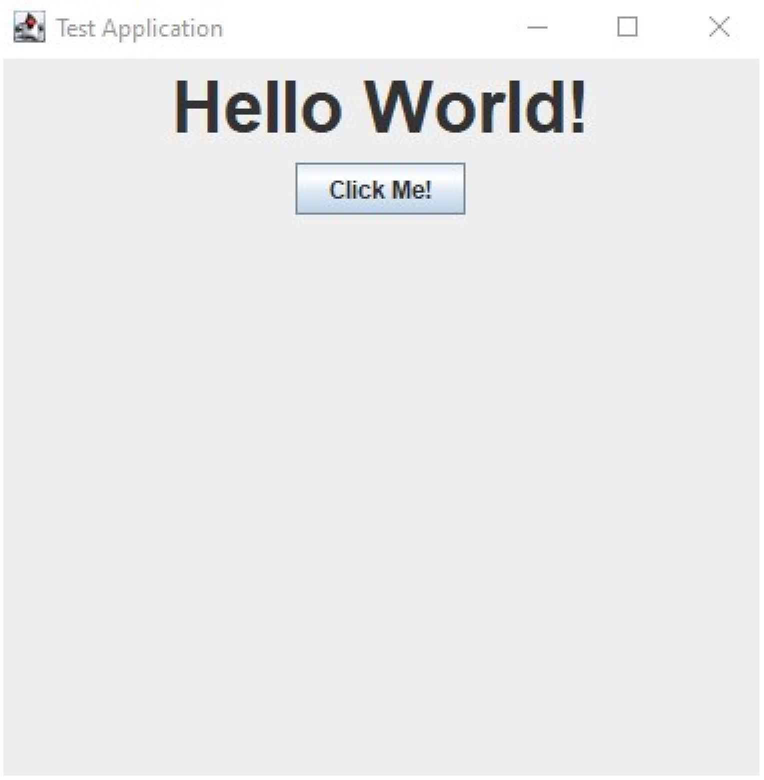
How to Create a Graphical User Interface in Java



Created by Evan Vicidomini, [redacted]

WARNING: Please make sure to follow the instructions carefully and in order. This is very important for understanding the development process.

Introduction and Intended Audience

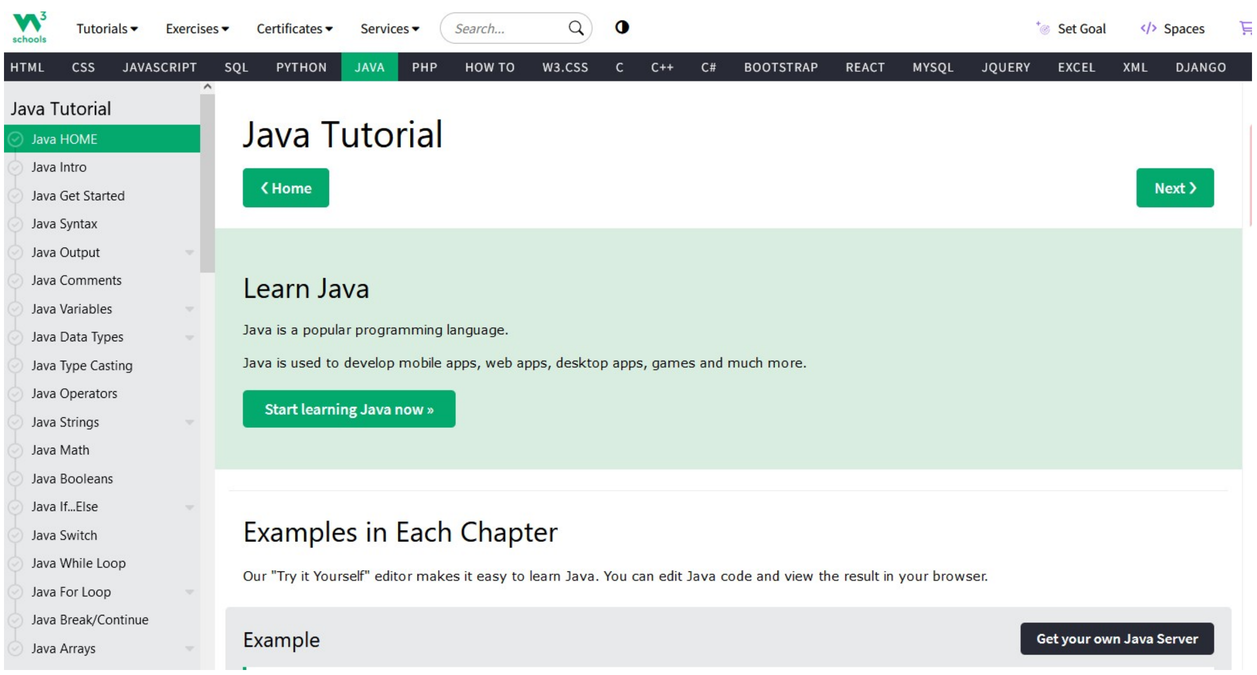
Learning computer programming can be very challenging for people. This is because it is based on higher level mathematics, like algebra. Math is a subject that many students struggle in. However, when someone understands it, they develop their problem solving skills, which is crucial to understanding computer programming. For this guide, we will be looking at Java, a high level computer programming language that is used for so many applications.

