Execute, Verify & Submit – Java Programs

Author(s)	Seetha Lakshmi
Authorized by	Head, ETA
Creation/Revision Date	Mar 2021
Version	1.1

Infosys® | Building Tomorrow's Enterprise

Usage Guidelines

Do not forward this document to any non-Infosys mail ID. Forwarding this document to a non-Infosys mail ID may lead to disciplinary action against you, including termination of employment.

Contents of this material cannot be used in any other internal or external document without explicit permission from ETA@infosys.com.

COPYRIGHT NOTICE

Education, Training and Assessment Department
Infosys Limited
Electronics City
Hosur Road
Bangalore – 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infosys.com
mailto:ETA@infosys.com

Document Revision History

Version	Date	Author(s)	Reviewer(s)	Description
1.0	Mar 2021	Seetha Lakshmi	Malathi_M,Komal_Papdeja	First draft of document
1.1	Mar 2021	Seetha Lakshmi	Malathi_M,Komal_Papdeja	Added steps related to verify, submit and last submission

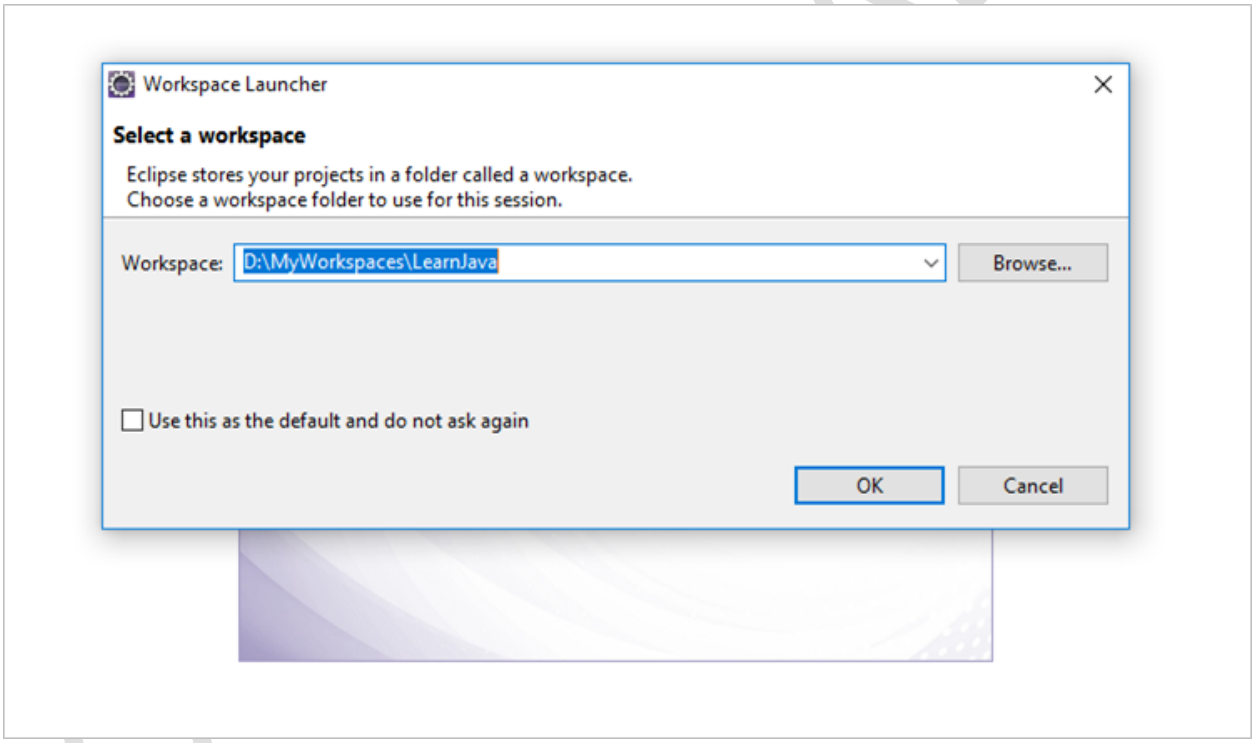
CONTENTS

Follow the below Steps to run the Java programs

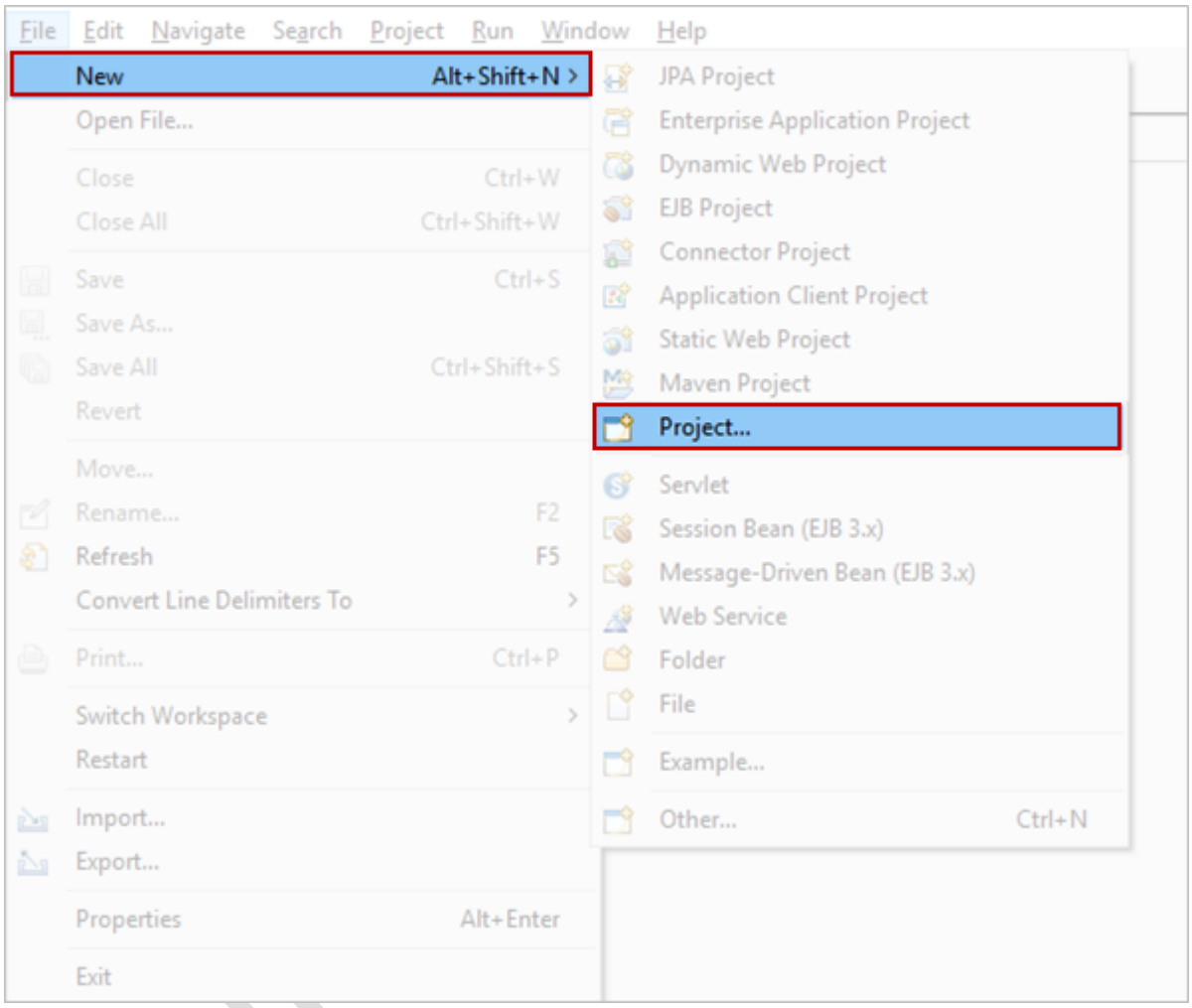
Step 1: For installing Java, click here (Topics Name - Adopt OpenJDK8)

Step 2: To configure Eclipse IDE, click here (Topic Name : Eclipse IDE)

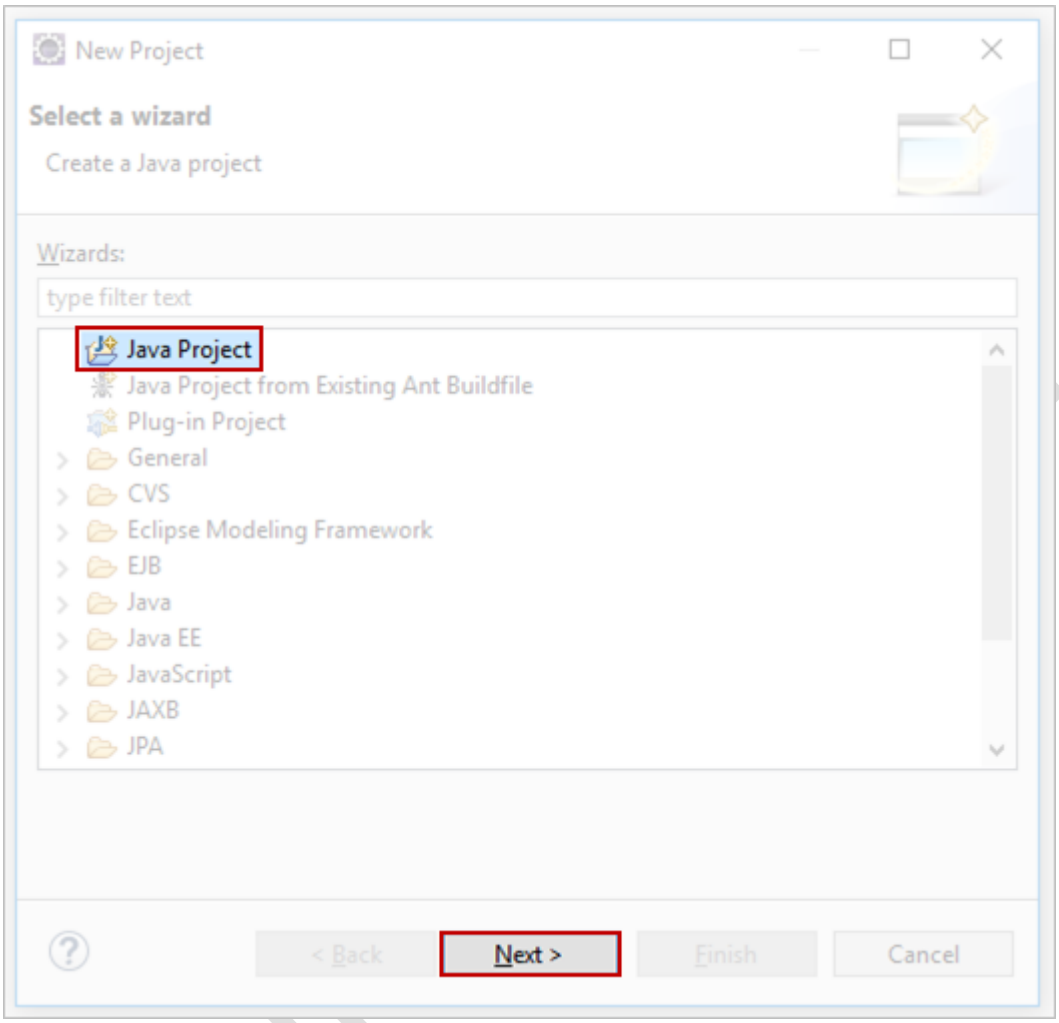
Step 3: Once eclipse is configured, launch Eclipse by clicking on "eclipse.exe" and create a workspace.



