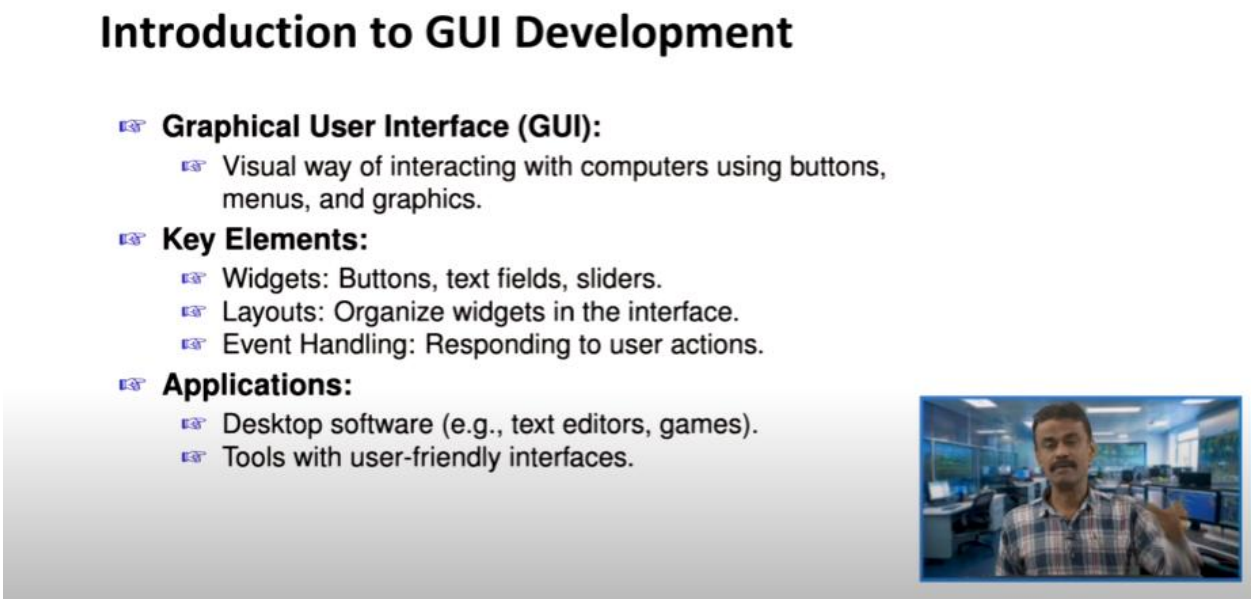
FUNDAMENTALS OF OBJECT ORIENTED PROGRAMMING

Lecture55

Lecture 55: GUI Development



Welcome to lecture number 55, Advanced Topics in Object Oriented Programming. So in this lecture, I will start with a graphical user interface GUI, right. So GUI, so far we have seen the output, right. Suppose you are writing let us say 2 plus 3, right, equal to 5 or addition of two numbers, right. So you see the output in the console.

So now is it possible to see in a graphical way or the visual way? So, now if you do the visual way of interacting with computers using buttons or menus and graphics you call it as the graphical user interface. So, the key elements in graphical user interface are the widgets, buttons right the text fields and sliders. So, we will see an example right. So, suppose you can see in some of the softwares you can see the menu right top down menu.

So, top down button and you can even print I mean you can even type few things you call it as a text field and then you will have a sliders right. So, like this you yourself can create now right. So, this is what we are going to study in this chapter and then after widgets there is a layouts right. So, this is organizing widgets in the interface then we have a event handling. So, responding to user's action.

So, these are all the main key elements, widgets, layouts and event handling. So, what are all the applications? So, you can have like desktop software. So, for example, text editors and games, right? The text editors, games, if you want to develop the software for, so let us say, I mean, you are working in a project and you want to create the graphical user interface.