This guide is for students or people who understand the basics of Java. They want to take their basic programming skills and use them to make real applications for a desktop computer. It is recommended that the person reading this guide has completed a computer programming course related to Java in either high school or early college. One example of a class would be AP Computer Science A, (advanced placement course in high school). It would also be very helpful to take a second programming course in college, as there are difficult topics discussed here in this guide. Below are some of the important topics that should be understood.

- Java Syntax
- Variables
- Conditionals and Switch Cases
- Loops (While, For, Do While)
- Exception Handling (try catch)
- Classes and Objects
- MVC (Model, View, Controller)

One of the best websites to help anyone learn and understand these topics is W3Schools. This is a website that has sections dedicated to each important topic. Also understand that programming takes practice and time to understand before it can be used for other purposes. Make sure to complete practice programs and learn how it works before moving into graphical user interface development.

Graphical user interfaces in Java are commonly used anymore, but it is still a very important concept to understand. The most important applications are created and used on the internet, but all of these concepts should be understood before moving onto something this complicated. In this tutorial, we will show the process of how to create a window and all of the features that can be added to it. This will be accomplished using Java's Swing library. There are other ways to accomplish this, which include JavaFX and Java AWT. We will not look at these for this example.



Java Tutorial

Learn Java

Java is a popular programming language.

Java is used to develop mobile apps, web apps, desktop apps, games and much more.

Examples in Each Chapter

Our "Try it Yourself" editor makes it easy to learn Java. You can edit Java code and view the result in your browser.

Example

Link: https://www.w3schools.com/java/default.asp

How to Get Started
To start building a graphical user interface (or programming in general), it is helpful to have an IDE (integrated development environment). This makes the development process easy and straightforward.

It seems simple, but anyone who wants to get started here needs to have the technical requirements to achieve this. Here is a list of some of the best computer brands to use to start working with Java and computer programming.

- Lenovo, Dell, Asus Laptops (Thinkpad, Etc.)
- Modern desktop computers that run Windows 10 or Windows 11 operating systems
- HP Desktop or Laptops
- Modern MacOS Desktop or Laptops

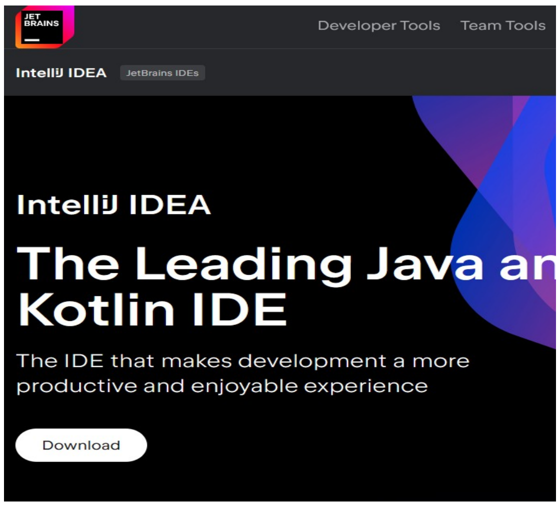
Here is one great laptop for all purposes. This is An HP Pavilion Plus. It comes with Windows 11 and 512 GB of storage. It is super fast and a good start for a new programmer.