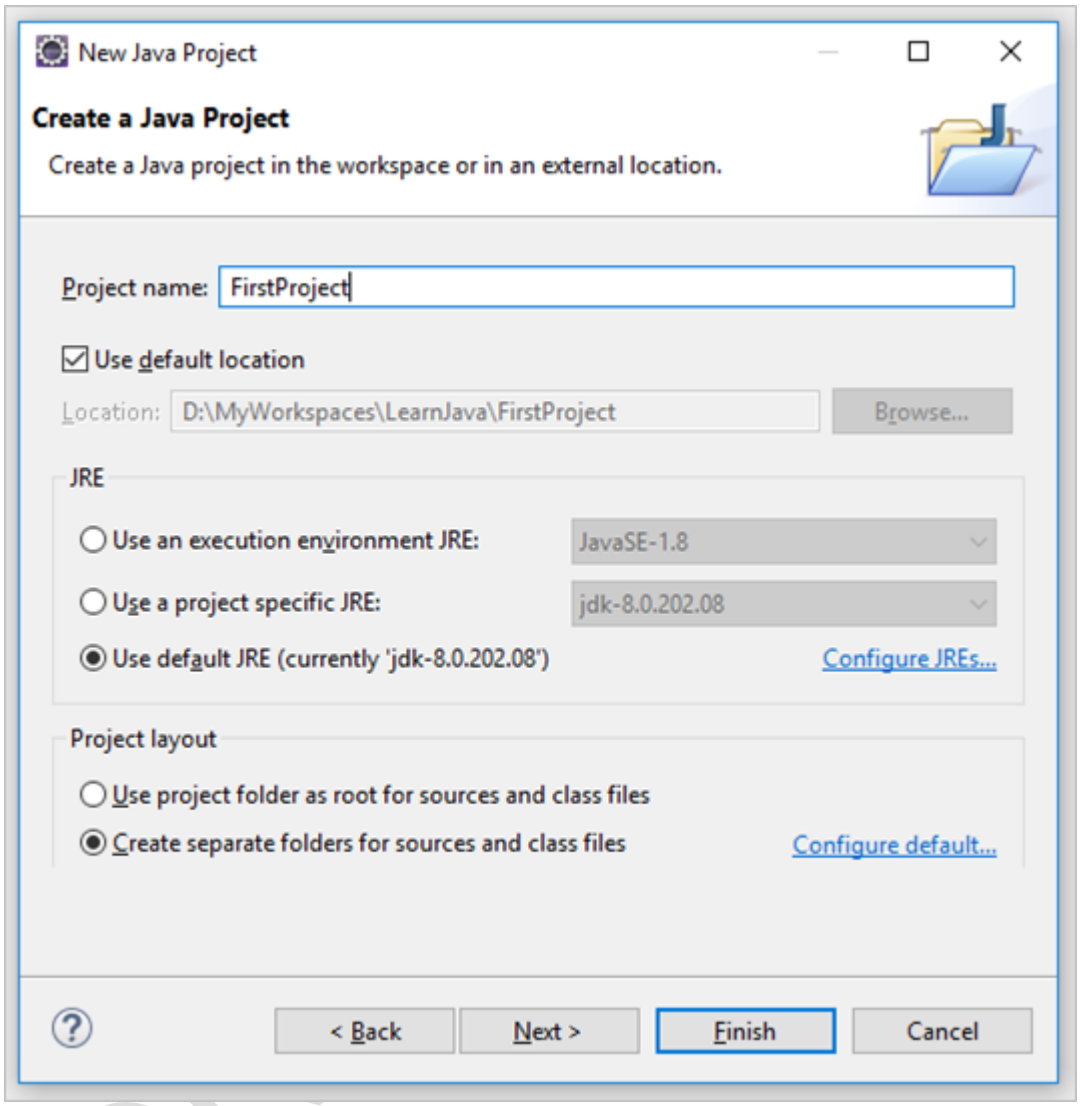
Step 4: Create a new Java project in Eclipse by clicking on menu options – **File → New →Project**



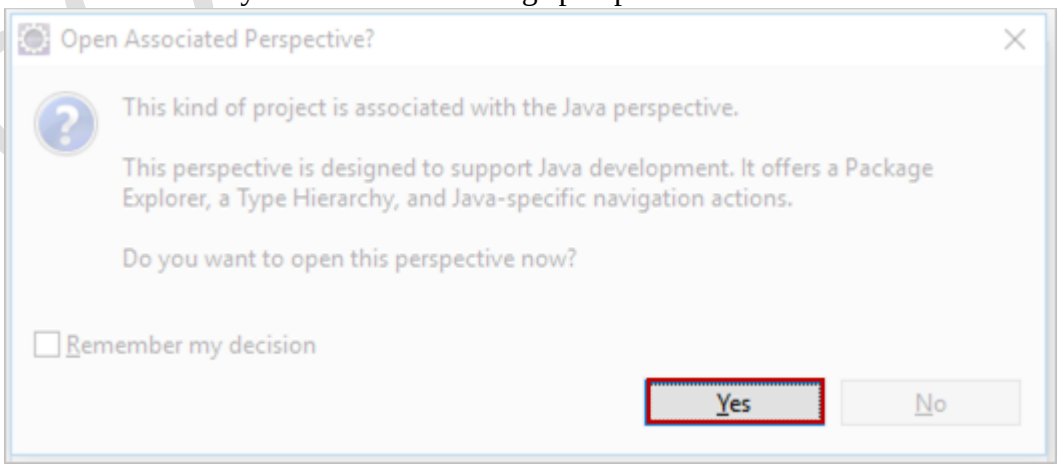
Step 5: Select **"Java Project"** and click on **"Next"**



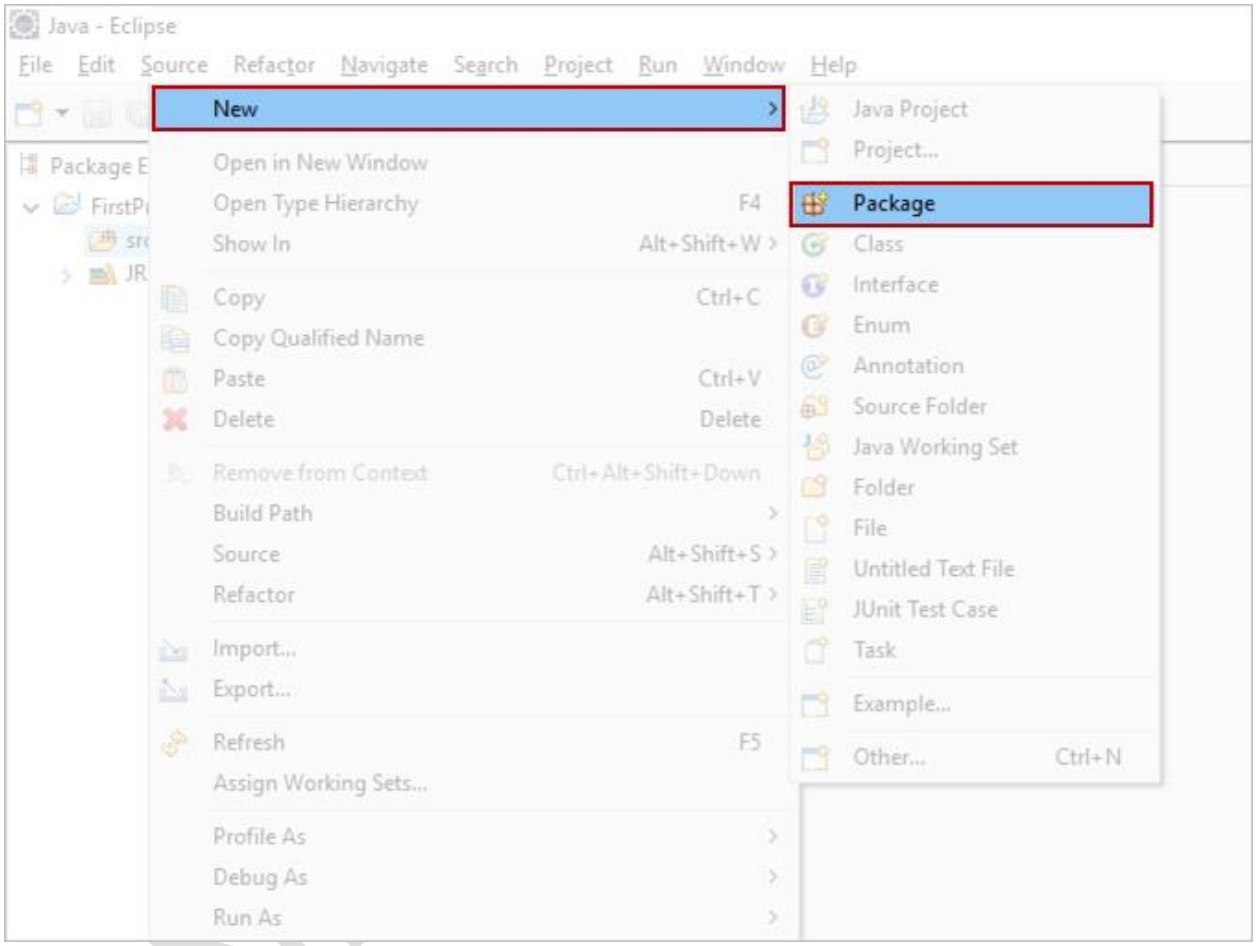
Step 6: Enter the **"Project name"**, use the default options for JRE and click on **"Finish"**.