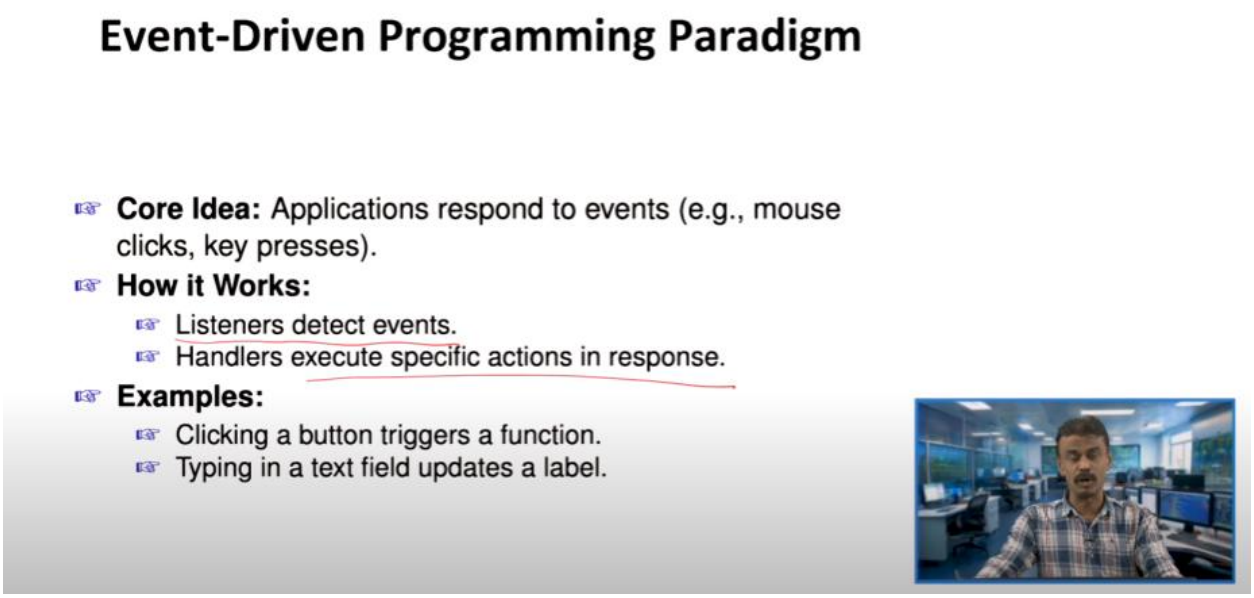
the elegant way for example the end user is giving let us say some input you are taking the input data right so you are displaying the message right once you run the application you are displaying a message right so enter the input you are entering the input right and then I mean in the back end this will do some process that means assume that it is calculating factorial of a given number right so you are only entering so it is displaying in a graphical way enter the number right you are entering let us say 7 So, what is 7 factorial? So, this is a simple example.

So, like this you can think of right. So, nowadays this is a AI ML world right. So, you can think of right. So, these are all the input the input even you can take from the file right. So, you are asking the user to enter the input right.

So, many AI ML applications or even quantum computing right applications text editors games So, these are all main applications and also you can have the tools with the user right. So, for the friendly interaction that means the user friendly interaction. So, the event driven programming paradigm right. So, the core idea is application respond to events for example, I mean I can use the mouse to give the input or I can do some clicks right with the mouse clicks.

Or I can have, let us say, pressing some keys. All right. So, for example, I mean, we know control C, control V kind of. Right. So, you yourself can do it in your program.

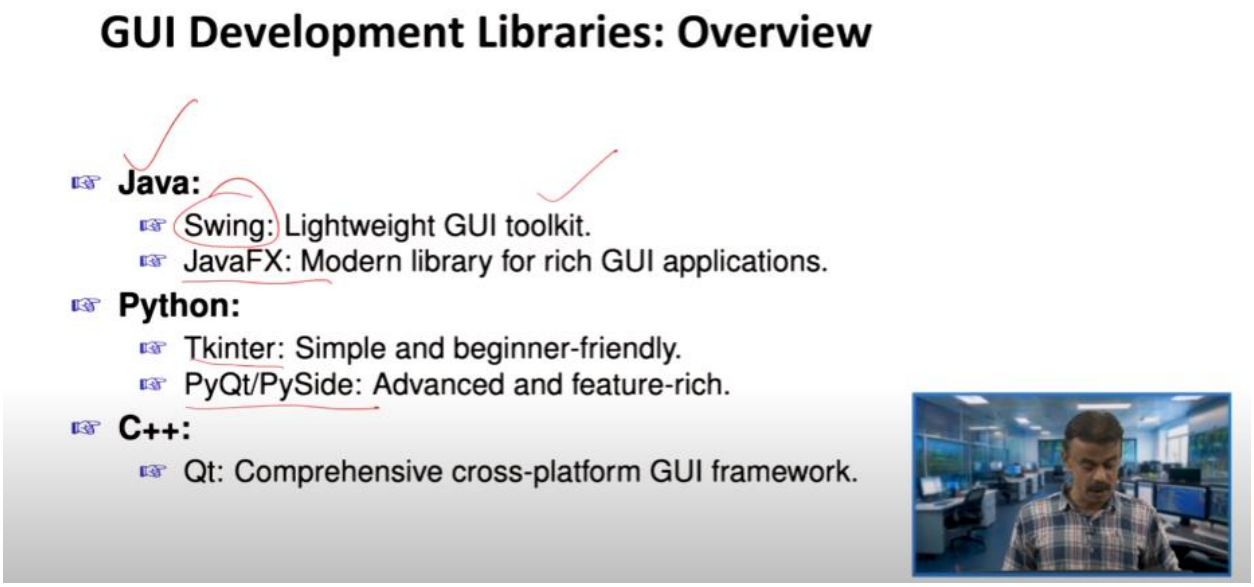
Right. So, how this is working? So, listeners detect events. This is we are going to see. Right.