WARNING: Please consider using a modern computer. Do not use a computer with outdated hardware or operating system. This can result in the loss of data from an old hard disk drive. An old operating system has security problems as well, which can result in loss of data.

Installing an IDE

For this example, we will be using a free integrated development environment. This is IntelliJ IDE Community Edition. This is the best IDE for Java, and there is never a need to pay for extra services when creating a simple graphic user interface.

Link: https://www.jetbrains.com/idea/



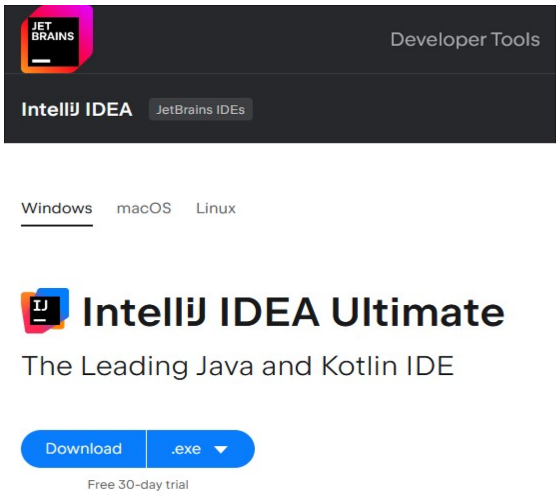
IntelliJ IDEA

The Leading Java an
Kotlin IDE

The IDE that makes development a more productive and enjoyable experience

Step 1: Click the Link

Use this link to download the IDE. Once there, click the white download button. This will bring you to all the different versions of the IDE.



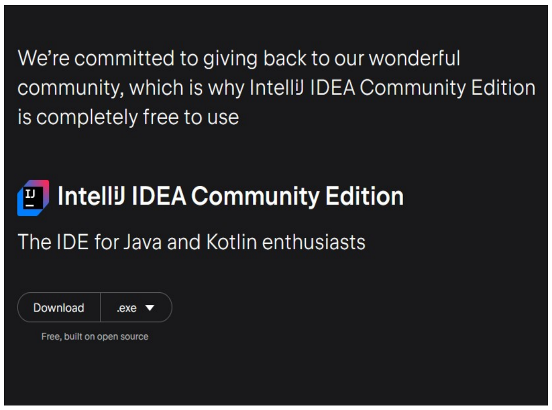
IntelliJ IDEA Ultimate

The Leading Java and Kotlin IDE

Free 30-day trial

Step 2: Scroll Down to Community Edition and Download

Make sure to get this version because it is free. The community edition gives anyone an advantage for computer programming. It is mainly for Java and Kotlin, a newer computer programming language similar to Java.



We're committed to giving back to our wonderful community, which is why IntelliJ IDEA Community Edition is completely free to use

IntelliJ IDEA Community Edition

The IDE for Java and Kotlin enthusiasts

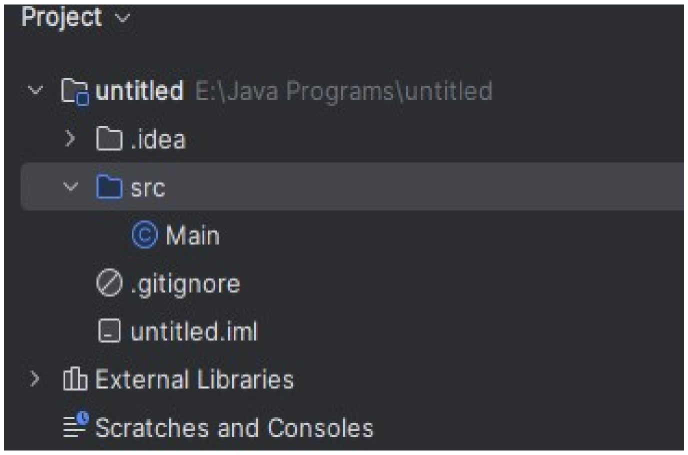
Free, built on open source

Step 3: Follow the Instructions of the Download

The download file on Windows is an exe file and dmg for MacOs. Follow the instructions and make sure to save the application to the desktop or taskbar for easy access. Make sure that this IDE is stored on a fast hard drive (specifically, a solid state drive with at least 250 GB). Most modern computers have this storage drive. After this, everything should be good to go.

