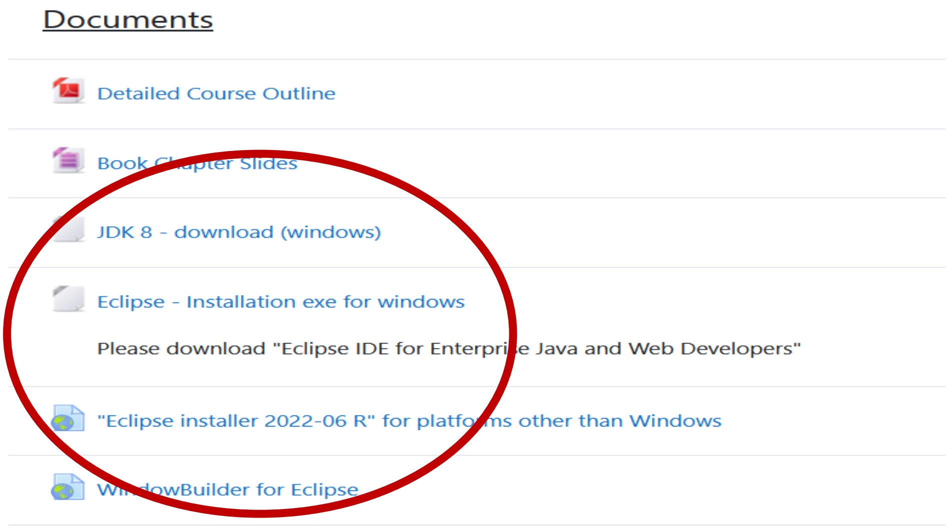
Department of Information Systems and Technologies
CTIS221 – Object Oriented Programming
SPRING 2024 – 2025
Lab Guide 1
Week 3 - 1

OBJECTIVES : Introduction to Eclipse, Java GUI API, Swing GUI Components by using WindowBuilder, Primitive Data Types and Operations
Instructor : Leyla SEZER **Assistant** : Efe Mert ŞAHİNKOÇ, Engin Z. KIRAÇBEDEL

Download Eclipse IDE and jdk8 from the Moodle page anywhere in your local PC, then extract them to any folder (Preferably extract Eclipse IDE to C:\eclipse). Do not forget, you have to download jdk version 8 first.



Documents

Detailed Course Outline

Book Chapter Slides

JDK 8 - download (windows)

Eclipse - Installation exe for windows

Please download "Eclipse IDE for Enterprise Java and Web Developers"

"Eclipse installer 2022-06 R" for platforms other than Windows

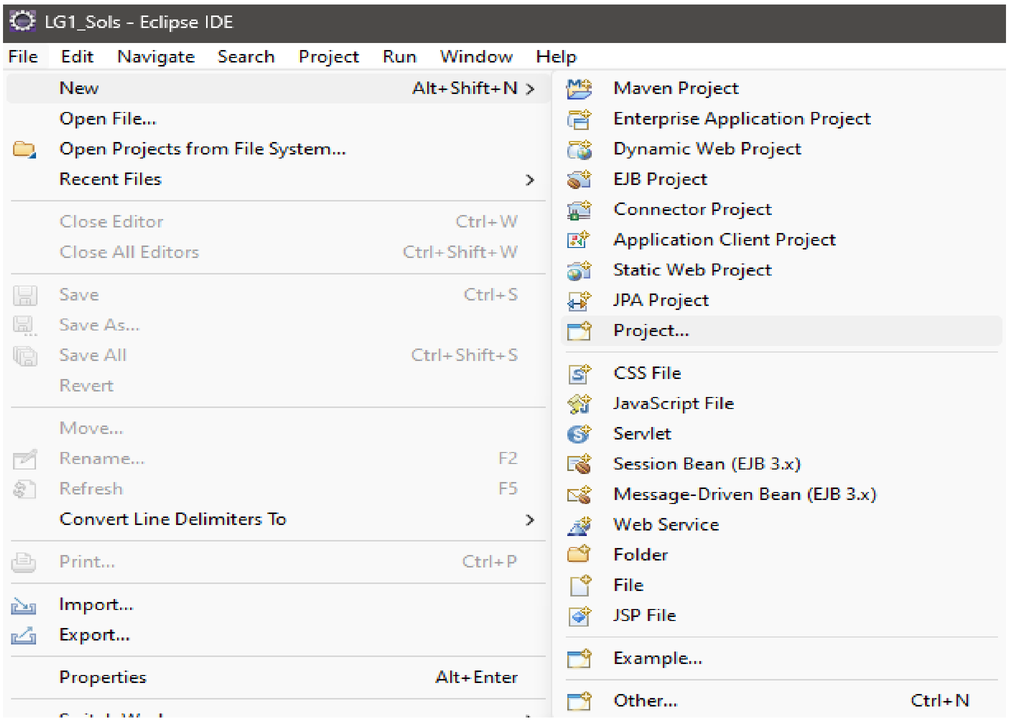
WindowBuilder for Eclipse

<u>There is also a link for the WindowBuilder. The connection of the Eclipse with the WindowBuilder will be explained in the class creation part of this document.</u>

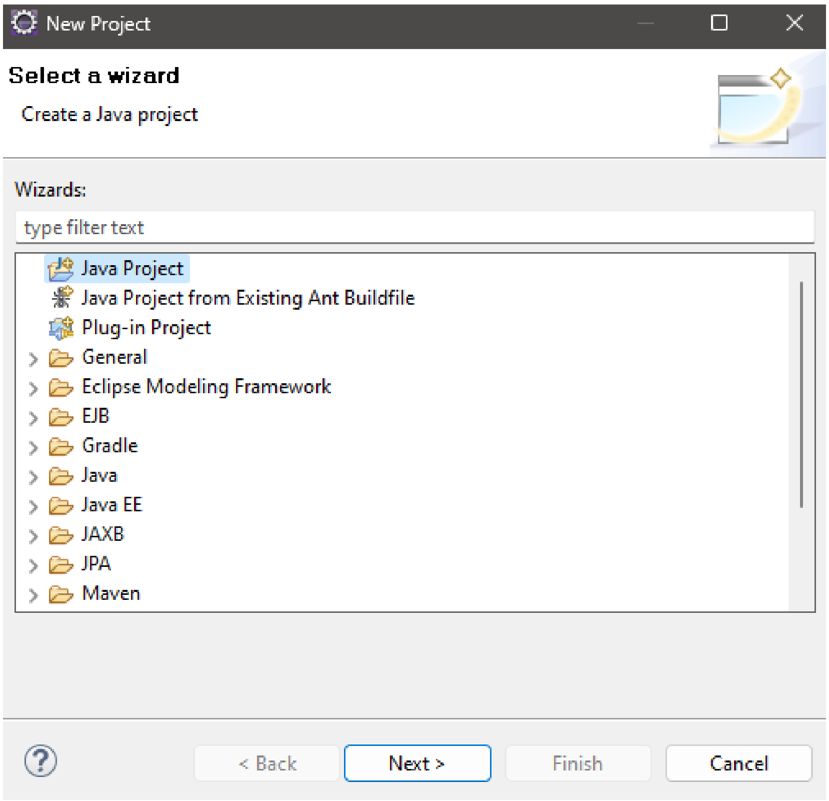
<u>Using Eclipse:</u>

While using Eclipse IDE, there are some steps you need to follow:

1. In the “**File**” menu, select “**New**” then **“Project”**.



2. Here, select **“Java Project”** then click **“Next”**.



New Project

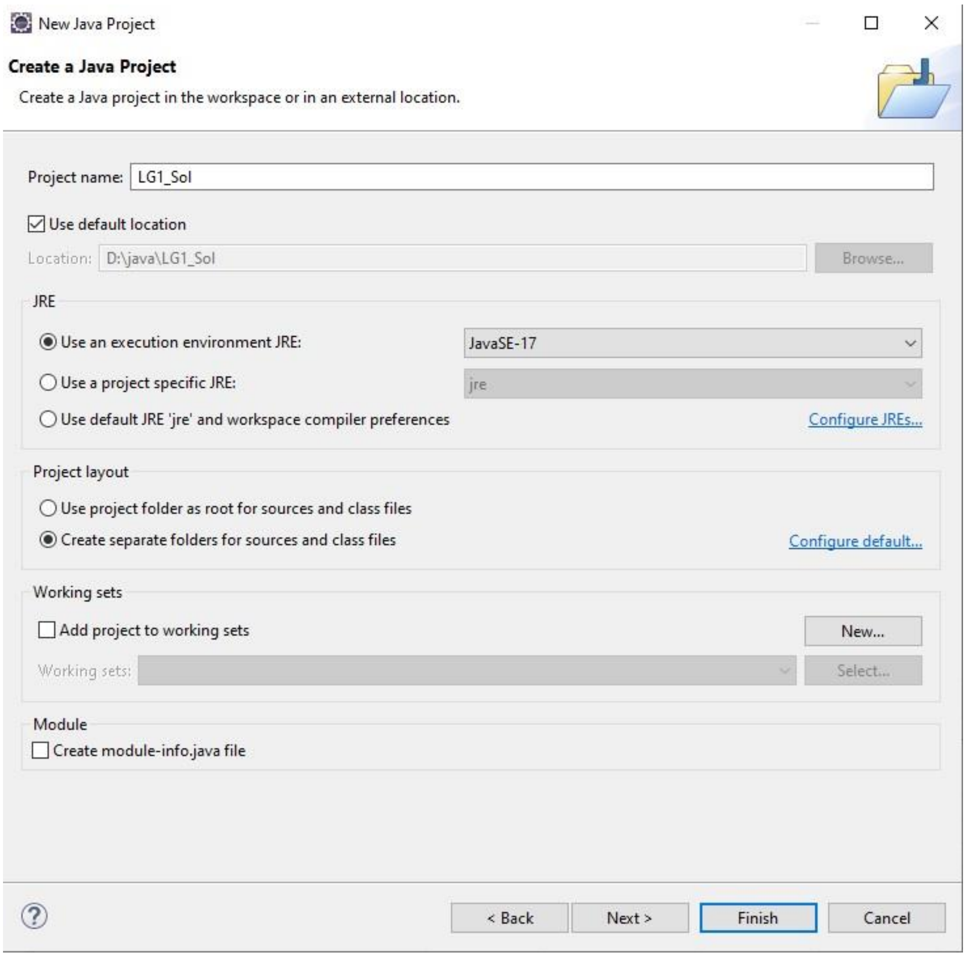
Select a wizard

Create a Java project

Wizards:

type filter text

3. Then name your project and follow the screenshot below. Uncheck **“Create module-info.java file”** and click **“Finish”**.



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: LG1_Sol

☑ Use default location

Location: D:\java\LG1_Sol

JRE

● Use an execution environment JRE: JavaSE-17

○ Use a project specific JRE: jre

○ Use default JRE 'jre' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

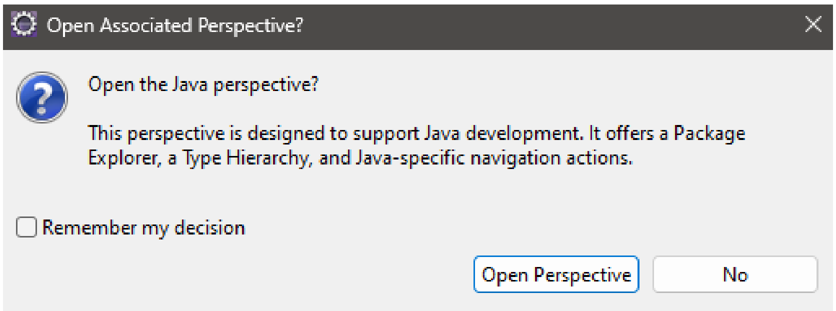
□ Add project to working sets

Working sets:

Module

□ Create module-info.java file

4. If a window appears asking you to open the Java Perspective, click **"Open Perspective"**. You may also check **"Remember my decision"** for your future Java projects, so that this window will not appear again.



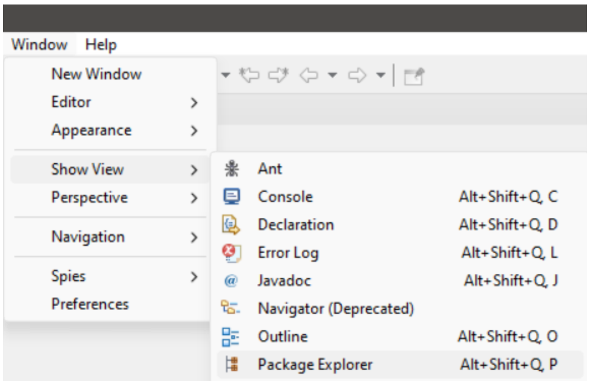
Open Associated Perspective?

Open the Java perspective?

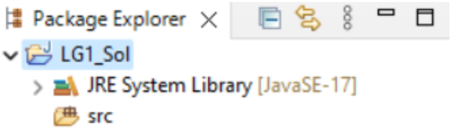
This perspective is designed to support Java development. It offers a Package Explorer, a Type Hierarchy, and Java-specific navigation actions.

☐ Remember my decision

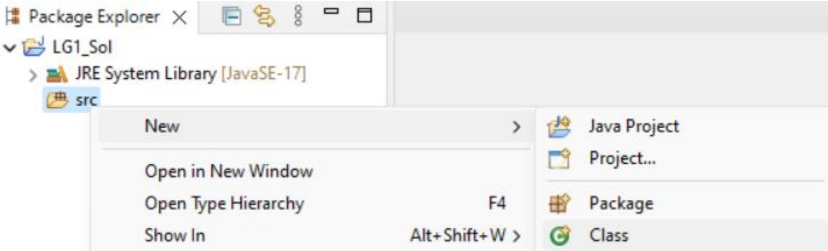
5. You can close the **"Welcome"** tab and if **"Package Explorer"** is not shown, click **"Window"** menu, **"Show View"** then **"Package Explorer"**.



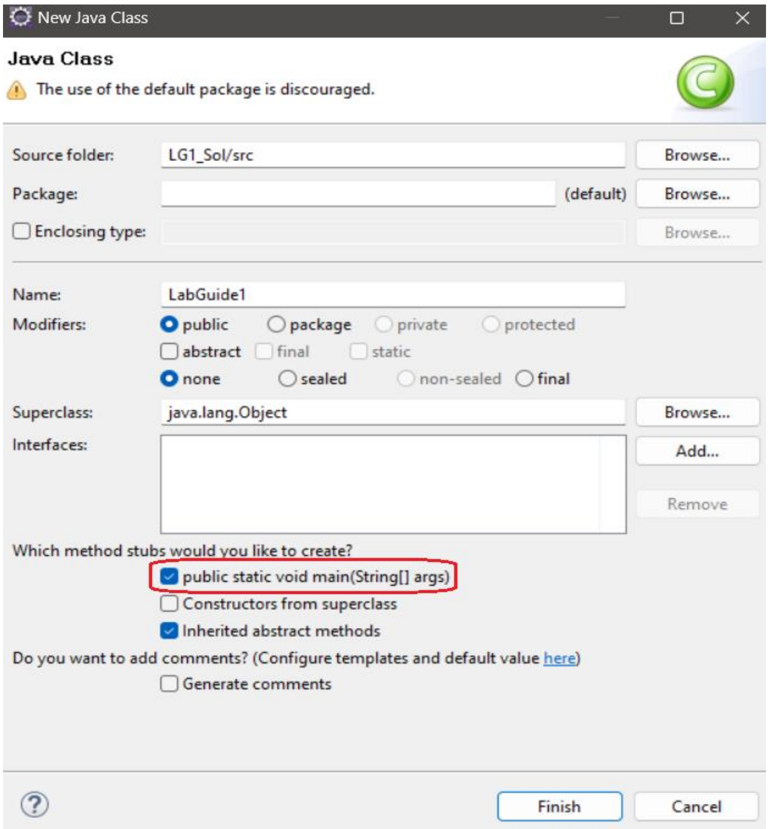
6. You should now see the **"Package Explorer"** on the left side of your **"Eclipse IDE"**.