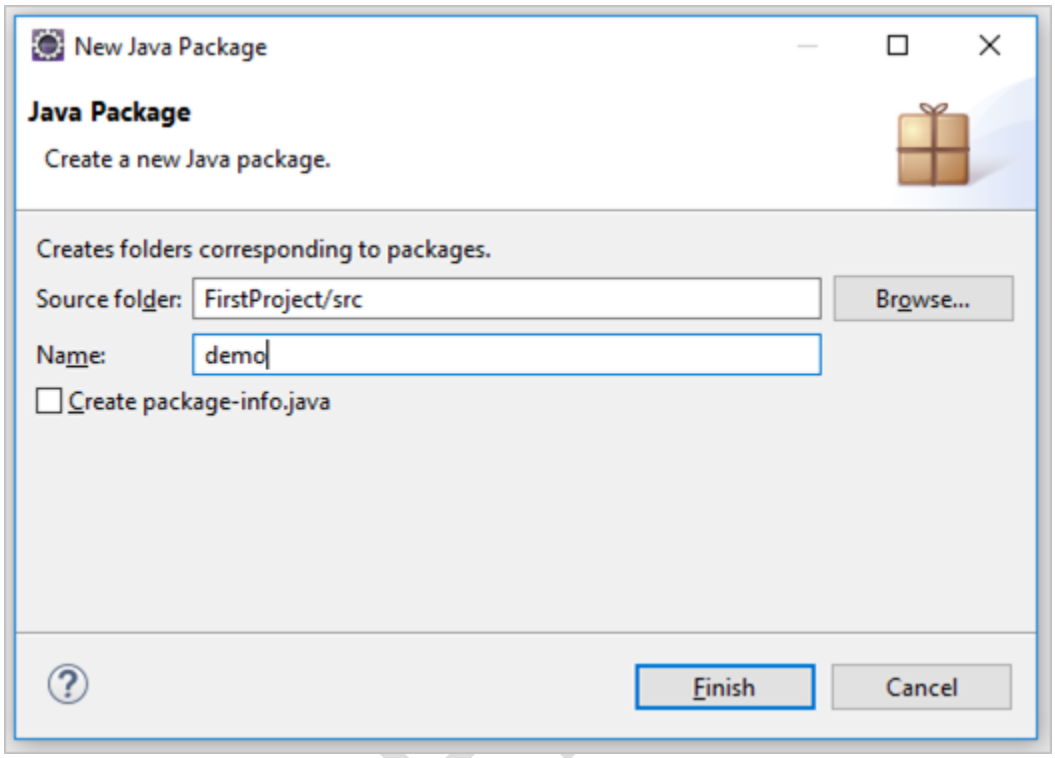
Click on "**Yes**" if you are asked to change perspective.



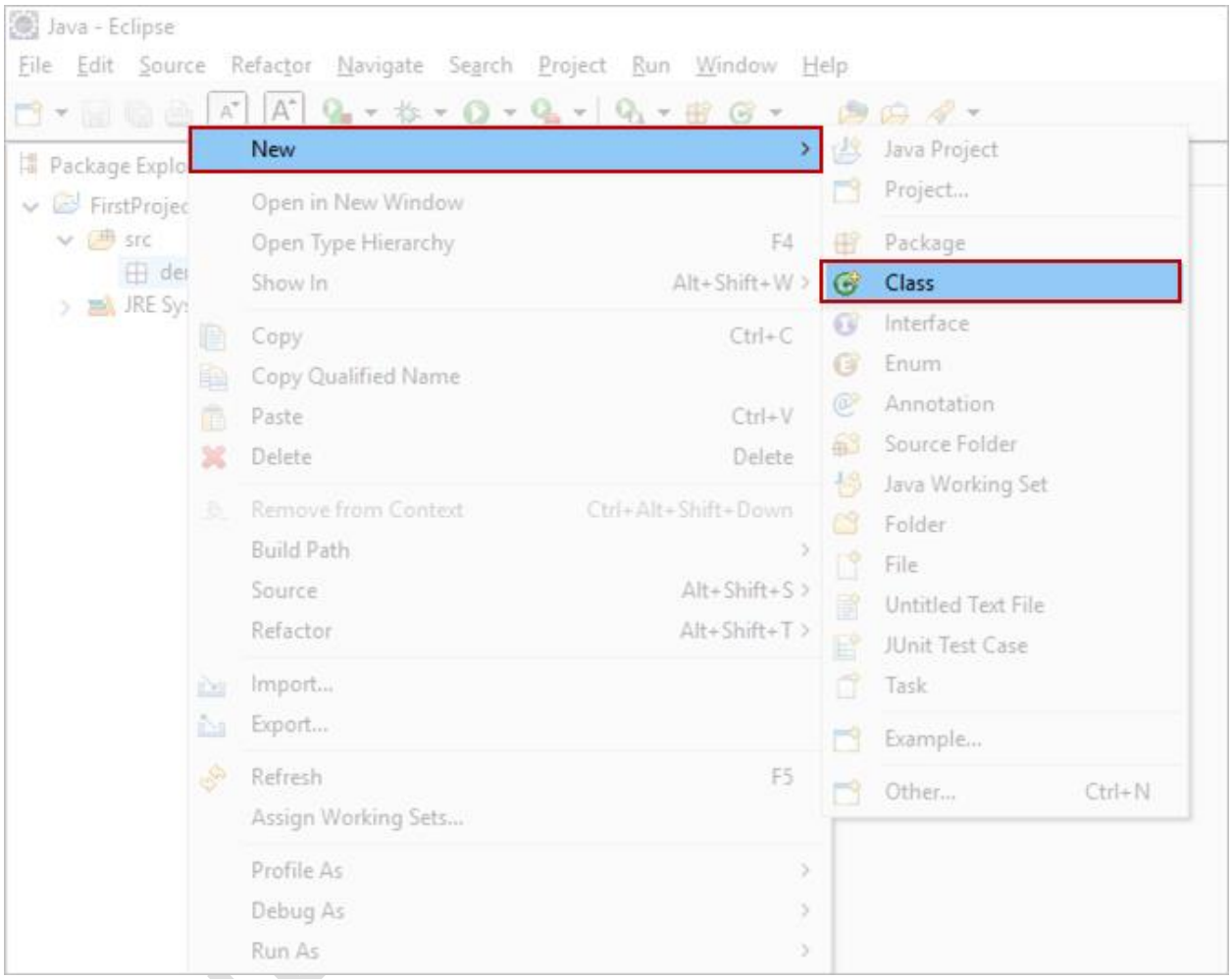
Step 7: Right-click on **"src"** and select **New -> Package**



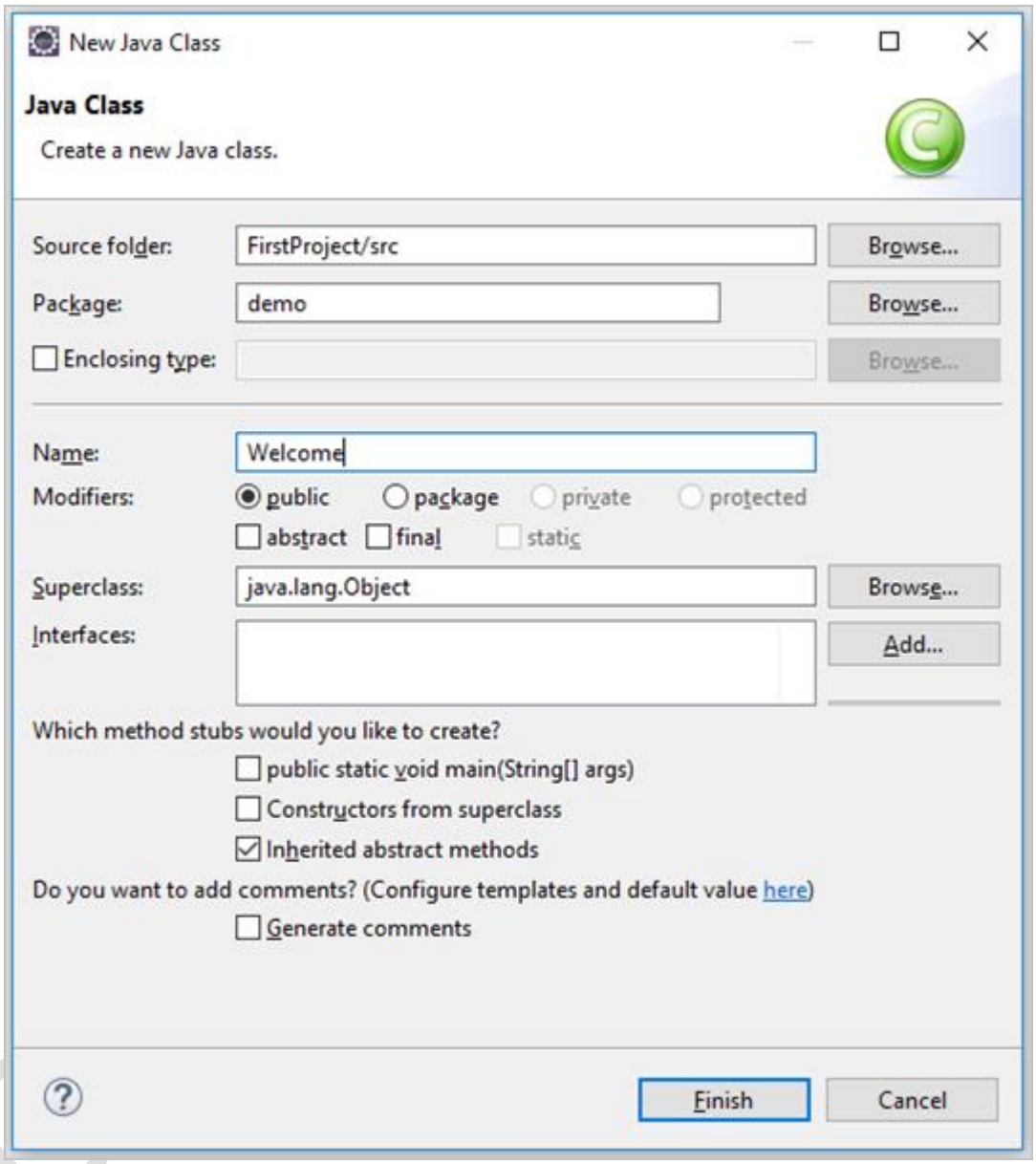
Step 8: Enter the **"Name"** of the package and click on **"Finish"**.



