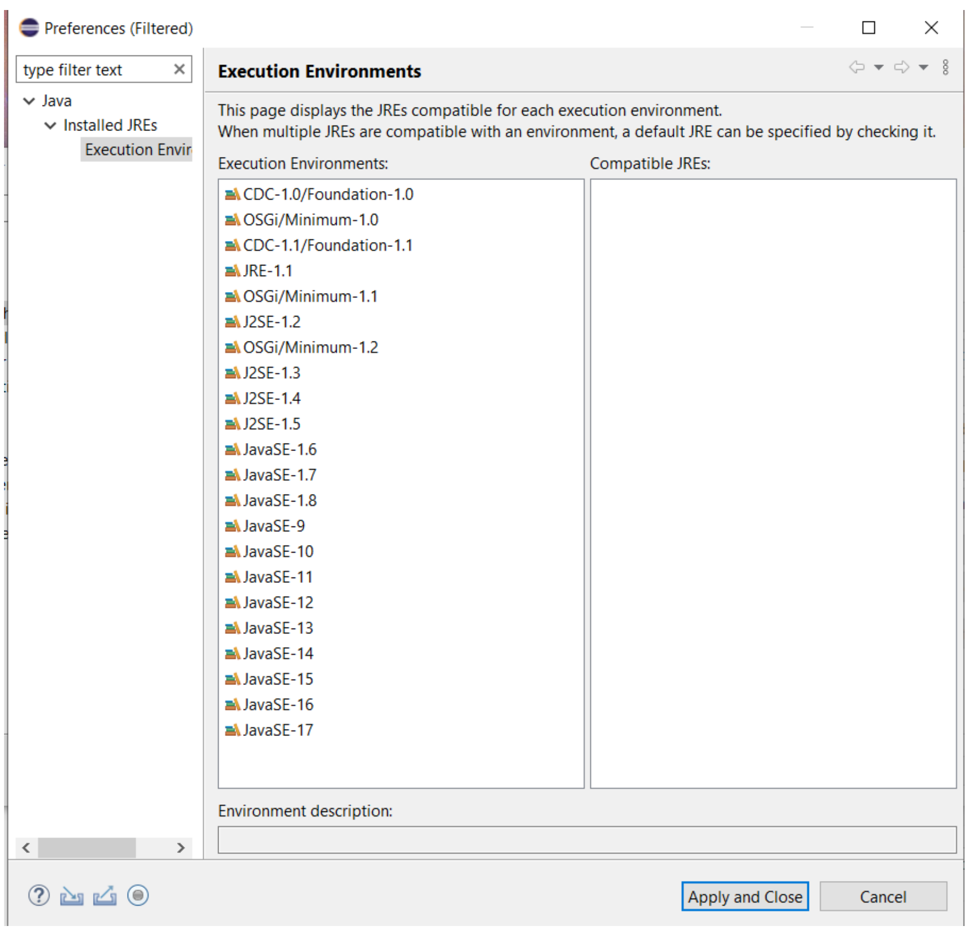

Add Library

Add Library

Select the library type to add.

JRE System Library
JUnit
Maven Managed Dependencies
User Library

- *After clicking **Next**, select **Java** and then select **Installed JREs** from the next window.*
- *Then select **Execution Environments** option.*


Preferences (Filtered)