7. In order to create your first class, over **"Package Explorer"** right click to **"src"** (source) folder, click **"New"** then **"Class"**.



8. Give **“LabGuide1”** name to your class and select **“public static void main(String[] args)”** check box, then click **“Finish”**.



New Java Class

Java Class

⚠ The use of the default package is discouraged.

Source folder: LG1_Sol/src [Browse...]

Package: (default) [Browse...]

☐ Enclosing type: [Browse...]

Name: LabGuide1

Modifiers: ● public ○ package ○ private ○ protected
☐ abstract ☐ final ☐ static
● none ○ sealed ○ non-sealed ○ final

Superclass: java.lang.Object [Browse...]

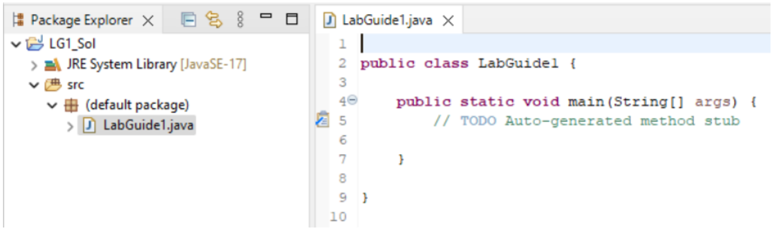
Interfaces: [Add...] [Remove]

Which method stubs would you like to create?
☑ public static void main(String[] args)
☐ Constructors from superclass
☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)
☐ Generate comments

[Finish] [Cancel]

9. Your class is created as in the screenshot below.



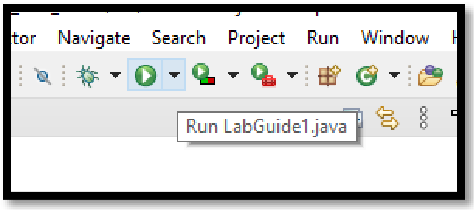
Package Explorer
- LG1_Sol
  - JRE System Library [JavaSE-17]
  - src
    - (default package)
      - LabGuide1.java

LabGuide1.java

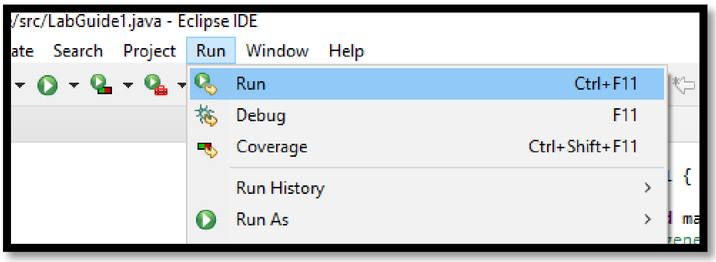
```

public class LabGuide1 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

	}

}

```

10. To run a project, there are 3 ways:

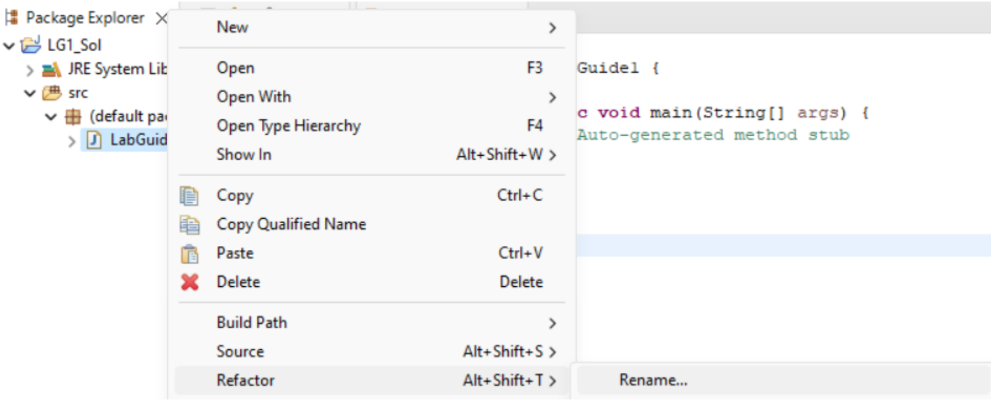
- First one is to use the toolbar item: **“Run Project”**.



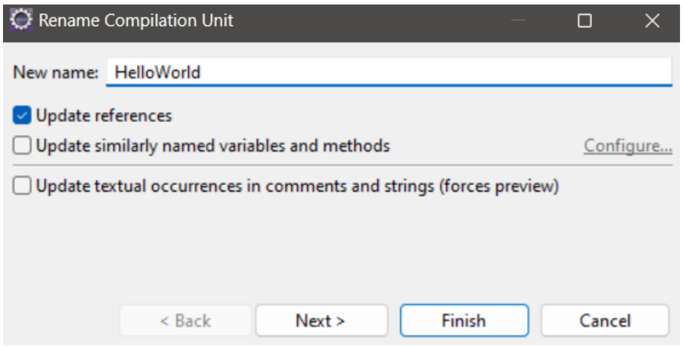
- Second one is to run the project from Run menu item.
- Third one is to run the project using **“Ctrl + F11”** keys.


/src/LabGuide1.java - Eclipse IDE
ate Search Project Run Window Help
Run Ctrl+F11
Debug F11
Coverage Ctrl+Shift+F11
Run History
Run As

11. If you want, you can **refactor (reform)** your project, for example selecting your class file under **"Package Explorer"**, right clicking it then clicking **"Refactor"** and **"Rename"**.



12. You can rename your class file to **"HelloWorld"** as shown in the screenshot below. It is preferred that class file names are same as class names in code, therefore do not forget to check **"Update references"** then click **"Finish"**.


Rename Compilation Unit

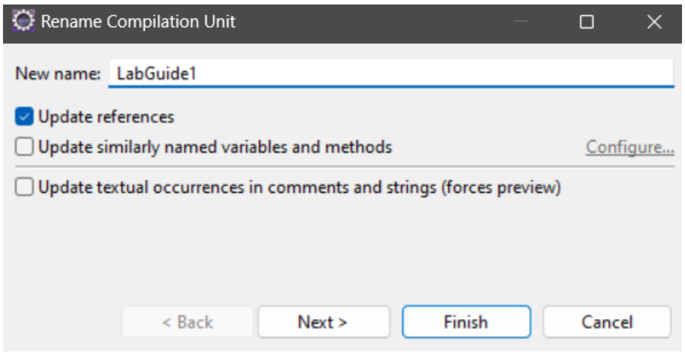
New name: HelloWorld

☑ Update references

□ Update similarly named variables and methods

□ Update textual occurrences in comments and strings (forces preview)