Step 9: Right-click on the package("demo"), select **New -> Class**. This will create a new .java file where you can write your program.

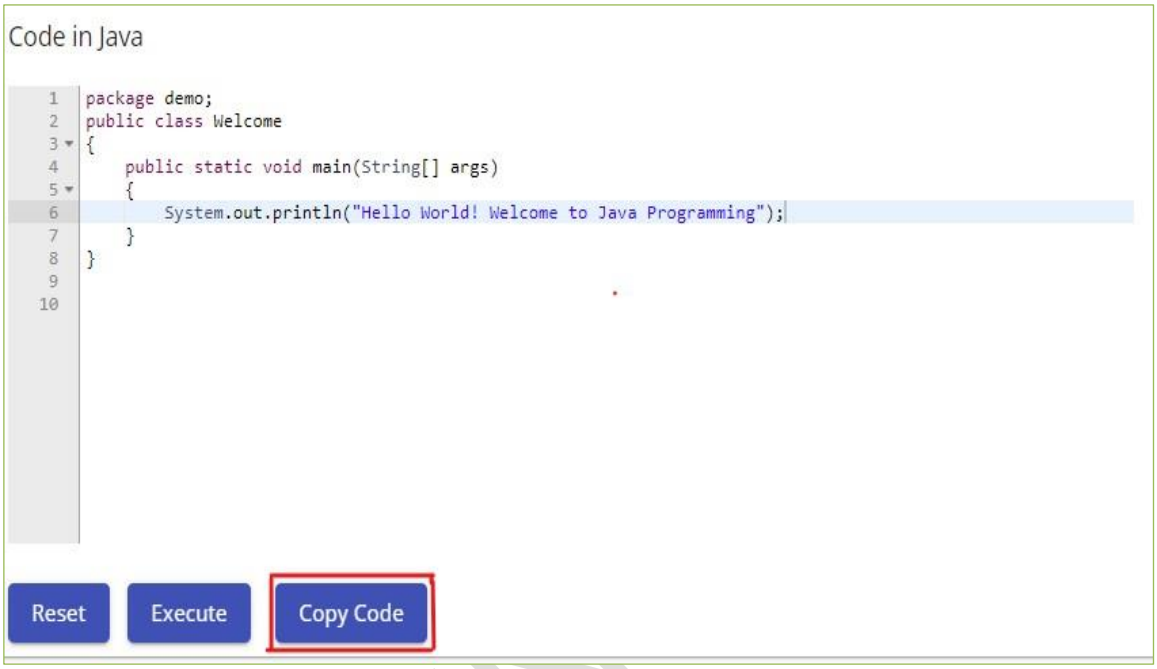


Step 10: Enter the name of the class as **"Welcome"** and click on **"Finish"**.



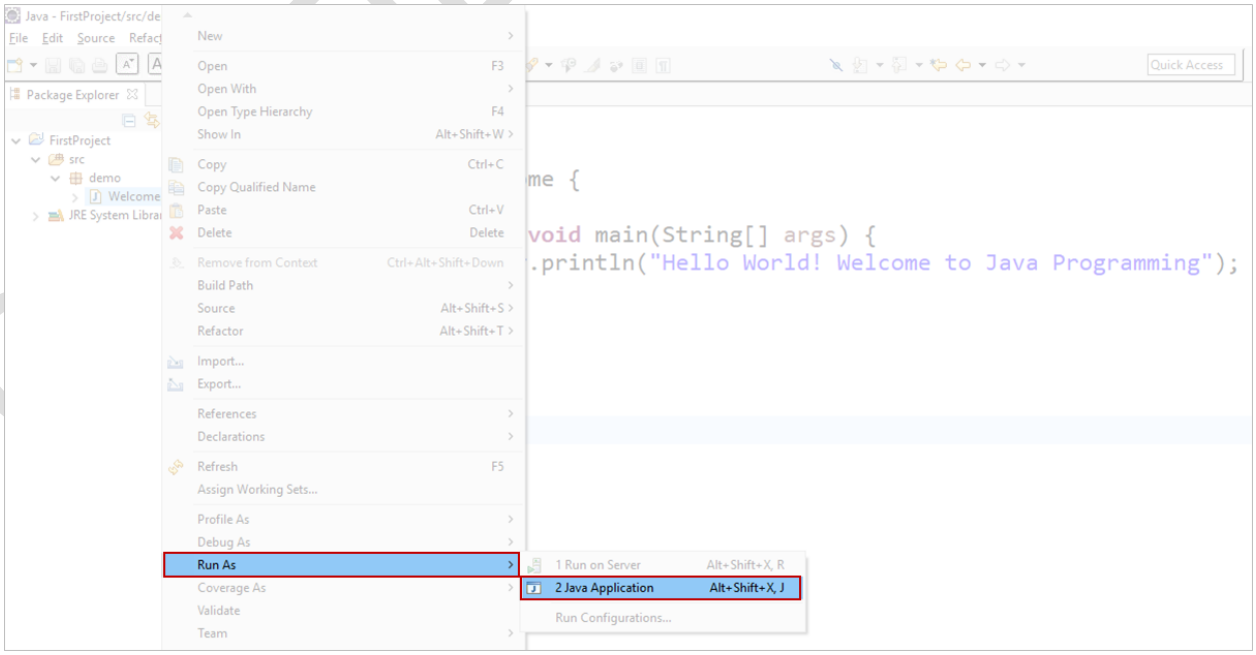
Step 11: The .java file will open in the editor.

Step 12: Copy the code from here

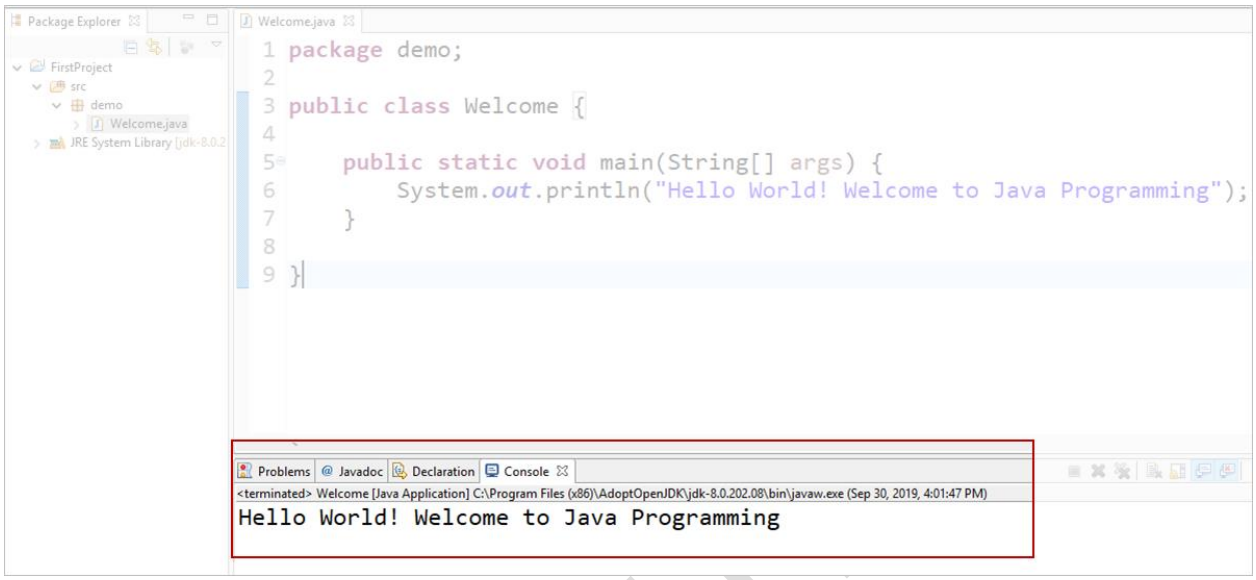


NOTE: By default, Eclipse will automatically compile the .java file into a .class file (byte code) when the file is saved.

Step 13: To execute the program, right-click on the .java file, select **Run As -> Java Application** or click button in the tool bar

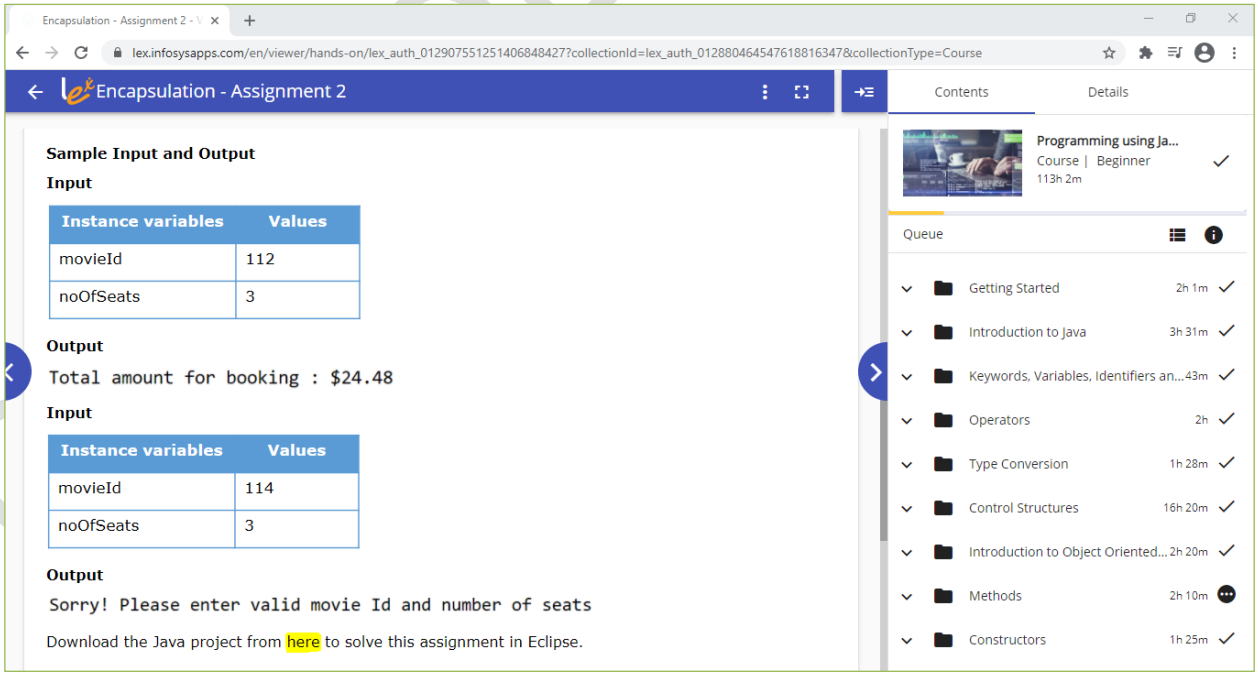


The output will be displayed in the Console window as shown below.