So, this is the main thing for all the programs that we are going to see. Listeners detect events. And then handlers execute specific actions in response. For example, I want to create a calculator. All right.

I want to create a login password environment. Right. So for all this, you need listeners to detect events. So you may use lambda expressions, and the handlers execute the specific actions. For example, you're clicking a certain event, or you're writing a program for equivalent control-C, control-V. Right.

So this works based on listeners and handlers. And yeah, as I said, clicking a button triggers a function. I gave an example: control-C, control-V kind of equivalent, right? And then typing in a text field updates the label, right? So you can, in the text field, I mean, you can type, right?



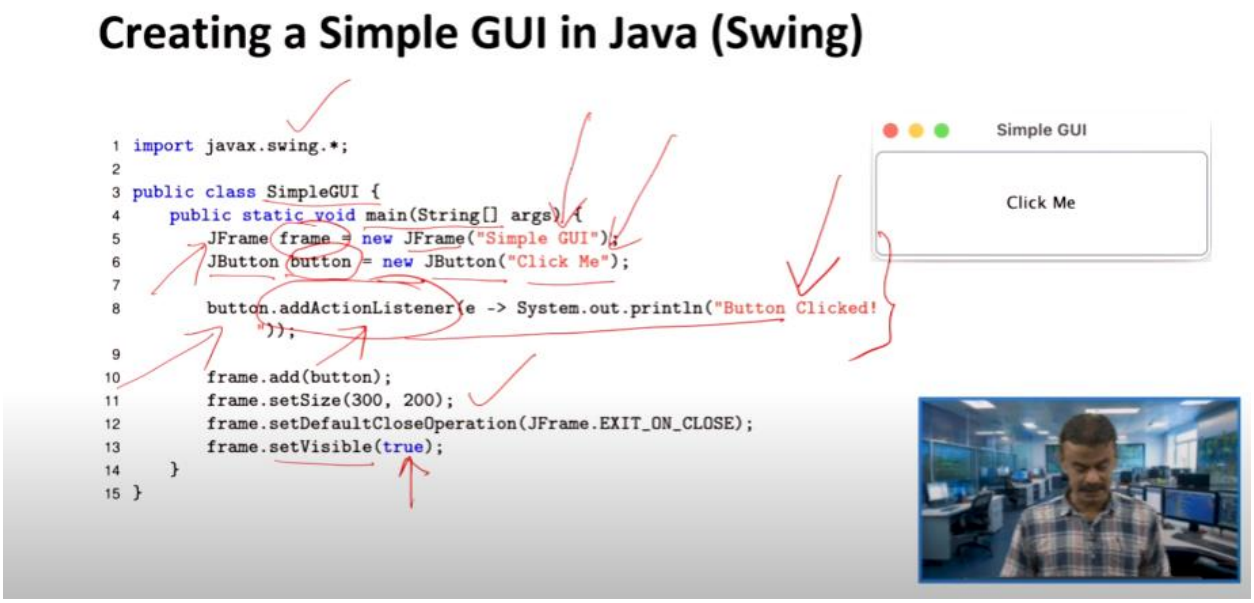
So, these are all the E1-driven programming paradigms. So, now, what are all the GUI development libraries? Java—if I take Java, I have Swing and JavaFX, right. So, when you have a lightweight graphical user interface toolkit, you have Swing, right. And for a modern library for rich graphical user interface applications, you have JavaFX, right.