13. After you complete renaming your class to **"HelloWorld"**, rename it back to **"LabGuide1"** as described above.


Rename Compilation Unit

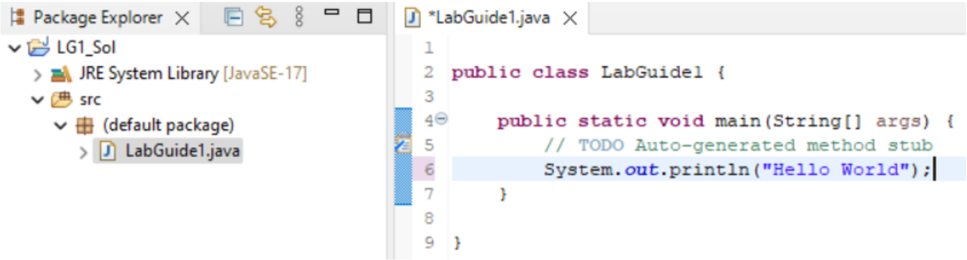
New name: LabGuide1

☑ Update references

□ Update similarly named variables and methods

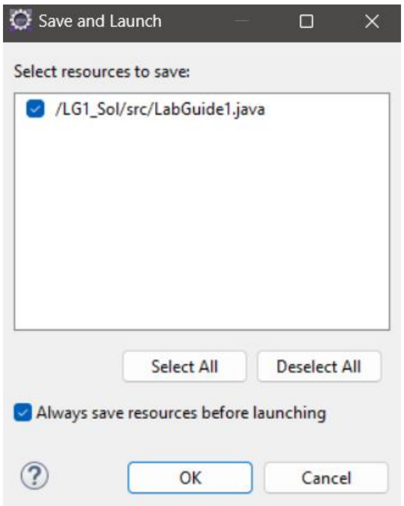
□ Update textual occurrences in comments and strings (forces preview)

14. From **“Package Explorer”** double click **“LabGuide1.java”** class file then modify the **<u>main method</u>** as shown below.



```

public class LabGuide1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello World");
    }

}
```

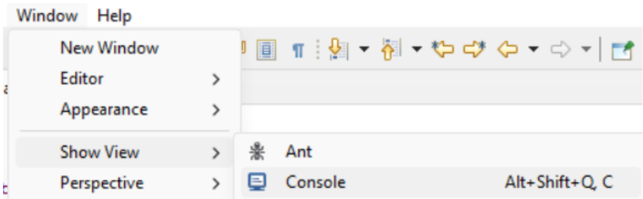
15. Run your project then click the **“Window”** menu, then **“Show View”** and **“Console”**. If you did not save your class file, a window may appear asking you to save it. It is a good way to click **“Always save resources before launching”** so that this window will not appear again and your modified files will be automatically saved when you run the project.



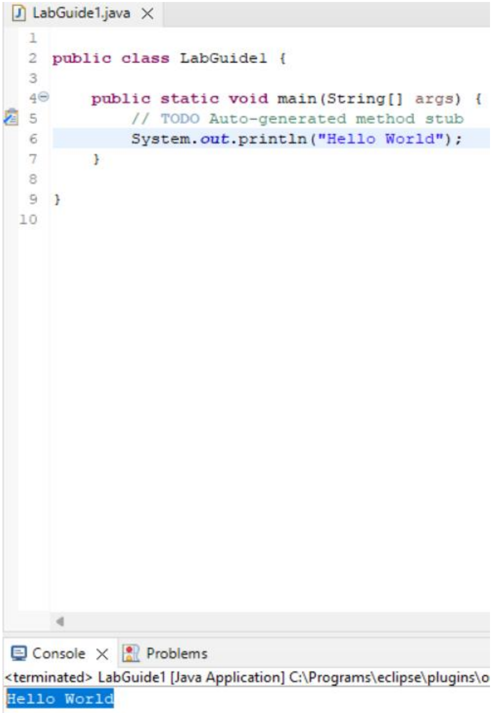
Select resources to save:

☑ /LG1_Sol/src/LabGuide1.java

☑ Always save resources before launching



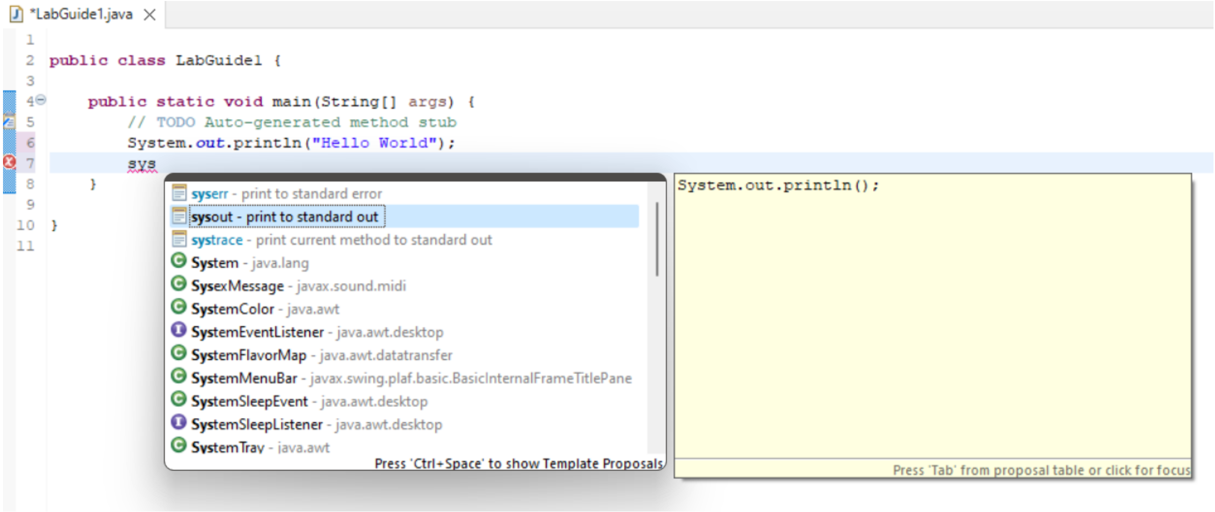
16. You should see the console output as in the screenshot below.



```

public class LabGuide1 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		System.out.println("Hello World");
	}

}

```

```
<terminated> LabGuide1 [Java Application] C:\Programs\eclipse\plugins\o
Hello World
```

17. In Java programming language, you don’t have to memorize anything. The classes and their methods are predefined, so you are going to use them according to your needs. However, in order to get acquainted with predefined Java classes and methods, you need to understand the **javadoc** principles. For example, if you want to use the **“println”** method of the **“System”** class, other than writing it as **System.out.println(“Hello World”);** you can write **sys**, then hit **“Ctrl + Space”** keys and select **sysout** from the shown menu. This will write the code line for you. You can also write **sysout** then hit **“Ctrl + Space”** keys and the code line will also be written.


*LabGuide1.java
syserr - print to standard error
sysout - print to standard out
systrace - print current method to standard out
System - java.lang
SysexMessage - javax.sound.midi
SystemColor - java.awt
SystemEventListener - java.awt.desktop
SystemFlavorMap - java.awt.datatransfer
SystemMenuBar - javax.swing.plaf.basic.BasicInternalFrameTitlePane
SystemSleepEvent - java.awt.desktop
SystemSleepListener - java.awt.desktop
SystemTray - java.awt
Press 'Ctrl+Space' to show Template Proposals
Press 'Tab' from proposal table or click for focus