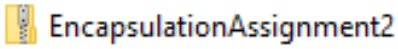


Let us now understand the steps to be followed to solve the assignments in Eclipse, execute and verify in Lex

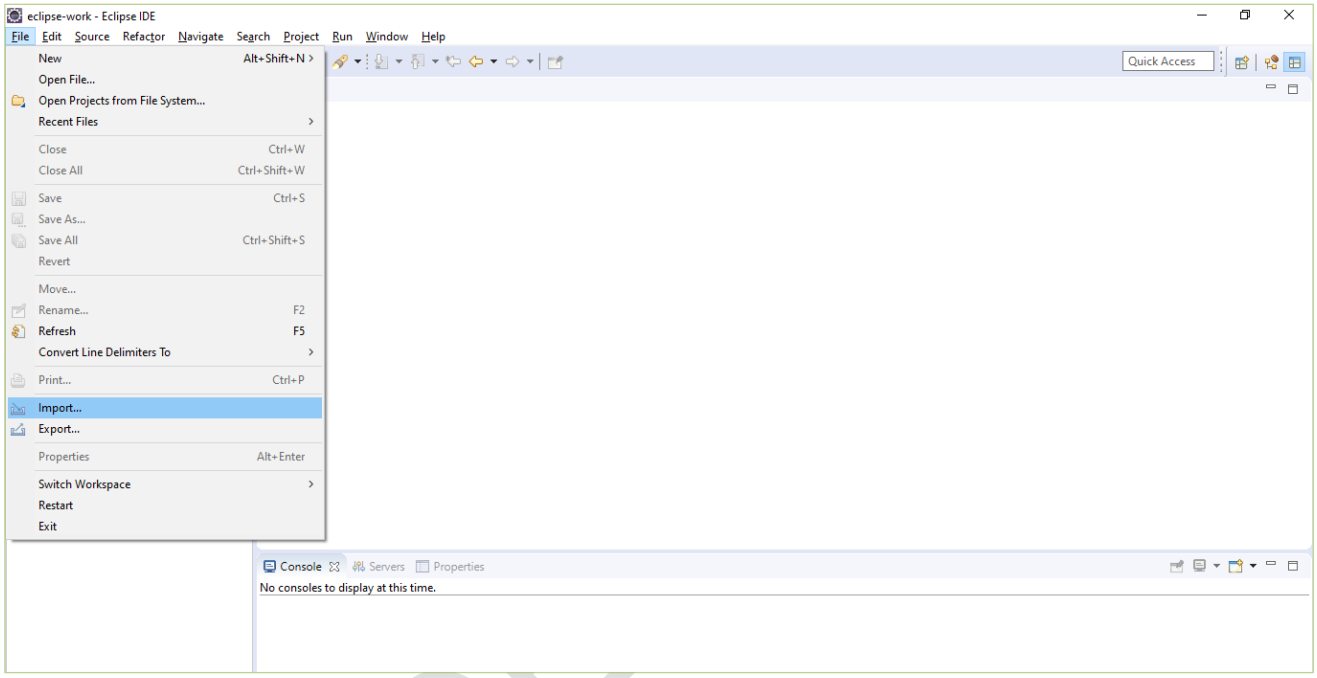
Step 14: Download the Java project artifacts for any of the assignments (Eg: Encapsulation – Assignment 2)

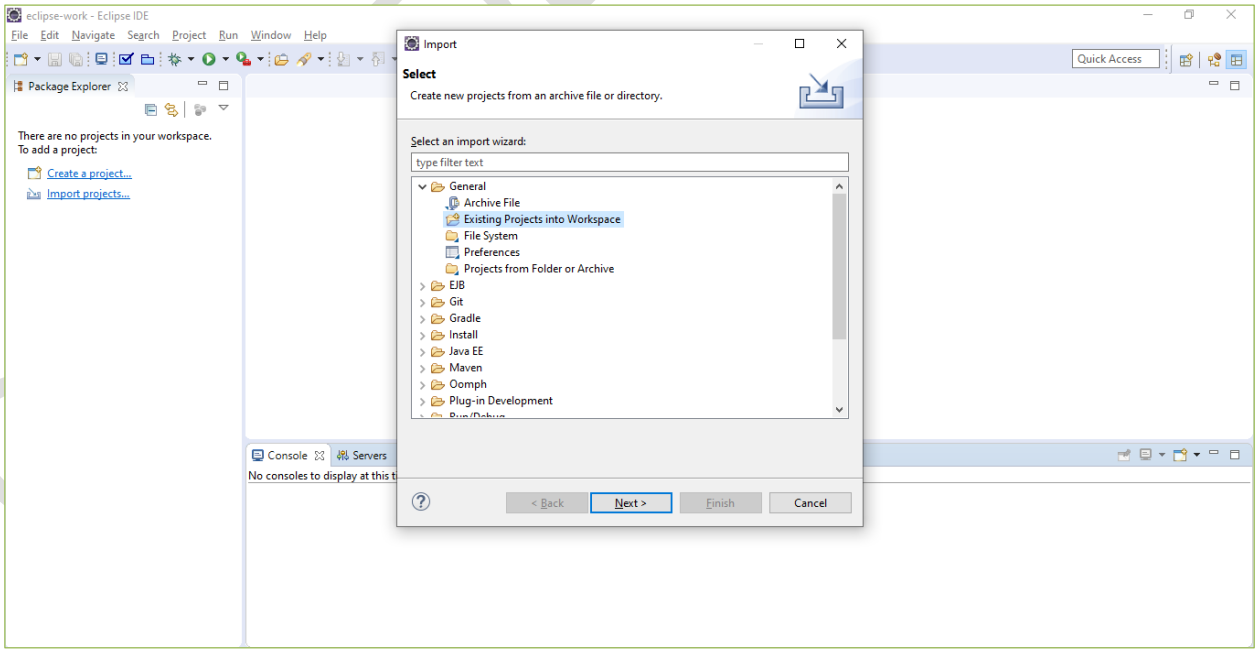


Step 15: Project artifacts as a zipped file will be downloaded under **"Downloads"** as below. Unzip the file.