type filter text
Java
Installed JREs
Execution Envir

Execution Environments

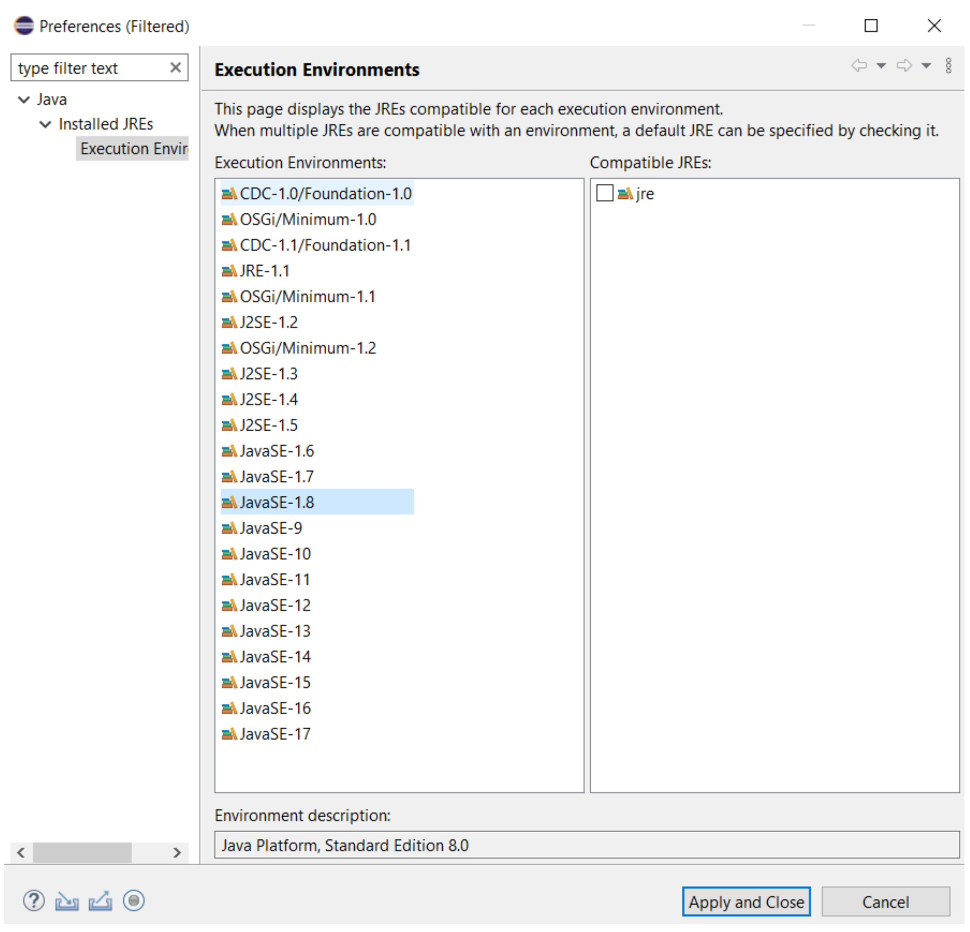
This page displays the JREs compatible for each execution environment.
When multiple JREs are compatible with an environment, a default JRE can be specified by checking it.

Execution Environments:
CDC-1.0/Foundation-1.0
OSGi/Minimum-1.0
CDC-1.1/Foundation-1.1
JRE-1.1
OSGi/Minimum-1.1
J2SE-1.2
OSGi/Minimum-1.2
J2SE-1.3
J2SE-1.4
J2SE-1.5
JavaSE-1.6
JavaSE-1.7
JavaSE-1.8
JavaSE-9
JavaSE-10
JavaSE-11
JavaSE-12
JavaSE-13
JavaSE-14
JavaSE-15
JavaSE-16
JavaSE-17

Compatible JREs:

Environment description:

- *Then select **JavaSE-1.8** and click Apply and Close button.*


Preferences (Filtered)

type filter text
Java
Installed JREs
Execution Envir

Execution Environments

This page displays the JREs compatible for each execution environment.
When multiple JREs are compatible with an environment, a default JRE can be specified by checking it.

Execution Environments:
CDC-1.0/Foundation-1.0
OSGi/Minimum-1.0
CDC-1.1/Foundation-1.1
JRE-1.1
OSGi/Minimum-1.1
J2SE-1.2
OSGi/Minimum-1.2
J2SE-1.3
J2SE-1.4
J2SE-1.5
JavaSE-1.6
JavaSE-1.7
JavaSE-1.8
JavaSE-9
JavaSE-10
JavaSE-11
JavaSE-12
JavaSE-13
JavaSE-14
JavaSE-15
JavaSE-16
JavaSE-17