Then, Python. So, in the case of Python, you have Tkinter, PyQT, and PySide. So, when you want something simple and user-friendly or beginner-friendly, we use Tkinter. For advanced features, you use PyQT or PySide. So, these are all the graphical user interface libraries in Python.

And in C++, you have Qt. The comprehensive cross-platform framework for graphical user interfaces in the case of C++. So, these are all the GUI development libraries in Java, Python, and C++. So now, with these basics, we can explore interesting graphical user interfaces. So, I'll start with a simple GUI in Java.

So we have class SimpleGUI. This is a driver class, right? And you have the main program, right? So, you have the main program and here you have the object frame, right? So, you have the object frame, right?

So, object frame is under class Java frame, right? Here you have button that is an object under Java button, JButton, right? So, you are including as I said swing. Right. So you are including Java X swing.



So now let us say you have frame object. Right. So your frame object in line number five on the right hand side, you are allocating a memory where it is calling the constructor JFrame. And when it is calling a constructor, you are passing simple GUI. So, that means we are going to create sooner or later with the help of the frame object the frame object will call several methods right.

So, then the simple GUI will be printed right. So, in a simple GUI will be there in the graphical user interface I mean the title kind of the frame you call it as a frame the frame is simple GUI it will be printed. And similarly, in the case of the button, click me will be printed, right? So, that means in the right hand side, you have the memory allocation, the dynamic memory allocation. And here you have the constructor, right?

So, the button there is click me. And here you are having this lambda expression, right? So, the button, the object button is invoking add action listener, right? So, this adds an event listener, that is called the event listener. So this is adding the event listener to the button.

So this event listener is implemented using this is called the lambda expression. So which is a concise way to define behavior of the button click. So when you click the

button or when the button is clicked, assume that you are clicking the button, then this button click will be displayed. are printed right so this button click will be printed so this is the lambda expression correct so this is where i talked about the even listener right so the even listener addActionListener right so this is called the even listener i i in fact told in the one of the slides right yes listener detect events how it works right so this here in this case

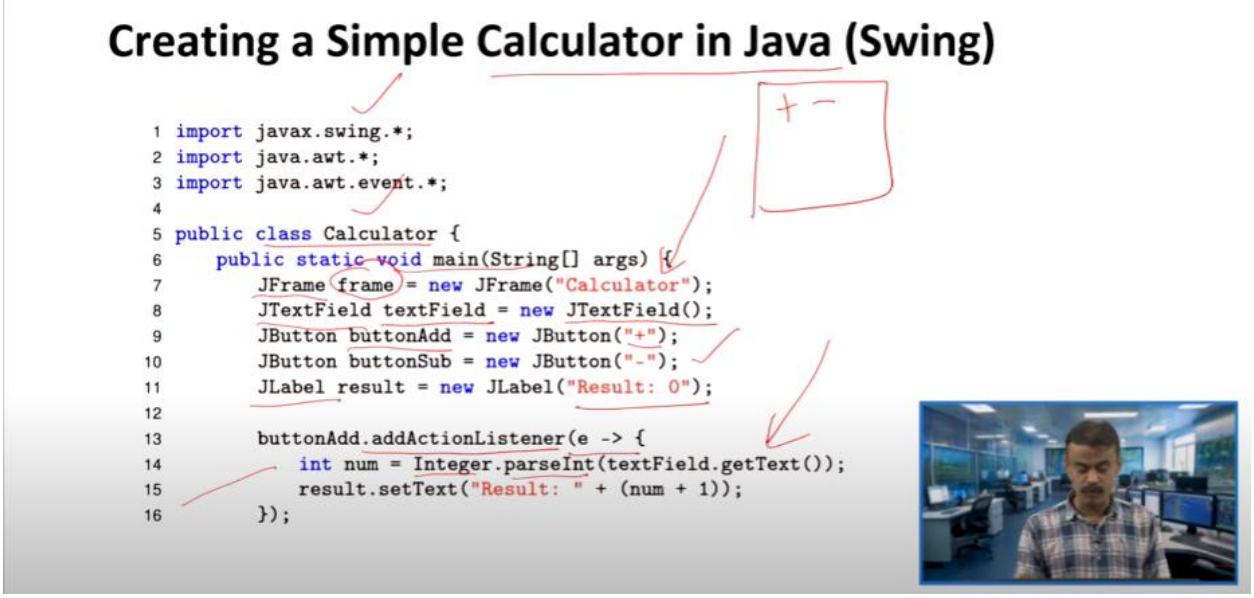
The line number 8, you call it as the event listener, right? So, rest of them are very trivial. Now, the object frame is invoking add function, right? Add method, right? So, you are passing button and you are setting a size.