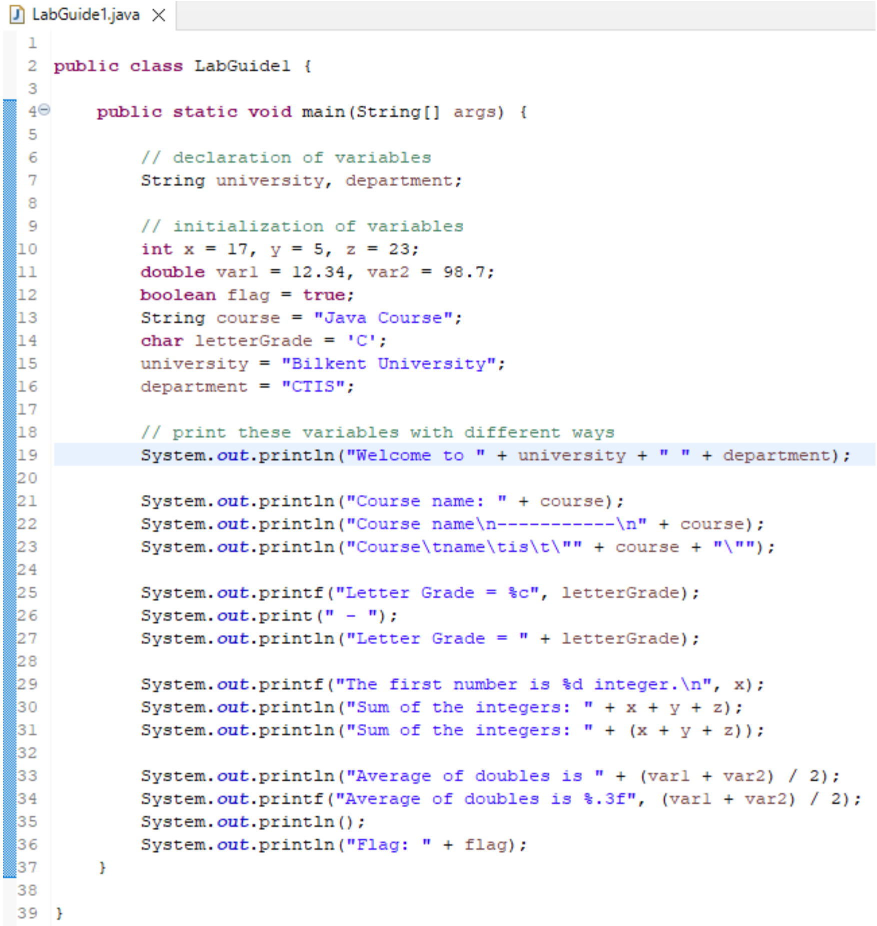
```

public class LabGuide1 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		System.out.println("Hello World");
		sys
	}

}

```

```
System.out.println();
```

18. While a developer is coding, s/he can write a predefined Java class name such as **“System.”**, then s/he can hit **“Ctrl + Space”** keys to see what **class** related **data members** or **methods** can be used. The explanations of the data members and methods can also be seen.

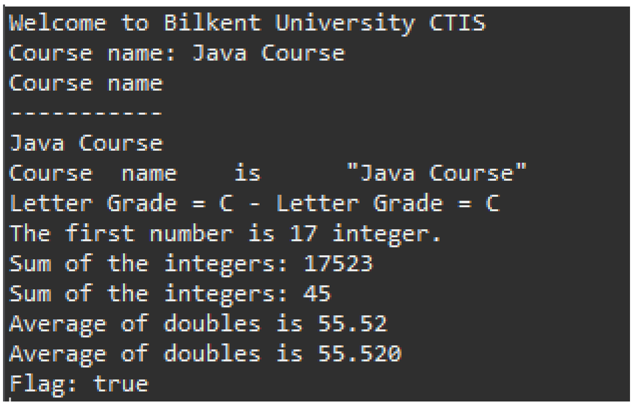
Exercises

Q1. Modify the **LabGuide1.java** file to get the following output.



```java

public class LabGuide1 {

    public static void main(String[] args) {

        // declaration of variables
        String university, department;

        // initialization of variables
        int x = 17, y = 5, z = 23;
        double var1 = 12.34, var2 = 98.7;
        boolean flag = true;
        String course = "Java Course";
        char letterGrade = 'C';
        university = "Bilkent University";
        department = "CTIS";

        // print these variables with different ways
        System.out.println("Welcome to " + university + " " + department);

        System.out.println("Course name: " + course);
        System.out.println("Course name\n-----------\n" + course);
        System.out.println("Course\tname\tis\t\"" + course + "\"");

        System.out.printf("Letter Grade = %c", letterGrade);
        System.out.print(" - ");
        System.out.println("Letter Grade = " + letterGrade);

        System.out.printf("The first number is %d integer.\n", x);
        System.out.println("Sum of the integers: " + x + y + z);
        System.out.println("Sum of the integers: " + (x + y + z));

        System.out.println("Average of doubles is " + (var1 + var2) / 2);
        System.out.printf("Average of doubles is %.3f", (var1 + var2) / 2);
        System.out.println();
        System.out.println("Flag: " + flag);
    }

}
```

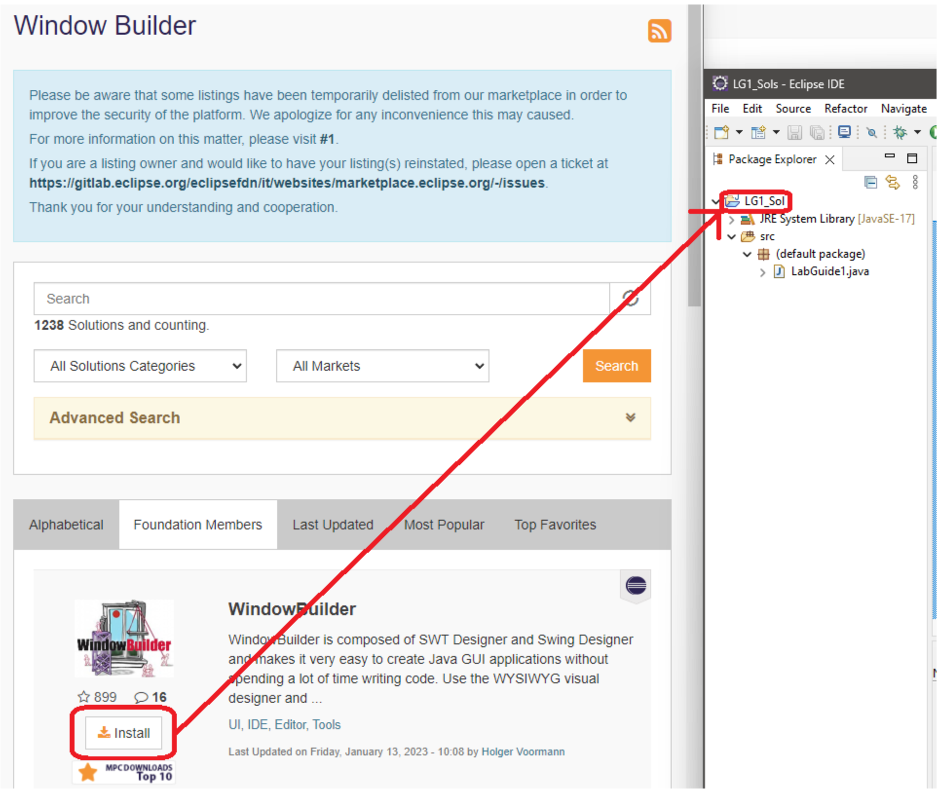
<u>Output:</u>



```
Welcome to Bilkent University CTIS
Course name: Java Course
Course name
-----------
Java Course
Course  name    is      "Java Course"
Letter Grade = C - Letter Grade = C
The first number is 17 integer.
Sum of the integers: 17523
Sum of the integers: 45
Average of doubles is 55.52
Average of doubles is 55.520
Flag: true
```

Q2. Creating a Java project with **Graphical User Interface (GUI)**.

- Go to the **"WindowBuilder"** link from the **Moodle** page, then drag and drop the **Install** icon into your Eclipse Java project named **"LG1_Sol"**. After the installation, restart Eclipse.



Window Builder

Please be aware that some listings have been temporarily delisted from our marketplace in order to improve the security of the platform. We apologize for any inconvenience this may caused.