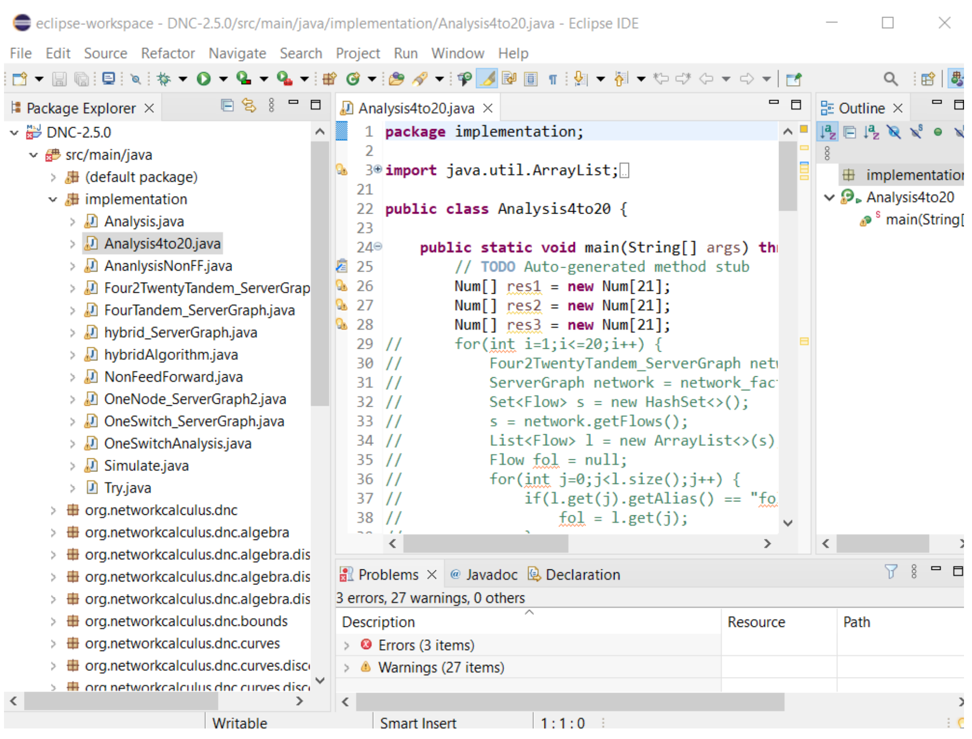
Compatible JREs:
☐ jre

Environment description:
Java Platform, Standard Edition 8.0

- *Click on Finish and **Apply and Close** Button*

8. Running the Program.

- *Inside **DNC-2.5.0**, go to **src/main/java***
- *Then inside **src/main/java**, go to **implementation** folder.*
- *Then inside **implementation**, go to **Analysis4to20.java***



eclipse-workspace - DNC-2.5.0/src/main/java/implementation/Analysis4to20.java - Eclipse IDE

Package Explorer
- DNC-2.5.0
  - src/main/java
    - (default package)
    - implementation
      - Analysis.java
      - Analysis4to20.java
      - AnanlysisNonFF.java
      - Four2TwentyTandem_ServerGrap
      - FourTandem_ServerGraph.java
      - hybrid_ServerGraph.java
      - hybridAlgorithm.java
      - NonFeedForward.java
      - OneNode_ServerGraph2.java
      - OneSwitch_ServerGraph.java
      - OneSwitchAnalysis.java
      - Simulate.java
      - Try.java
    - org.networkcalculus.dnc
    - org.networkcalculus.dnc.algebra
    - org.networkcalculus.dnc.algebra.dis
    - org.networkcalculus.dnc.algebra.dis
    - org.networkcalculus.dnc.algebra.dis
    - org.networkcalculus.dnc.bounds
    - org.networkcalculus.dnc.curves
    - org.networkcalculus.dnc.curves.disc

Analysis4to20.java

```
package implementation;

import java.util.ArrayList;

public class Analysis4to20 {

    public static void main(String[] args) thi
        // TODO Auto-generated method stub
        Num[] res1 = new Num[21];
        Num[] res2 = new Num[21];
        Num[] res3 = new Num[21];
//      for(int i=1;i<=20;i++) {
//          Four2TwentyTandem_ServerGraph netw
//          ServerGraph network = network_fac
//          Set<Flow> s = new HashSet<>();
//          s = network.getFlows();
//          List<Flow> l = new ArrayList<>(s)
//          Flow fol = null;
//          for(int j=0;j<l.size();j++) {
//              if(l.get(j).getAlias() == "fo
//                  fol = l.get(j);
```