Height is 200, width is 300, right? That is the meaning of this. And frame is invoking setDefaultCloseOperation, right. So, once everything is over, this will be closed and setVisible, it is invoking set visible by pausing true, right. So, when you are running the code, right, you will get the output like this, right.

So, add, you are passing. button right when the button is clicked that will be taken care on this add action listener or you call it as a even listener set with this i mean to say set sizes 200 height and 300 width so when you run the code you will get the output like this right JFrame simple GUI is being printed in line number five and next you have click link so this visual information you will get this first time you are learning right so this is based on the swing maybe we can try to run this code in java because first time we are learning it is always better to run the code yes here you can see simple gui correct and click me

so when you click this that is what i said button click will be displayed right so that is what i said so here you have simple GUI whenever i click you can see button clicked right i am trying one more time See, you are getting the output, correct? So, this is what exactly I told. So, once you click me, right?

So, the button clicked, that means the add action listener or you call it as a even listener, whenever you are clicking the click me, right? So, whenever you are click me button, right? So, the button click will be displayed. In fact, we have seen this. right so you can see the output button click will be display I hope it is clear so you can enjoy the program you can write it and I mean do many things so that we will see



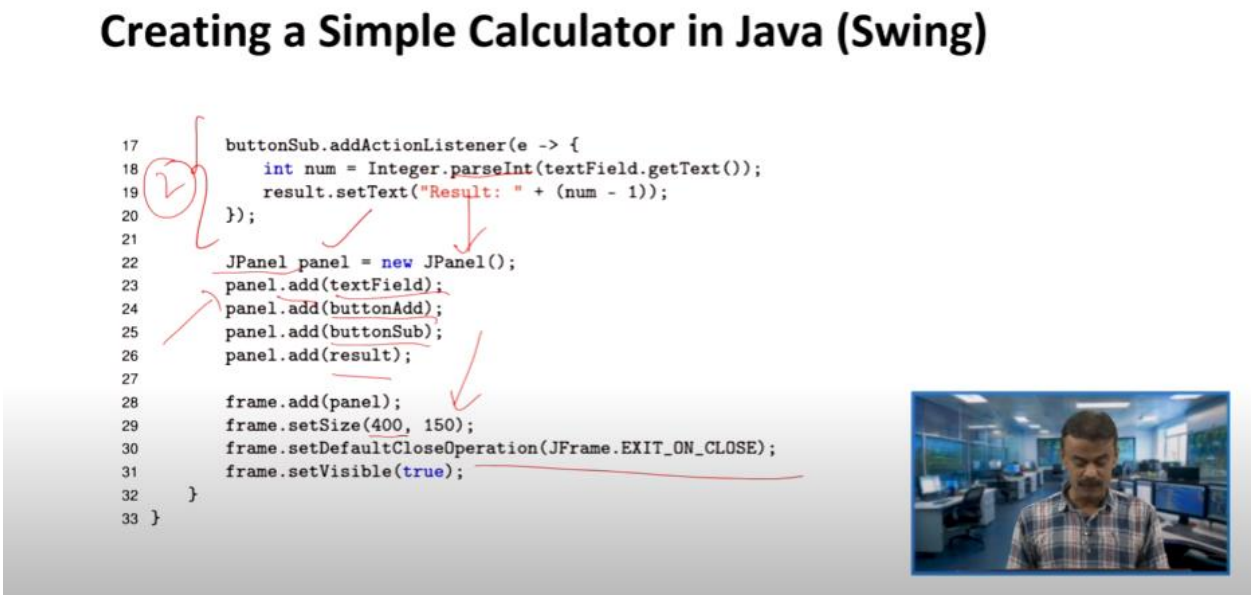
so here we have calculator simple calculator in java visualized one right so here I have a class calculator so you are adding swing awt and awt dot event importing so class calculator this is driver class and here I have the main So, as usual right I have frame which is an object under JFrame. So, here the calculator frame the calculator will be displayed like the previous program and here I have text field object under JTextField you will have a text field all right. So, with you are calling the default constructor JTextField and then button add you have plus right you are passing plus all right.