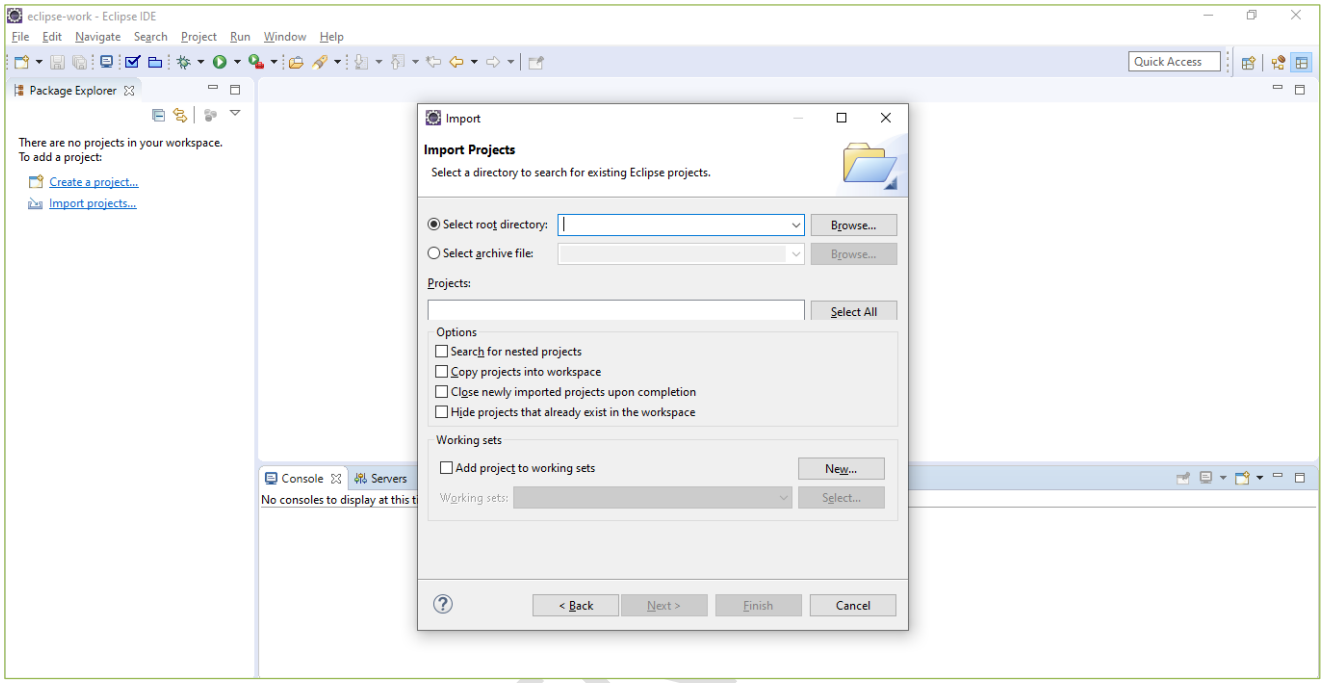


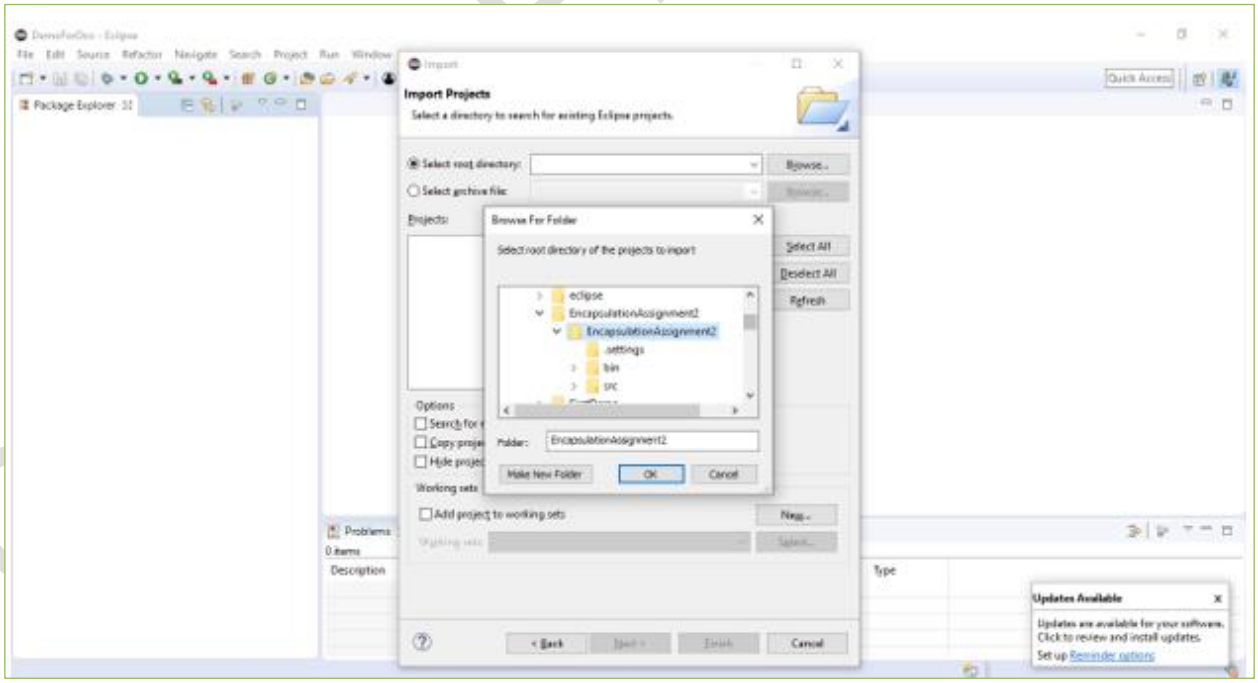
Step 16: In eclipse, Go to **File → Import**. Expand **"General"** and select **"Existing Projects into Workspace"**



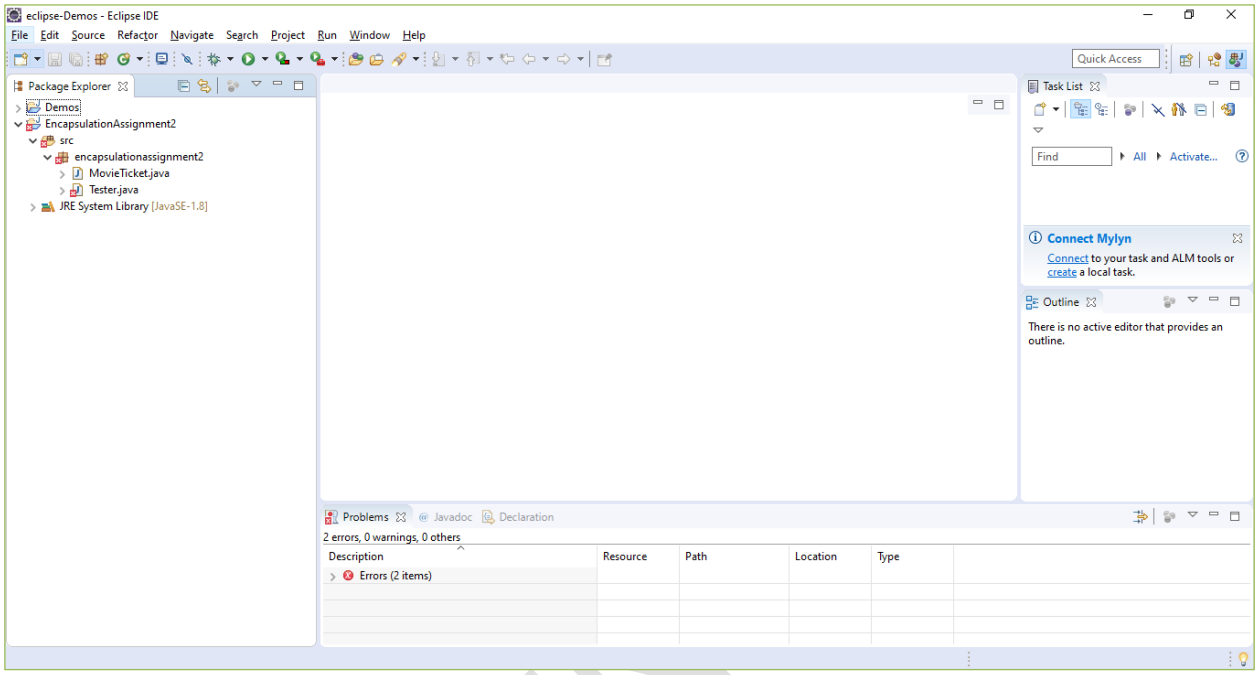


Step 17: Click on **"Browse"** and select the unzipped folder containing **".settings"** folder **ONLY**

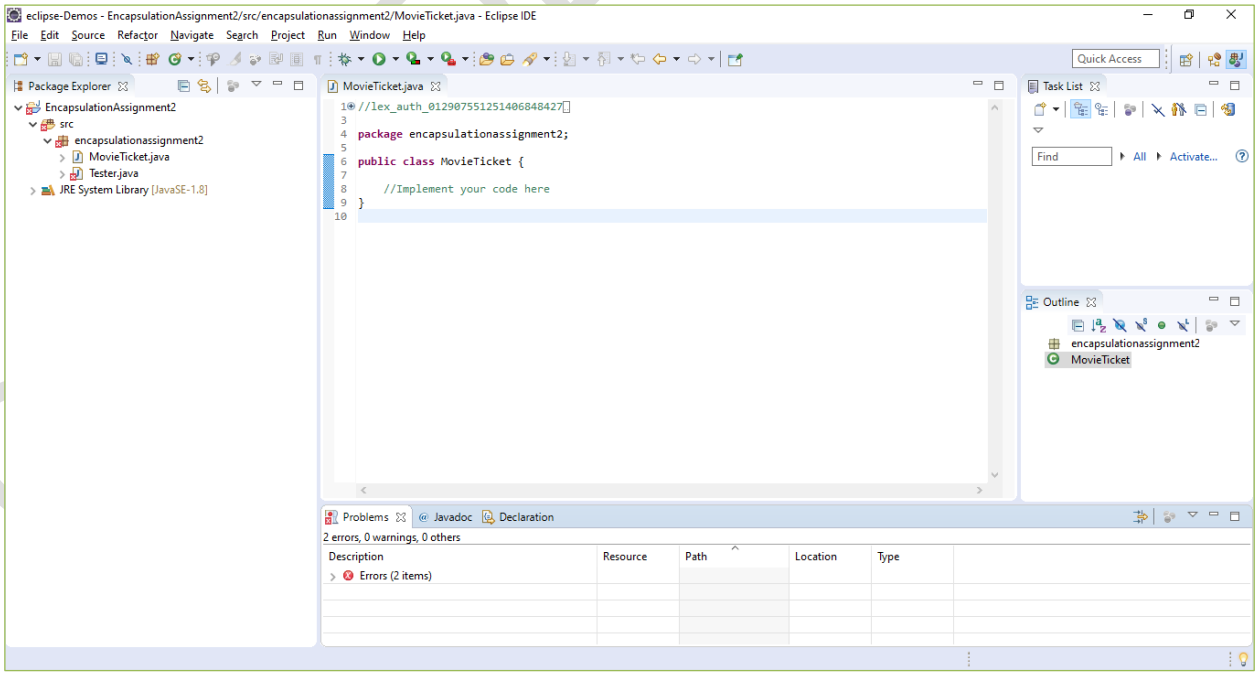




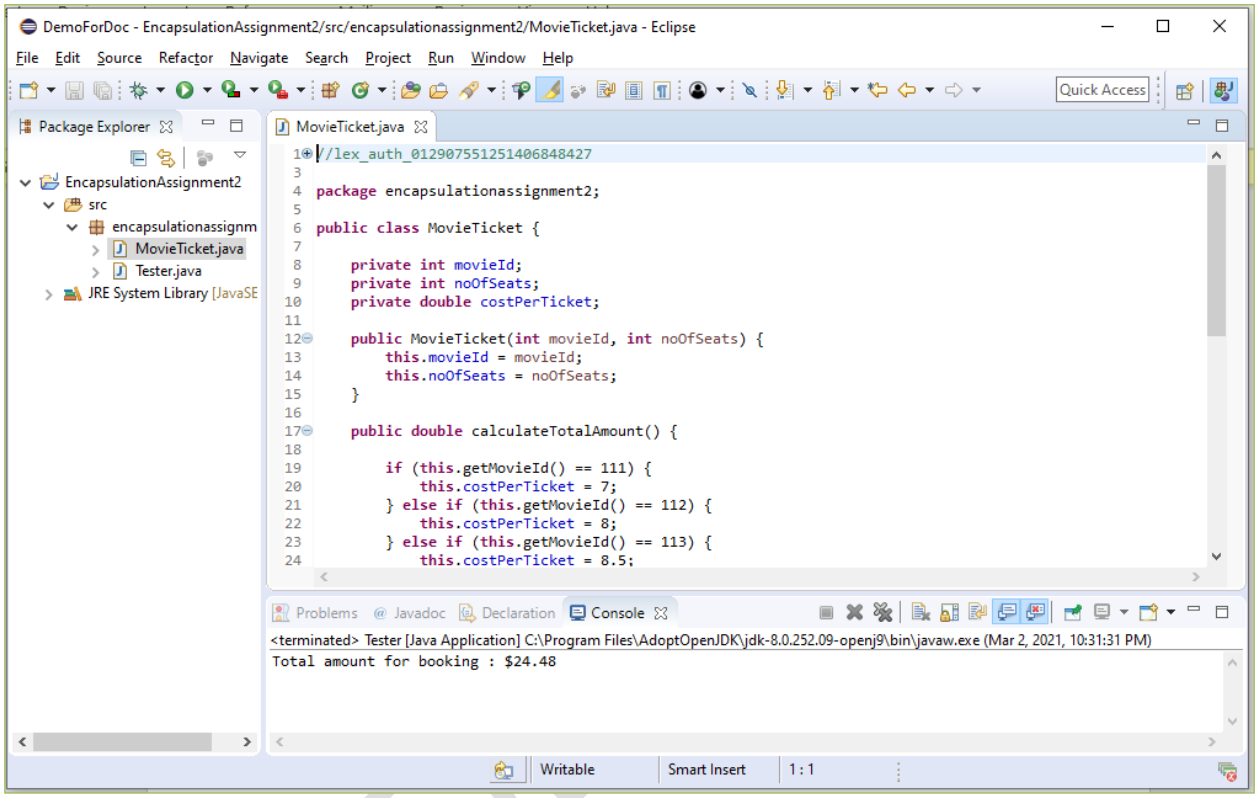
Step 18: Click on **"OK"** and **"Finish"**
Step 19: Expand the imported project
(Tester.java has errors as the code is not completely implemented)



Step 20: Double click on **"MovieTicket.java".** The file opens towards the right.