Outline: implementation, Analysis4to20, main(String[

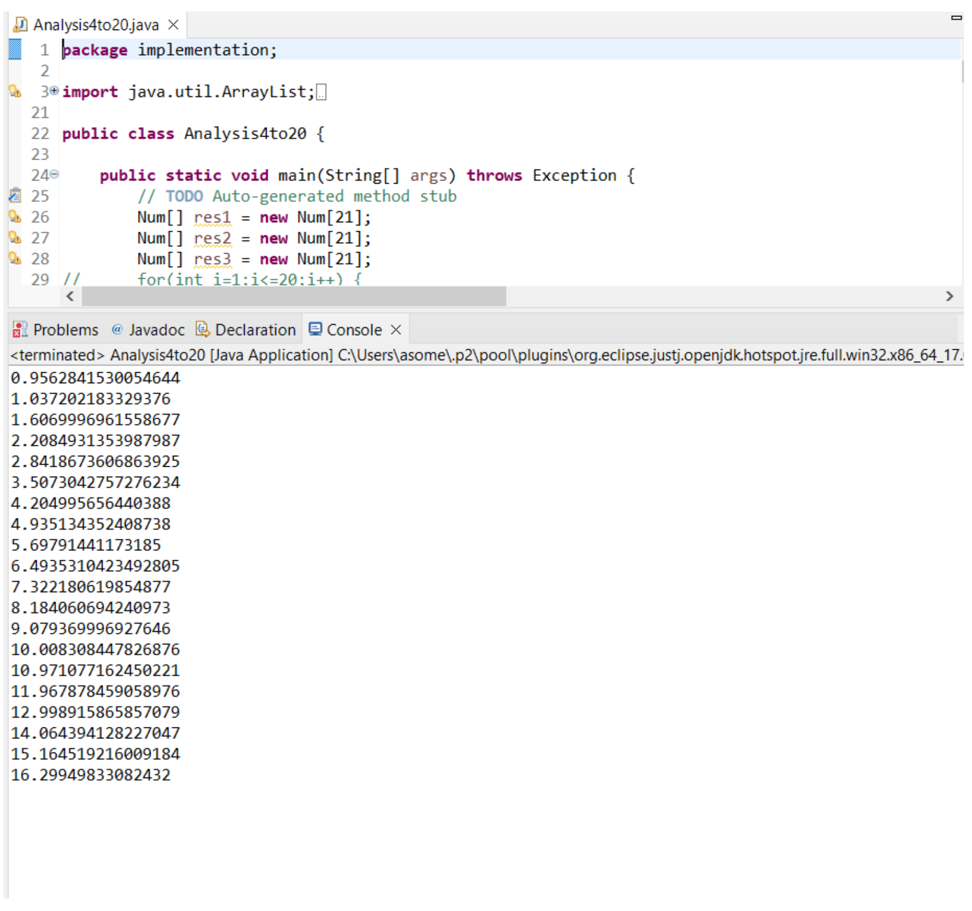
Problems | Javadoc | Declaration

3 errors, 27 warnings, 0 others

Description	Resource	Path
Errors (3 items)		
Warnings (27 items)		

Writable | Smart Insert | 1 : 1 : 0

- Then run that file and record the result.