So, in the window, we will have plus, right. So, like this, it will be printing. It will be there, right. So, the JButton is calling the constructor JButton where you are passing plus and subtraction similarly, and then you have the result object under JLabel, right. So, it is printing result 0, your plus, minus, and result 0.

So, this is what we are going to see. So now, as usual, like the previous program, here we have the event listener, Right, so this is your event listener. Again, the lambda expression you are seeing over here, right? So, this is a lambda expression. In this lambda expression here, we have addActionListener. The button add is invoking addActionListener, right? So here, when I have this, we call it a lambda expression: e arrow and everything.

And here, if you look, the integer is invoking parseInt, right. So whenever you are typing some text, getText, right? The text field is invoking getText, right. So whenever you are typing some text, it will be converted into integer. parseInt converts it into integer, and

then this result, right? The object result. Which is invoking setText, right? Suppose I have some number.



So the number will be incremented by 1, right? Number plus 1. I have result number plus 1. And similarly, this is for the case of addition, right? Button add, addActionListener, right?

Button add. Yes, you have line number 9. So line number 9 object, which is invoking the event listener. You call it addActionListener. So now again, button sub, which is in line number 10, another event, another event listener, or you call it addActionListener, it is invoking button sub.

So button sub object you are defining in line number 10. Right. So, now exactly same whatever text you are typing. So, that will be converted into integer and the result will be subtracted by 1, right. Result will be subtracted by 1, num minus 1, right.

So, that is what you are doing in this is also a lambda expression, right. So, we are seeing two lambda expression. One is this and the second one is this. So, now rest of them is not very difficult. So, you have a panel object under Jpanel class and this is your constructor.

right. You have the object panel which is invoking add right. So, you are adding text field, you are adding button add right, you are adding button subtraction right and then add result right. So, you are invoking this add like exactly what we have seen in the previous program. All these field will be there in the graphical user interface.

So, now frame is invoking add. So, panel will be there right. So, panel whose size is 150 height And width is 400. That is the meaning.

400 comma 150. 150 is height. 400 is the width. And then whatever the operation that you are doing, once it is over, it will be closed. Right.

So that is happening in line number 30 where frame is invoking setDefaultCloseOperation. right and then you have a setVisible right we want to see the visibility of the GUI by passing through right and then result by default it is 0 , right, that is what you are doing line number 11 right so my GUI look like this so what we can do we will run the code and we can see the output we will run the second code java calculator yes here you go we are seeing the output and then you see if I click plus yeah you can type whatever you want the number

3 plus 1, 4, right. So, you can put 5. 5 plus 1 is what? 6. So, 5 plus 1, 6.

Now, what can we do? Put 10. 10 minus 1, right. So, 10 minus—if I put minus, you know—10 minus is what? 10 minus 1 is what?

9. So, we are getting 9, okay. So, these kinds of simple programs you can start with, right. And you yourself can, right, create the GUI like this. Right, at one point in time, right? So once you are familiar, you can create the graphical user interface in Java, okay.

Creating a Color Changer GUI in Java (Swing)