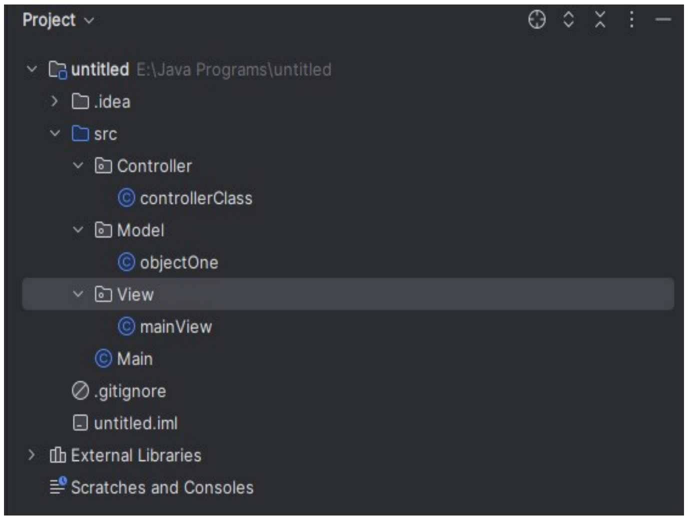
WARNING: Do not download the Ultimate edition, as it costs hundreds of dollars. It should only be used for more experienced developers or companies.

Creating the Graphic User Interface
Each step here is important for making the graphical user interface. The model we will be using for this guide is called the model view controller (MVC). This is helpful to organize Java code based on what it accomplishes. The controller is Java code that creates functionality for buttons, labels, etc. The model package stores code for objects created with Java. It is very important to understand object oriented programming going into this. Finally, there is the view, which stores all the forms and displays the windows to the user of the application. Here, we will look at the basic code for creating one of these.



```
Project
  untitled E:\Java Programs\untitled
    .idea
    src
      Main
    .gitignore
    untitled.iml
  External Libraries
  Scratches and Consoles
```

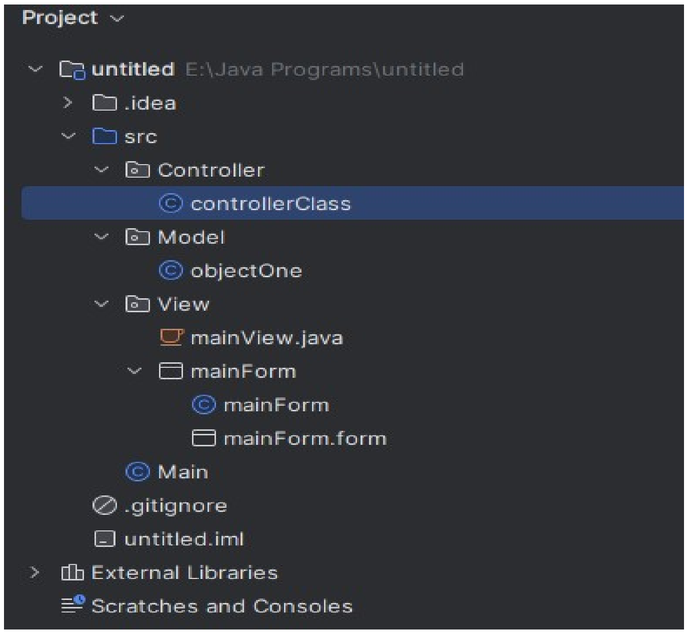
Step 1: Creating a Program and Packages
Make sure to create a Java program for this demonstration. Click the four bars and select "new" then "project". Start by creating the packages for the program to store the information that is needed. To do this, click on the four bars in the top right and select "new" then "package". Create three packages that look like the image to the right.



```
Project
  untitled E:\Java Programs\untitled
    .idea
    src
      Controller
        controllerClass
      Model
        objectOne
      View
        mainView
      Main
    .gitignore
    untitled.iml
  External Libraries
  Scratches and Consoles
```

Step 2: Create Classes for a GUI
Each package should contain one Java class. To create a new Java class, click the four bars on the top right and click "new" and then "Java class". Make sure that you right click on the package that you want to insert the Java class.