For more information on this matter, please visit **#1**.

If you are a listing owner and would like to have your listing(s) reinstated, please open a ticket at **https://gitlab.eclipse.org/eclipsefdn/it/websites/marketplace.eclipse.org/-/issues**.

Thank you for your understanding and cooperation.

1238 Solutions and counting.

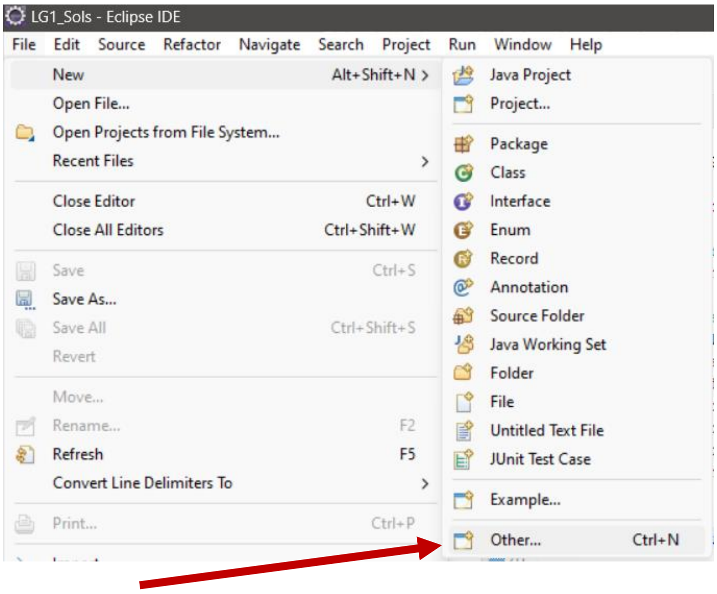
WindowBuilder

WindowBuilder is composed of SWT Designer and Swing Designer and makes it very easy to create Java GUI applications without spending a lot of time writing code. Use the WYSIWYG visual designer and ...

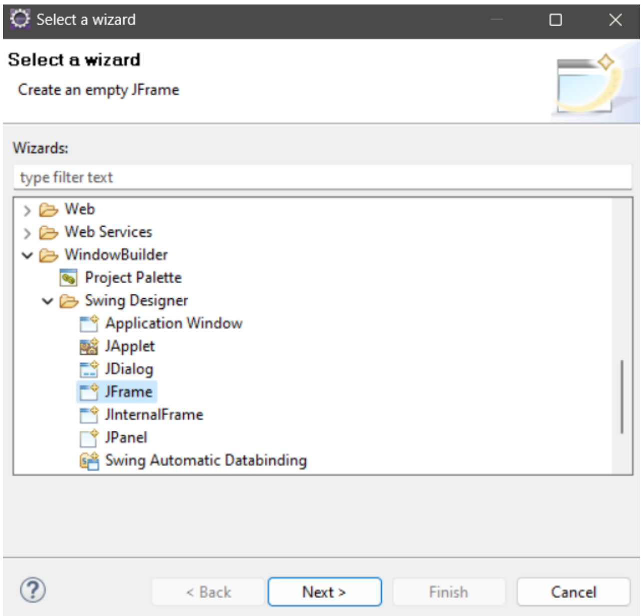
UI, IDE, Editor, Tools

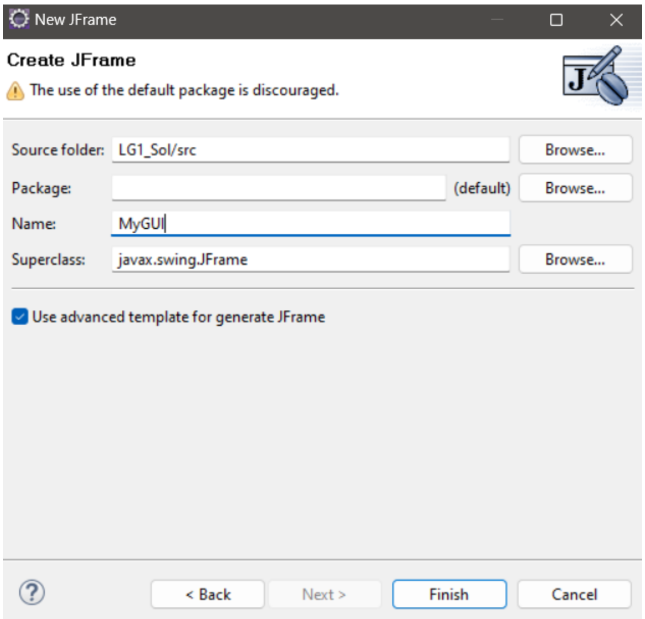
Last Updated on Friday, January 13, 2023 - 10:08 by Holger Voormann

- Then click **"File"** menu of Eclipse, then **"New"** and **"Other"**.



- Next, select **"WindowBuilder"** then **"JFrame"** to create your **GUI** named **"MyGUI"** as shown in the below screenshots**.**





⚠ The use of the default package is discouraged.

Source folder: LG1_Sol/src

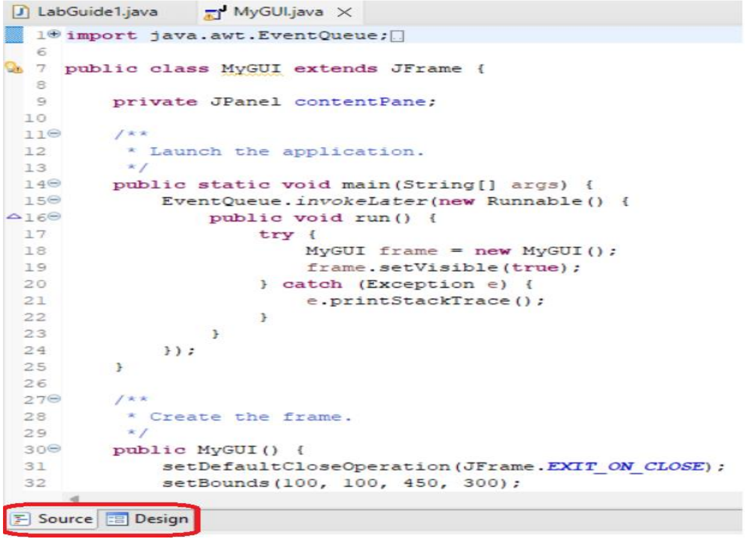
Package: (default)

Name: MyGUI

Superclass: javax.swing.JFrame

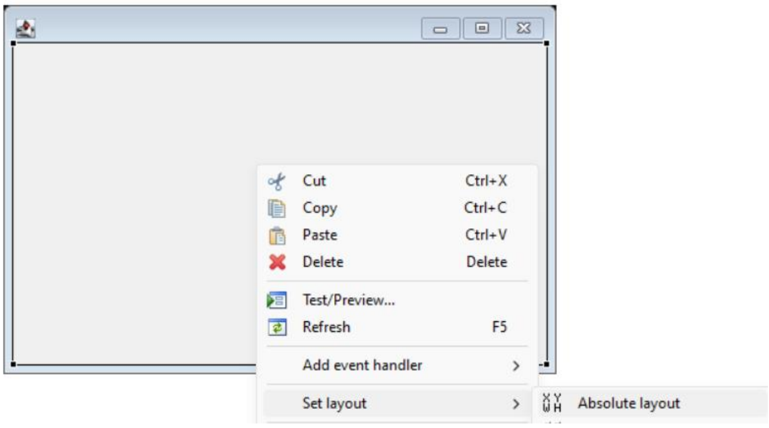
☑ Use advanced template for generate JFrame

- **“Source”** part is for writing your code, **“Design”** part is the **GUI Frame** which you can do drag and drop operations using **“Palette”** components.