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  public class ColorChanger {
6      public static void main(String[] args) {
7          JFrame frame = new JFrame("Color Changer");
8          JButton button = new JButton("Change Color");
9          JPanel panel = new JPanel();
10
11         button.addActionListener(e -> {
12             Color newColor = new Color(
13                 (int)(Math.random() * 255),
14                 (int)(Math.random() * 255),
15                 (int)(Math.random() * 255)
16             );
17             panel.setBackground(newColor);
18         });
```

We will see the third program: Color Changer, a graphical user interface in Java. All right, another interesting program. So, let us have So here, you have Swing AWT dot event star. This you are importing. So, public class ColorChanger, all right? This is a driver class. And I have the main program, right? So, frame is an object, like exactly what we had seen: frame object under JFrame. So, that means ColorChanger will be displayed in the frame, and the button you have is 'Change Color,' right?

Button is an object and JButton is the constructor where you are passing change color, right? Similarly, JFrame in the line number 7, JFrame was the constructor passing color changer. So, color changer will be displayed and inside you will have change color. Line number 9, you have object panel under the class java panel, right. You have a constructor over here.

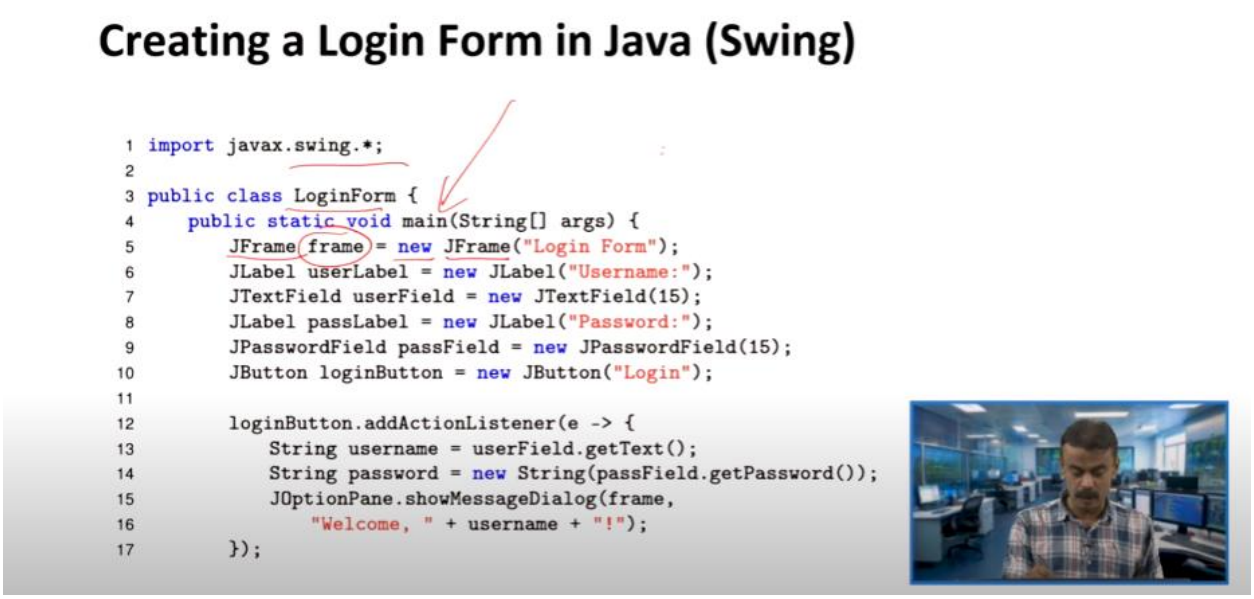
So, I will have a panel. So, now again you have the event listener. Here you have the event listener. The button is invoking the event listener. You call it as a addActionListener.

So, now you have an object newColor under class color, right, which is inbuilt. and here you have the constructor you are pausing RGB, red, green, blue, so, here what you are doing you are calling mathrand so mathrandom here is the closed interval 0.0 to open interval 1.0 this number will be generated math.random will generate the number between 0.0 the closed interval that means inclusive 1.0 exclusive and then when you multiply by 255 It's a random number, right?

Multiply by 255. So you will get a number between 0 and 254, right? That means the R value, for example, R is 100, G is 200, and B is 220. So this will form one color, right? That is why we have put this, right?

So some combination of colors will be generated, right? Then you set it as a background. setBackground means the panel is invoking setBackground. The program then invokes add panel, right? The frame invokes set size: 200 for height and 300 for width. Once you see it, you have to close it, meaning the visibility is set to true. You have a set default close operation in line number 22. Then, the visibility—if I run the code, I will get the output. This is the randomly generated color.

When I keep clicking 'Change Color,' we are going to test it. Then the color will change, right? This interesting program we will see in the Java compiler, right? We will go to the Java compiler, run the code, and you can see the color change, right? I am running Java. Here, if I click 'Change Color,' first the initial color appears. You are getting green now. Then you see another pink. The color is changing, right? When you run this code. So, it is very interesting that you are seeing the color change.