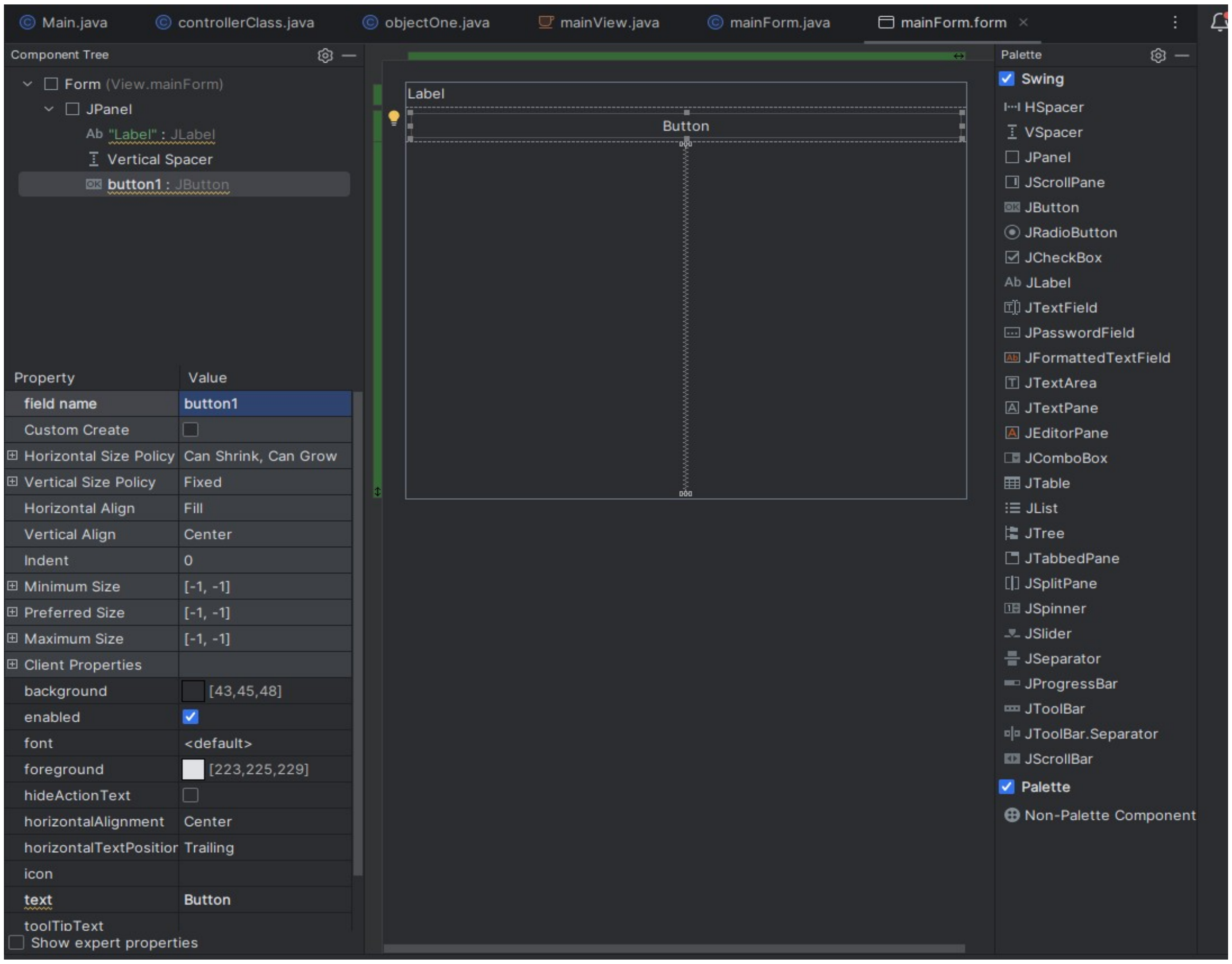
Step 3: Creating the Form.
Make sure to create the Java class that contains the main form. This is more complicated that just a Java class. To do this, click on the four bars on the top right. Select "new" then "Swing UI Designer" then "GUI Form". Make sure that this is contained in the view package.



```
Project
  untitled E:\Java Programs\untitled
    .idea
    src
      Controller
        controllerClass
      Model
        objectOne
      View
        mainView.java
        mainForm
          mainForm
          mainForm.form
      Main
    .gitignore
    untitled.iml
  External Libraries
  Scratches and Consoles
```

Finalizing the Graphic User Interface

Here, you should see the GUI designer. It is as easy as dragging and dropping. For this example, we will only use a label and a button. Drag and drop a label and a button to the top. It is quite easy to use. Once finished, you should see the screen below. For this example, the properties tab is not important. There is only one step that is needed. Make sure to double click on where it says JPanel and change the field name to mainForm. The program will not work if you do not do this.