Step 21: Implement the code for **"MovieTicket.java"** and save the file. You would now observe that there are no errors in **"Tester.java"**
Run **Tester.java"** and observe the output in the **"Console"**

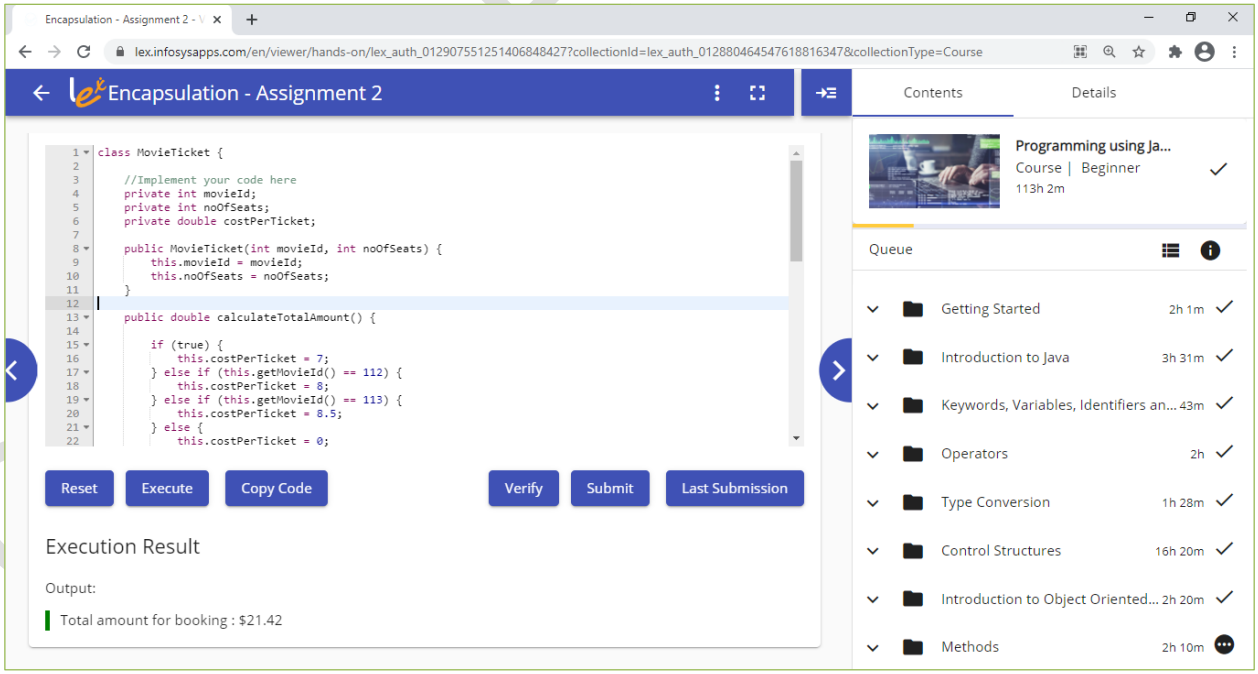


To Execute, Verify and Submit the implemented code on Lex, follow the below steps

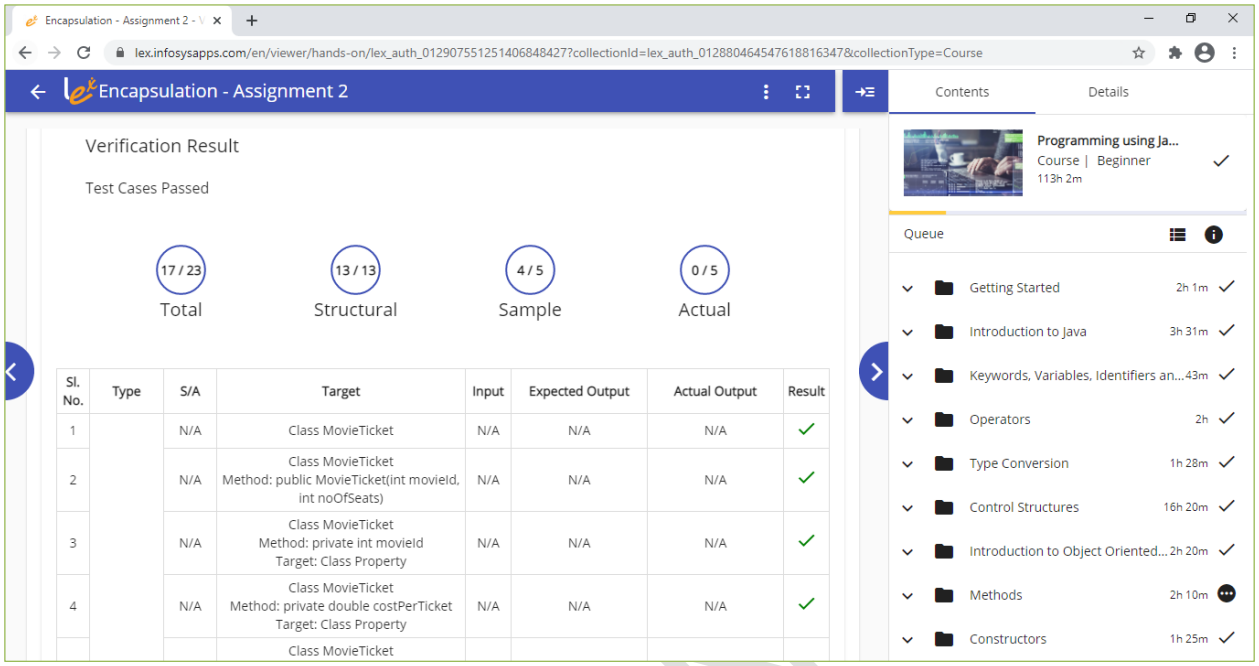
Step 22: Go to Encapsulation – Assignment 2 page in Lex. In the code pane, replace the line "//Implement your code here" with the code written in eclipse.

NOTE: In case you are copying the entire class definition, ensure that only "Tester class is public"

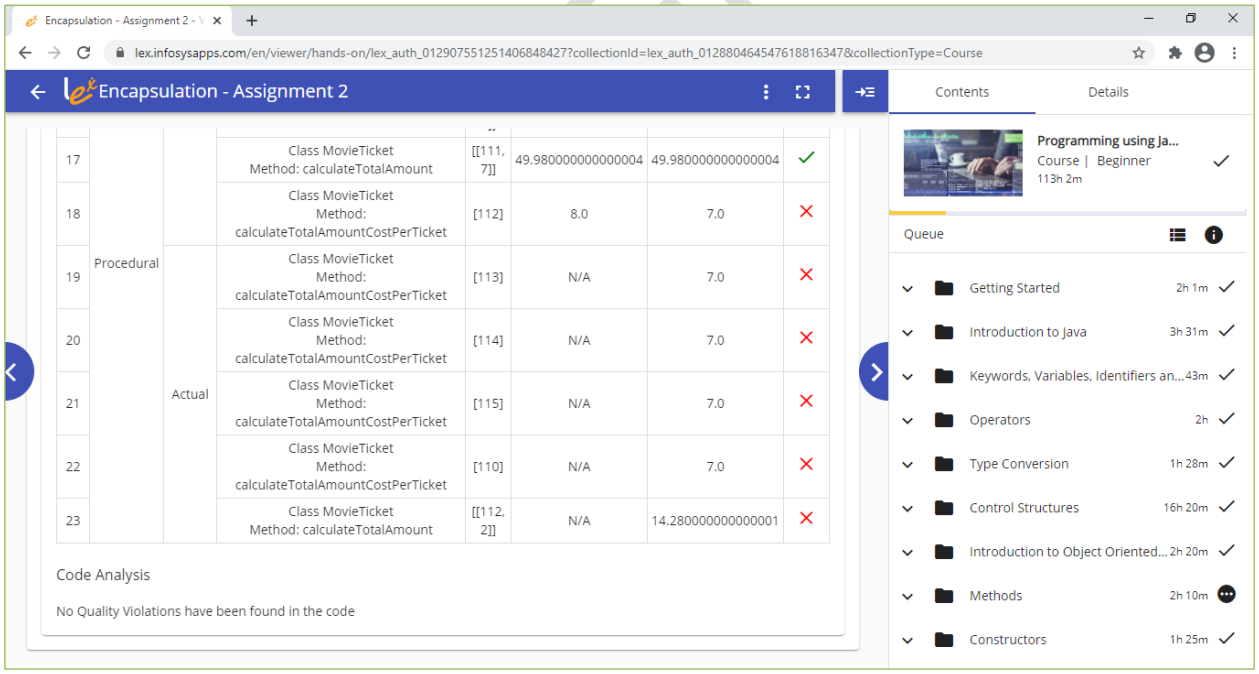
Step 23: Click on **"Execute"** and observe the results



Step 24: Click on **"Verify"** and observe the test cases. Few of the test cases are failing because the code implemented is partially correct