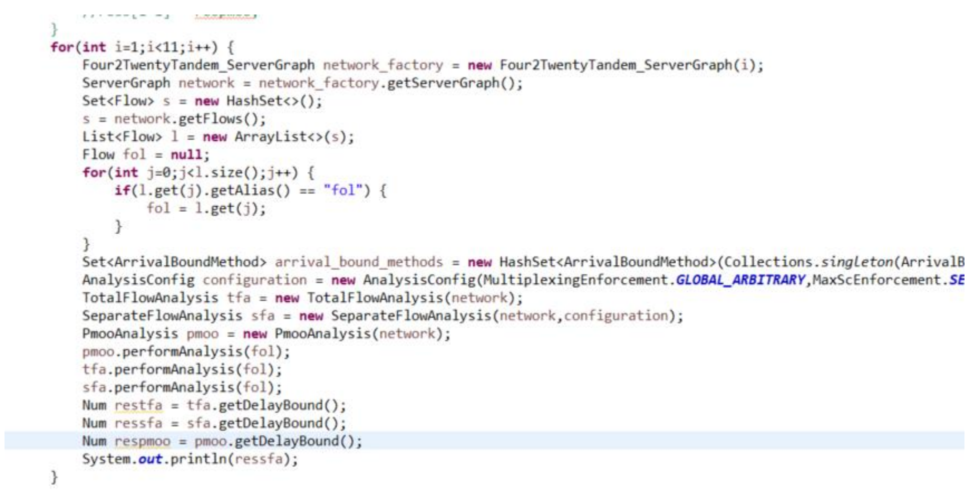
Analysis4to20.java

```
package implementation;

import java.util.ArrayList;

public class Analysis4to20 {

    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub
        Num[] res1 = new Num[21];
        Num[] res2 = new Num[21];
        Num[] res3 = new Num[21];
//      for(int i=1;i<=20;i++) {
```

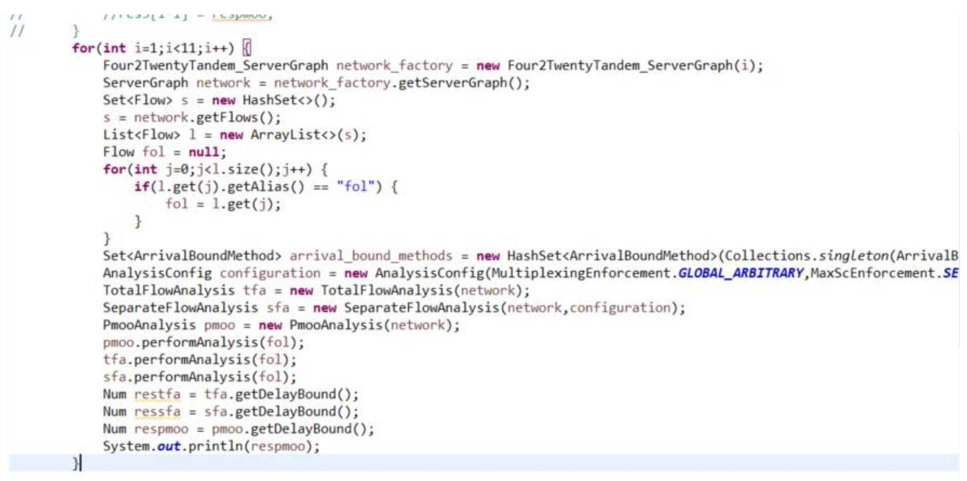
Problems | Javadoc | Declaration | Console

```
<terminated> Analysis4to20 [Java Application] C:\Users\asome\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.
0.9562841530054644
1.037202183329376
1.6069996961558677
2.2084931353987987
2.8418673606863925
3.5073042757276234
4.204995656440388
4.935134352408738
5.69791441173185
6.4935310423492805
7.322180619854877
8.184060694240973
9.079369996927646
10.008308447826876
10.971077162450221
11.967878459058976
12.998915865857079
14.064394128227047
15.164519216009184
16.29949833082432
```

- Inside ***Analysis4to20.java*** *use* ***System.out.println(ressfa)*** *and* ***System.out.println(respmoo)*** *to get values of delay for* ***SFA FIFO*** *and* ***PMOO*** *respectively.*