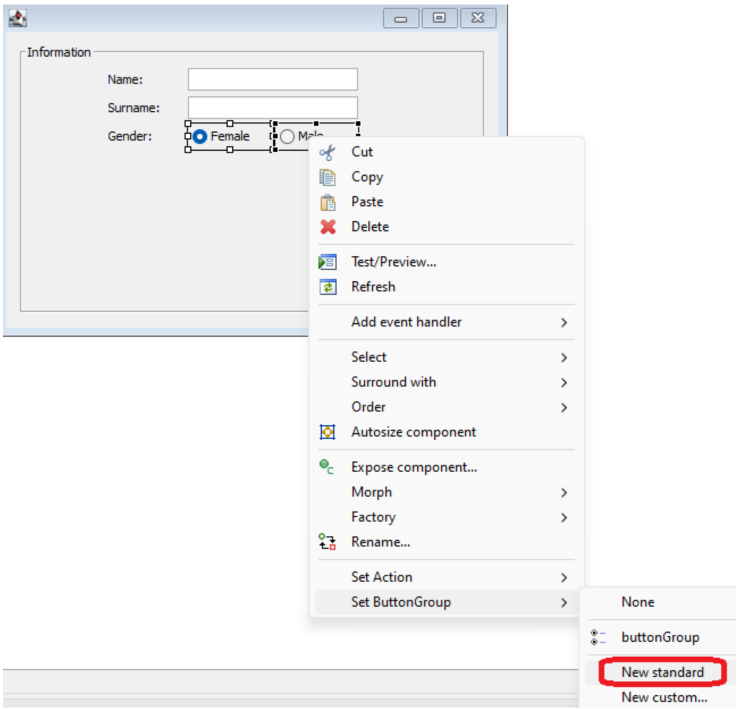


```
import java.awt.EventQueue;□

public class MyGUI extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    MyGUI frame = new MyGUI();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public MyGUI() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
```

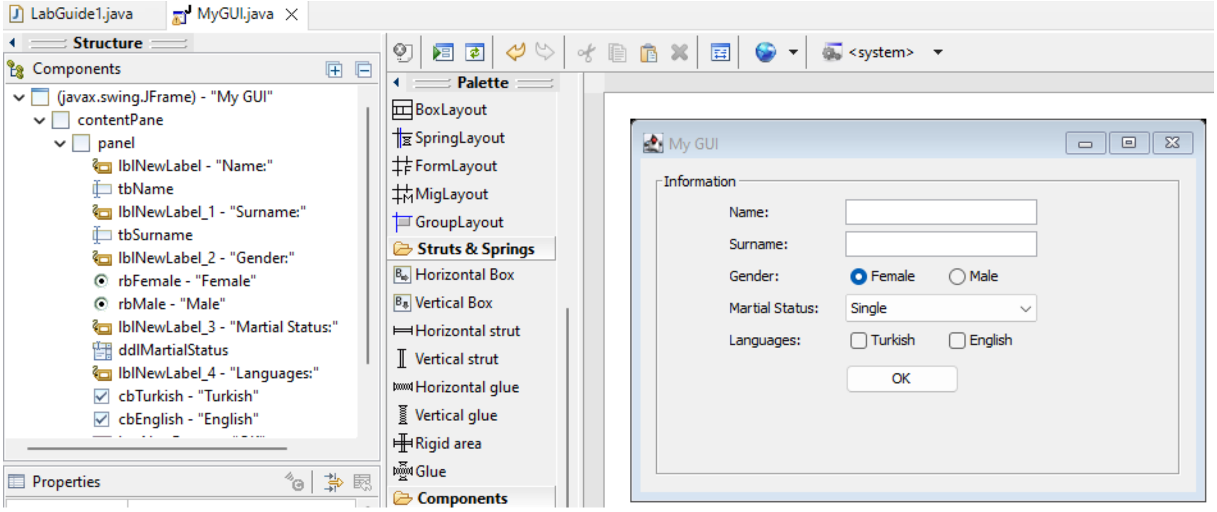
- After creating the **“JFrame”** for the first time, change layout to **“Absolute layout”** once as shown in the screenshot below by clicking the right mouse button anywhere on the **“JFrame”**.



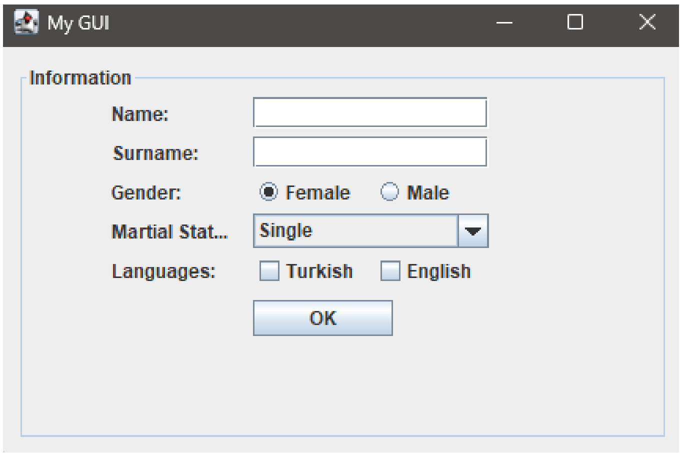
- **"Palette"** window is for dragging and dropping of **Swing** components, **"Components"** window is for tracking down the components stored in **"JFrame"** and **"Properties"** window is setting properties of the selected component. Design your **GUI** as shown in the screenshot below and run your project.
  - **Tip 1:** If you use a **"JPanel"** in your **"JFrame"**, in order to drag and drop components anywhere in your **"JPanel"**, you should right click anywhere on the **"JPanel"** and set layout to **absolute layout**.
  - **Tip 2:** If you use **"JRadioButton"** components and want to group them, you should look at the screenshot below.



Eclipse IDE screenshot:

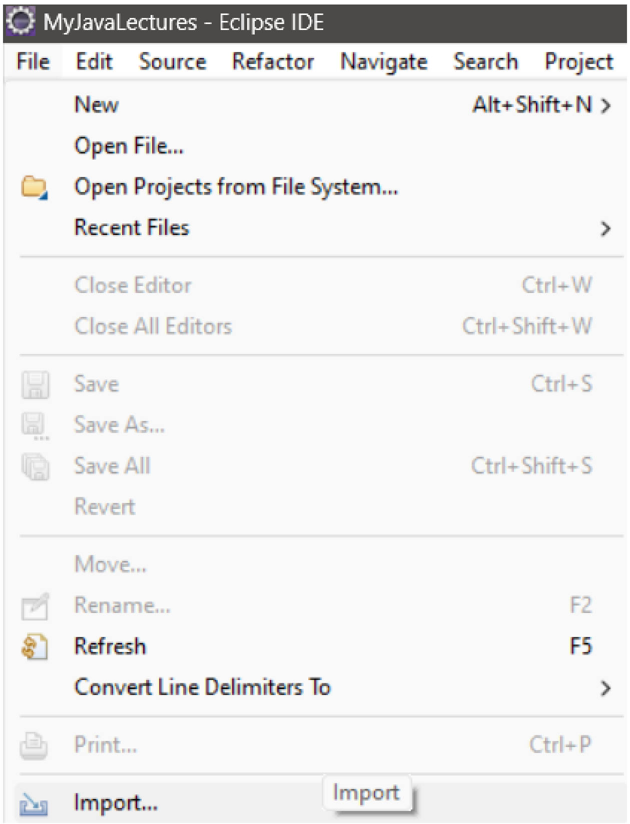


Running application screenshot:

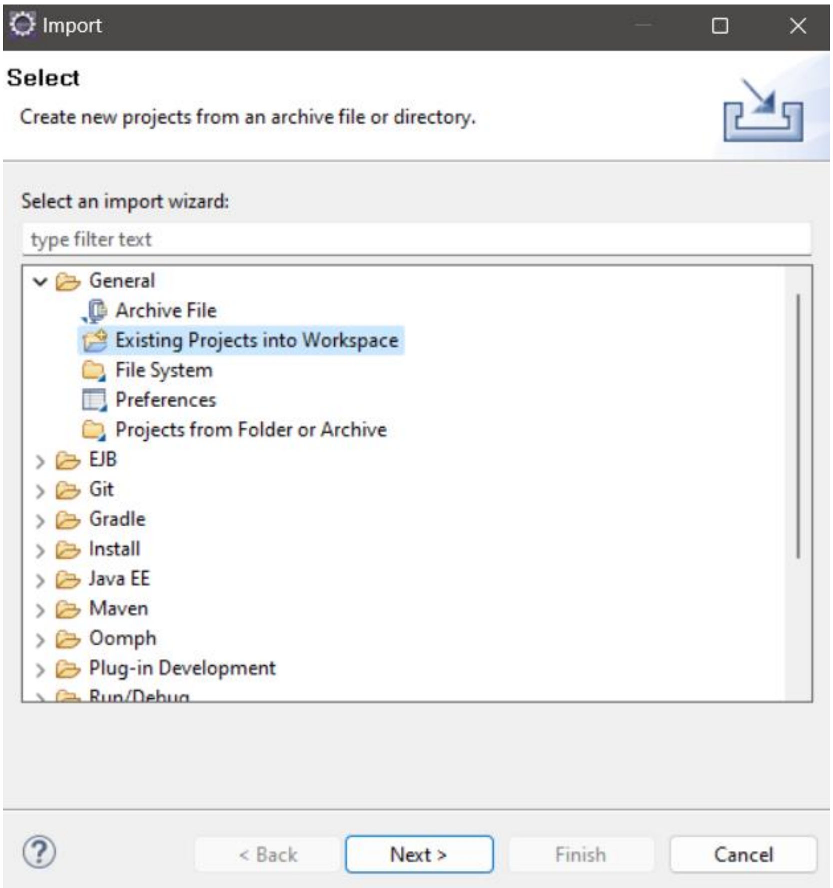


Q3. Importing a **Java Project** using **Eclipse IDE**.

- Run Eclipse IDE application.
- Click **"File menu"** → **"Import"**.



- Click **“General”** and then select **“Existing Projects into Workspace”**. Then click **“Next”**.



Import

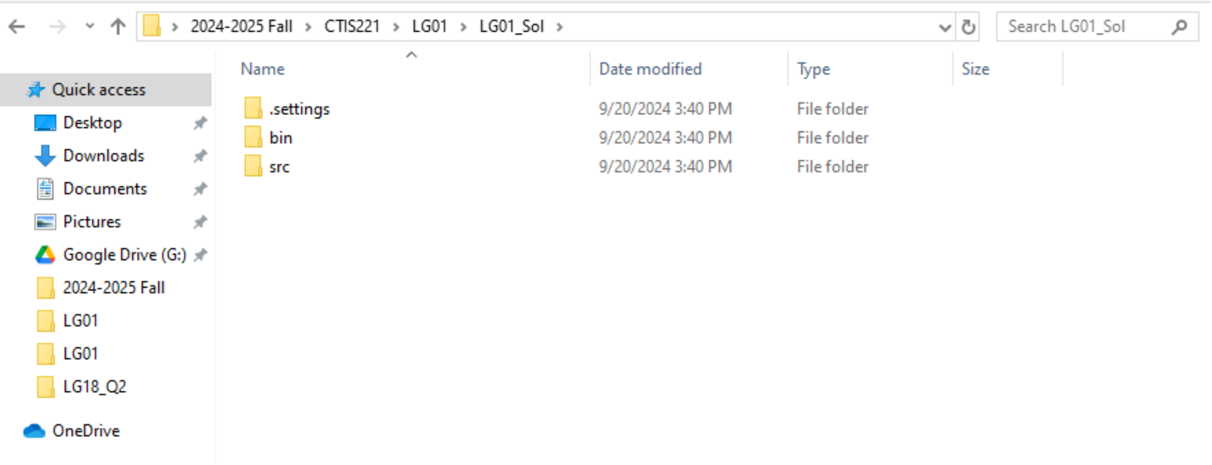
Select

Create new projects from an archive file or directory.

Select an import wizard:

type filter text

- Check **“Select root directory”** option then click **“Browse”** and select the **project** folder that contains **“src”** folder inside.

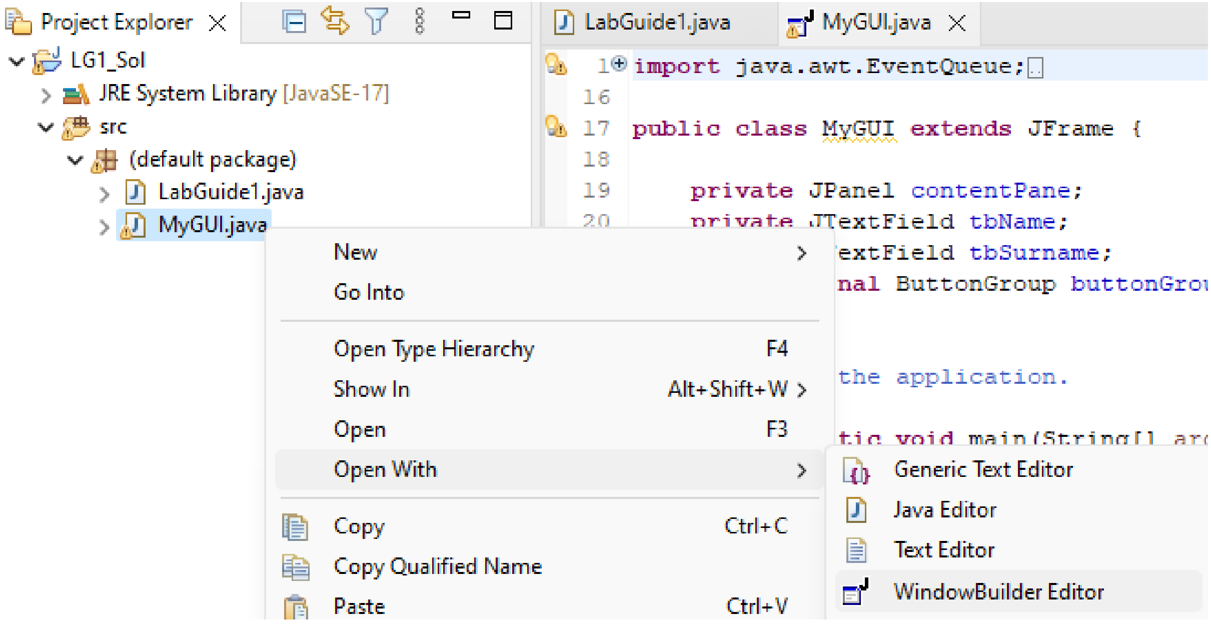


Name	Date modified	Type	Size
.settings	9/20/2024 3:40 PM	File folder	
bin	9/20/2024 3:40 PM	File folder	
src	9/20/2024 3:40 PM	File folder	

- After clicking **“Finish”**, you can now see your imported **project** in **“Package Explorer”**.

- If your project only has classes with **main method** inside, you can double click on them to open, modify and run.

- If your project has **GUI**, you should right click on your class file, move your mouse pointer to **“Open With”** then click **“WindowBuilder Editor”**. Therefore you can continue working with the **“Source”** and **“Design”** tabs.



```
import java.awt.EventQueue;

public class MyGUI extends JFrame {

    private JPanel contentPane;
    private JTextField tbName;
                          extField tbSurname;
                          nal ButtonGroup buttonGro


                          the application.

                          tic void main(String[] arg
```