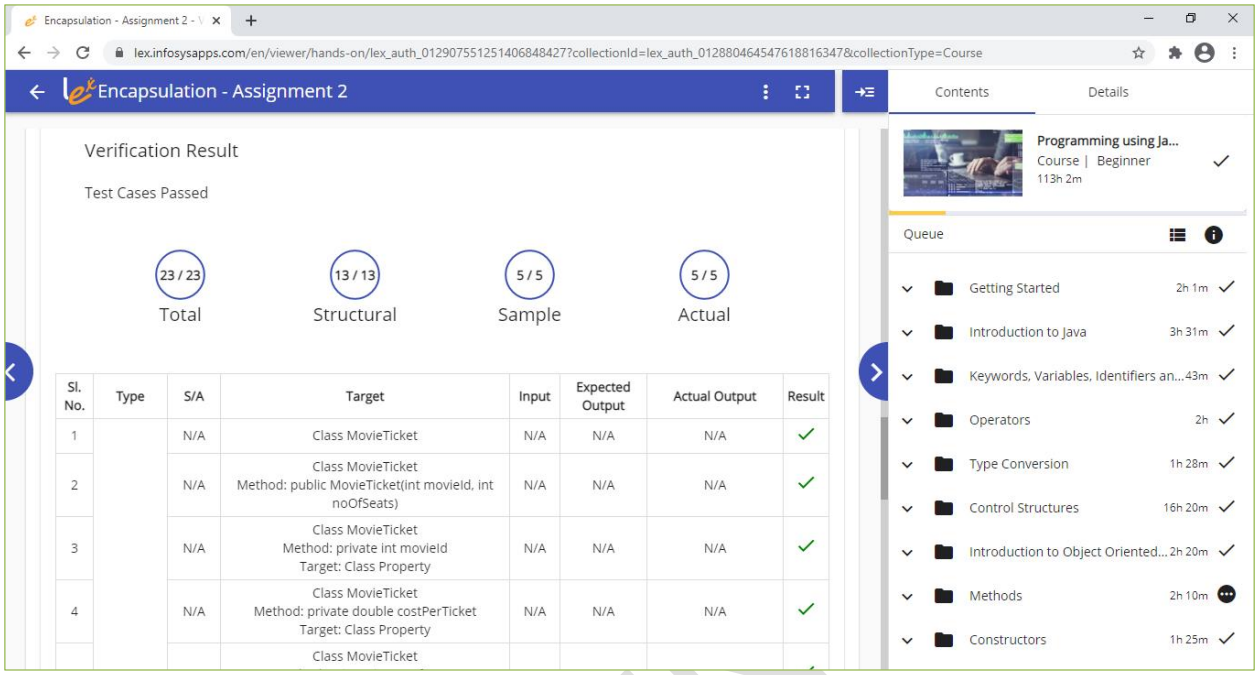


Sl. No.	Type	S/A	Target	Input	Expected Output	Actual Output	Result
1		N/A	Class MovieTicket	N/A	N/A	N/A	✓
2		N/A	Class MovieTicket Method: public MovieTicket(int movieId, int noOfSeats)	N/A	N/A	N/A	✓
3		N/A	Class MovieTicket Method: private int movieId Target: Class Property	N/A	N/A	N/A	✓
4		N/A	Class MovieTicket Method: private double costPerTicket Target: Class Property	N/A	N/A	N/A	✓
			Class MovieTicket				

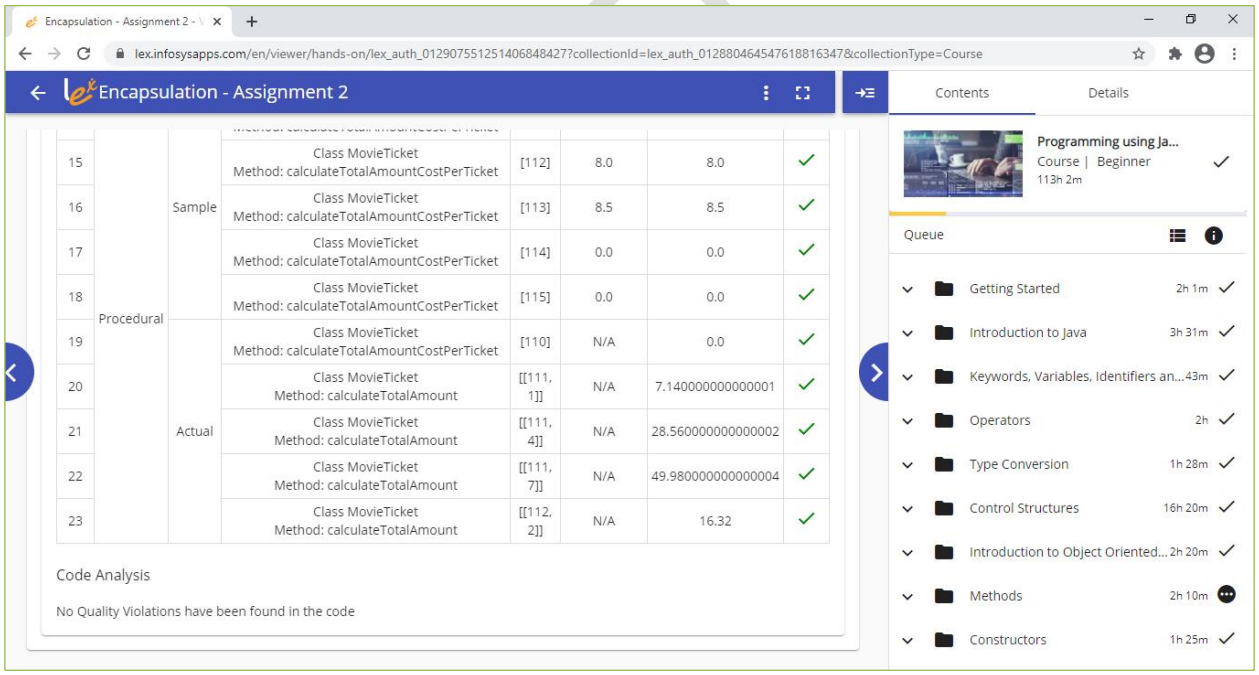


17	Procedural	Actual	Class MovieTicket Method: calculateTotalAmount	[[111, 7]]	49.980000000000004	49.980000000000004	✓
18			Class MovieTicket Method: calculateTotalAmountCostPerTicket	[112]	8.0	7.0	✗
19			Class MovieTicket Method: calculateTotalAmountCostPerTicket	[113]	N/A	7.0	✗
20			Class MovieTicket Method: calculateTotalAmountCostPerTicket	[114]	N/A	7.0	✗
21			Class MovieTicket Method: calculateTotalAmountCostPerTicket	[115]	N/A	7.0	✗
22			Class MovieTicket Method: calculateTotalAmountCostPerTicket	[110]	N/A	7.0	✗
23			Class MovieTicket Method: calculateTotalAmount	[[112, 2]]	N/A	14.280000000000001	✗

Step 25: Verify the code again with the correct solution. Observe the result. All the test cases are now passing

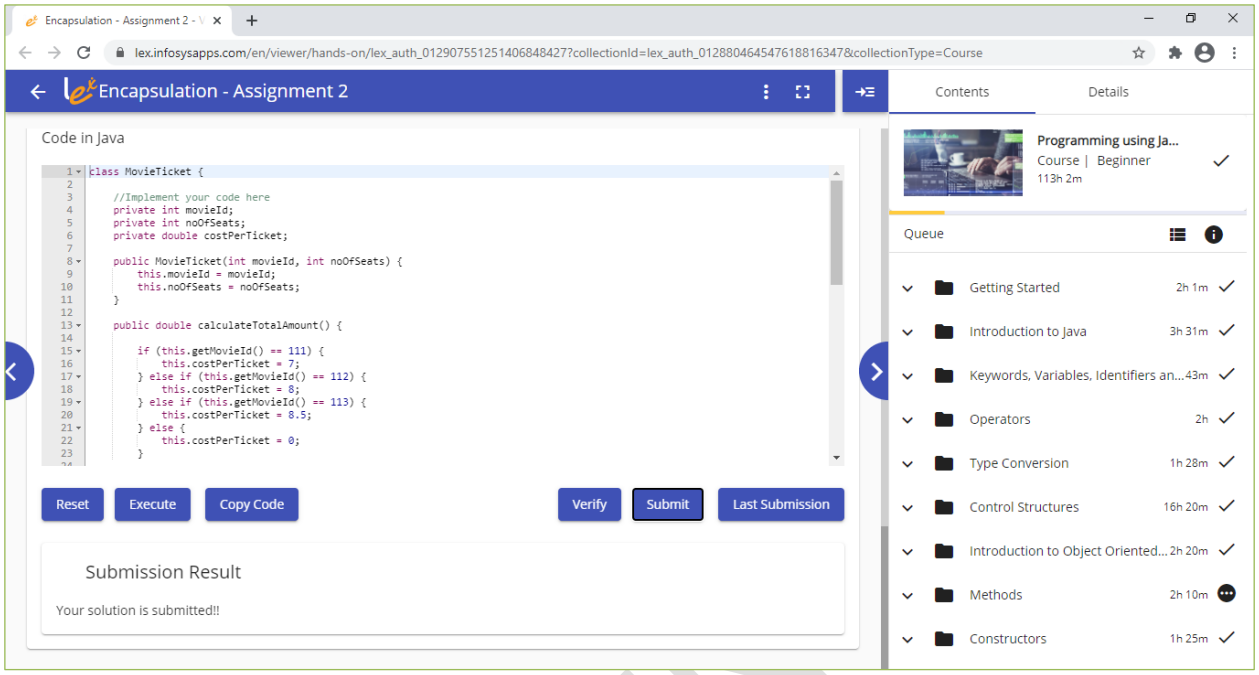


Sl. No.	Type	S/A	Target	Input	Expected Output	Actual Output	Result
1		N/A	Class MovieTicket	N/A	N/A	N/A	✓
2		N/A	Class MovieTicket Method: public MovieTicket(int movieId, int noOfSeats)	N/A	N/A	N/A	✓
3		N/A	Class MovieTicket Method: private int movieId Target: Class Property	N/A	N/A	N/A	✓
4		N/A	Class MovieTicket Method: private double costPerTicket Target: Class Property	N/A	N/A	N/A	✓
			Class MovieTicket				



15	Procedural	Sample	Class MovieTicket Method: calculateTotalAmountCostPerTicket	[112]	8.0	8.0	✓
16			Class MovieTicket Method: calculateTotalAmountCostPerTicket	[113]	8.5	8.5	✓
17			Class MovieTicket Method: calculateTotalAmountCostPerTicket	[114]	0.0	0.0	✓
18			Class MovieTicket Method: calculateTotalAmountCostPerTicket	[115]	0.0	0.0	✓
19		Actual	Class MovieTicket Method: calculateTotalAmountCostPerTicket	[110]	N/A	0.0	✓
20			Class MovieTicket Method: calculateTotalAmount	[[111, 1]]	N/A	7.140000000000001	✓
21			Class MovieTicket Method: calculateTotalAmount	[[111, 4]]	N/A	28.560000000000002	✓
22			Class MovieTicket Method: calculateTotalAmount	[[111, 7]]	N/A	49.980000000000004	✓
23			Class MovieTicket Method: calculateTotalAmount	[[112, 2]]	N/A	16.32	✓

Step 26: Once the verification is successful (All test cases passing) the code can be submitted on the server using **"Submit"**



Step 27: At any given point, the last submitted code can be retrieved by clicking on **"Last Submission"** option