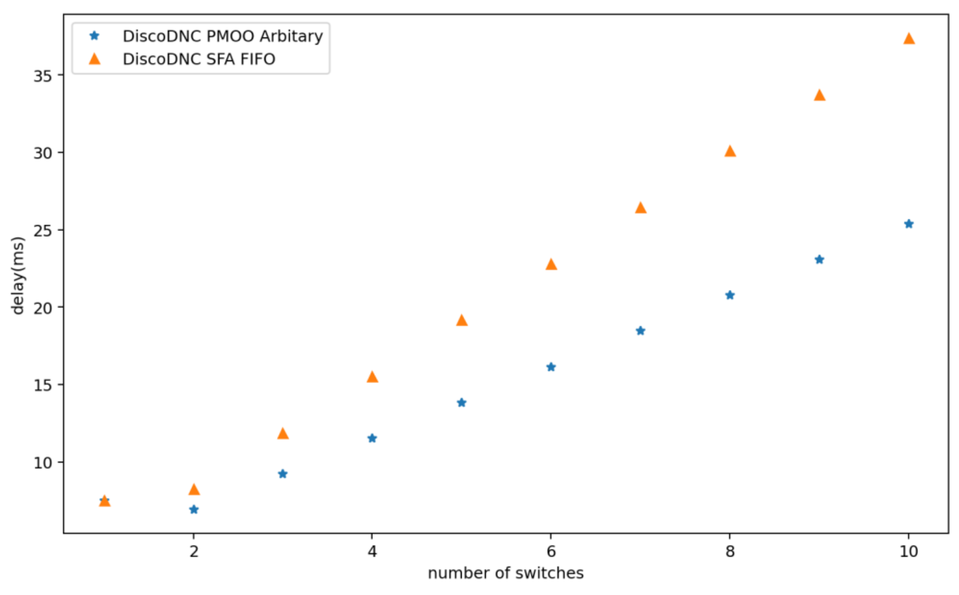


```
    }
    for(int i=1;i<11;i++) {
        Four2TwentyTandem_ServerGraph network_factory = new Four2TwentyTandem_ServerGraph(i);
        ServerGraph network = network_factory.getServerGraph();
        Set<Flow> s = new HashSet<>();
        s = network.getFlows();
        List<Flow> l = new ArrayList<>(s);
        Flow fol = null;
        for(int j=0;j<l.size();j++) {
            if(l.get(j).getAlias() == "fol") {
                fol = l.get(j);
            }
        }
        Set<ArrivalBoundMethod> arrival_bound_methods = new HashSet<ArrivalBoundMethod>(Collections.singleton(ArrivalB
        AnalysisConfig configuration = new AnalysisConfig(MultiplexingEnforcement.GLOBAL_ARBITRARY,MaxScEnforcement.SE
        TotalFlowAnalysis tfa = new TotalFlowAnalysis(network);
        SeparateFlowAnalysis sfa = new SeparateFlowAnalysis(network,configuration);
        PmooAnalysis pmoo = new PmooAnalysis(network);
        pmoo.performAnalysis(fol);
        tfa.performAnalysis(fol);
        sfa.performAnalysis(fol);
        Num restfa = tfa.getDelayBound();
        Num ressfa = sfa.getDelayBound();
        Num respmoo = pmoo.getDelayBound();
        System.out.println(ressfa);
    }
```



```
//      }
    for(int i=1;i<11;i++) {
        Four2TwentyTandem_ServerGraph network_factory = new Four2TwentyTandem_ServerGraph(i);
        ServerGraph network = network_factory.getServerGraph();
        Set<Flow> s = new HashSet<>();
        s = network.getFlows();
        List<Flow> l = new ArrayList<>(s);
        Flow fol = null;
        for(int j=0;j<l.size();j++) {
            if(l.get(j).getAlias() == "fol") {
                fol = l.get(j);
            }
        }
        Set<ArrivalBoundMethod> arrival_bound_methods = new HashSet<ArrivalBoundMethod>(Collections.singleton(ArrivalB
        AnalysisConfig configuration = new AnalysisConfig(MultiplexingEnforcement.GLOBAL_ARBITRARY,MaxScEnforcement.SE
        TotalFlowAnalysis tfa = new TotalFlowAnalysis(network);
        SeparateFlowAnalysis sfa = new SeparateFlowAnalysis(network,configuration);
        PmooAnalysis pmoo = new PmooAnalysis(network);
        pmoo.performAnalysis(fol);
        tfa.performAnalysis(fol);
        sfa.performAnalysis(fol);
        Num restfa = tfa.getDelayBound();
        Num ressfa = sfa.getDelayBound();
        Num respmoo = pmoo.getDelayBound();
        System.out.println(respmoo);
    }
```

- *Now plot the values and create a graph as follows.*


★ DiscoDNC PMOO Arbitary
▲ DiscoDNC SFA FIFO
delay(ms)
number of switches