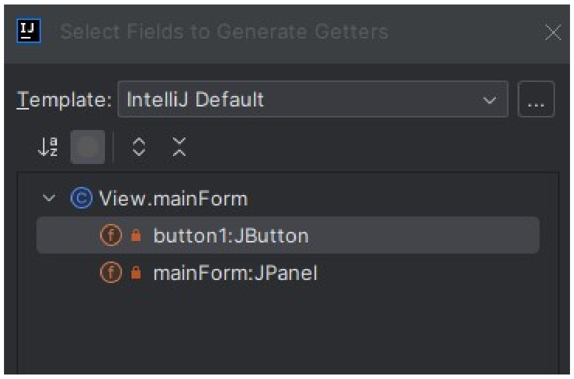


Property	Value
field name	button1
Custom Create	☐
Horizontal Size Policy	Can Shrink, Can Grow
Vertical Size Policy	Fixed
Horizontal Align	Fill
Vertical Align	Center
Indent	0
Minimum Size	[-1, -1]
Preferred Size	[-1, -1]
Maximum Size	[-1, -1]
Client Properties	
background	[43,45,48]
enabled	✓
font	<default>
foreground	[223,225,229]
hideActionText	☐
horizontalAlignment	Center
horizontalTextPositior	Trailing
icon	
text	Button
toolTipText	

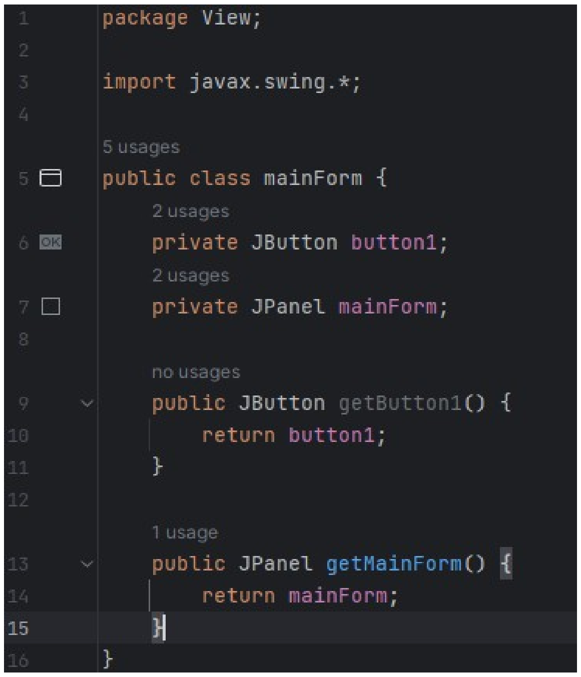
Warning: By not following the step above, the GUI will not display properly. Please make sure to do this.

Step 1: Create a Getter Method for the JPanel

Double click on the mainForm Java class and write a getter method for the JPanel. This is created so the form can be displayed to the user



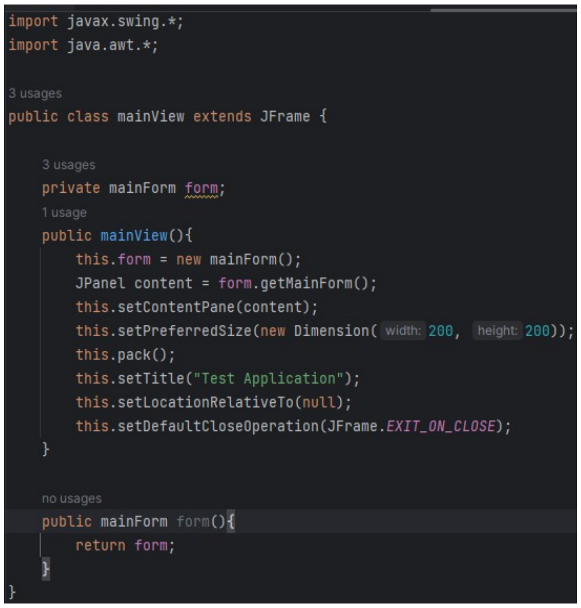
of the application. Click the four bars on the top right and select “code”, “generate”, “getter”, then select both the button and the mainForm objects that were created. For the mainForm Java class, your code should look like the code in the image to the right. These are extremely important to have, as this is needed for the mainView Java class.



```java
package View;

import javax.swing.*;

5 usages
public class mainForm {
    2 usages
    private JButton button1;
    2 usages
    private JPanel mainForm;

    no usages
    public JButton getButton1() {
        return button1;
    }

    1 usage
    public JPanel getMainForm() {
        return mainForm;
    }
}
```

Step 2: Code for mainView Class

The point of the mainView class is to define the size, position, and functionality of the JPanel window that is displayed to the user when running the application. For this example, please copy the code in the provided image. This is the only way the GUI will work.



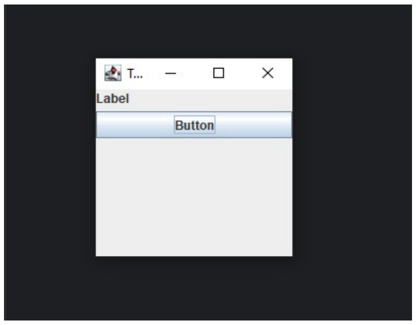
```java
import javax.swing.*;
import java.awt.*;

3 usages
public class mainView extends JFrame {

    3 usages
    private mainForm form;
    1 usage
    public mainView(){
        this.form = new mainForm();
        JPanel content = form.getMainForm();
        this.setContentPane(content);
        this.setPreferredSize(new Dimension( width: 200,  height: 200));
        this.pack();
        this.setTitle("Test Application");
        this.setLocationRelativeTo(null);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    no usages
    public mainForm form(){
        return form;
    }
}
```

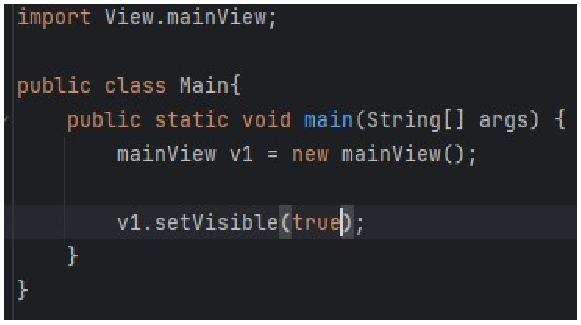
Step 3: Code for main Class

The main class should have been created when creating the project. If there is not a main Java class, create it now. Then, make sure that this class looks like the third image provided on this page. After this, run the main method. The window should appear on the screen.

The GUI is Complete!

You are finished! The code and program should appear as a white window with a label and button. Be sure to explore all the different features of Java Swing and explore other Java libraries. There is so much to create with the amazing programming language.





```java
import View.mainView;

public class Main{
    public static void main(String[] args) {
        mainView v1 = new mainView();

        v1.setVisible(true);
    }
}
```

Works Cited

“Creating and Opening Forms | IntelliJ IDEA.” *IntelliJ IDEA Help*, www.jetbrains.com/help/idea/creating-and-opening-forms.html.
https://www.hp.com/us-en/shop/pdp/hp-pavilion-plus-laptop-14z-ey000-14-7y8v4av-1?jumpid=cs_con_nc_ns&utm_medium=cs&utm_source=ga&utm_campaign=HP-Store_US_All_CPS_All_AMD_Google_All_Smart-PLA_Bestseller&utm_content=sp&adid=&addisttype=xpla&7Y8V4AV_1&cq_src=google_ads&cq_cmp=20537729678&cq_con=&cq_term=&cq_med=pla&cq_plac=&cq_net=x&cq_pos=&cq_plt=gp&gad_source=1&gclid=CjwKCAjwoa2xBhACEiwA1sb1BNZAoNAPMrA2kfrw-x5R-Uac7Fllk-iMkjtF1hLnisBnnScn0bRolBoCEtoQAvD_BwE&gclsrc=aw.ds
https://www.w3schools.com/java/default.asp
https://www.jetbrains.com/idea/