Maybe one more time if I do. So, right now green. Yeah. This is changing to some other color. Maybe one more time.

Alright. Now, it is a blue. Alright. So, change color. You can see that.

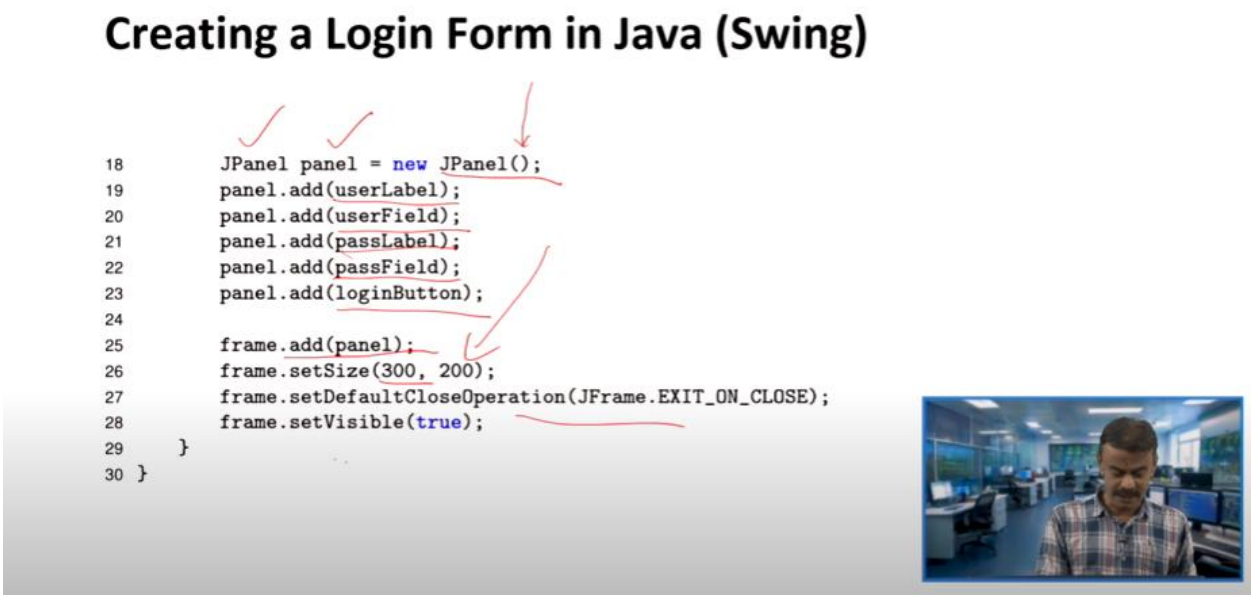
Alright. So, this is color changer GUI. We had seen this. Alright. So, now like any suppose you want to enter your email ID or let us say the social media.

Alright. If you want to enter the login and password. So, is it possible to create on your own, right, using Java Swing, right? So, we will see another interesting program, how we are going to have the login form. So, here you have Swing importing.

So, login form is a class that is a driver class. You have the main program, all right, object frame under Java frame, JFrame, that is a class. Here you have the dynamically allocated memory with the constructor JFrame, you have login form. So the frame will print login form. We are going to see the frame will be login form.

And you have userLabel under Java label. So constructor, username, right? So the username, the field you are giving 15 characters and the password, passLabel, object pass label under JLabel, you have password, right? You are giving 15 characters, right? And once you have, let us say login password, you will have a login, right?

Username, username, password, and there will be a login, right? So that will also be printed. So, the login button is under the Java button. Here you have a constructor, and you will have a login, right? So now here you have the event listener, right?



So this is your event listener with the addActionListener and username. Username is a string, right? Line number 13. The user field is invoking getText, right? So whatever text you are printing, that is your username.

And similarly, password, right? Similarly, password. The pass field is getting the password. So, that is also a string, right? And line number 15, your Java OptionPane, which is invoking showMessageDialog.

So, once you are logged in, it will show 'Welcome' followed by the username. For example, 'Bala IITR'. 'Welcome, Bala IITR.' It will display like this, right? So, here your lambda expression is on line number 17, all right.

So, we will further go into the class. So, you are still in the LoginForm class, as usual, right? You have an object panel under the class Java Panel, and you have a constructor JPanel, which has no arguments. We are adding a login button also, right? The usual one, right? So, that is why I have chosen this visualization, right? And on line number 25, the frame is invoking add panel. The height is 200, and the width is 300, as usual, with the close operation and setVisible operation. So, when I run the code, I will get the output like this, right? This is interesting. So, you have a login form, right?

That is what we have seen in line number 5. Right. Login form. And then you have a username. You can type a password.

You can type click login. So then you have to get like this. So whatever username I am typing, I have to get welcome the username. Right. Bala IITR.

Suppose assume that Bala IITR. Welcome Bala IITR. Right. So that we will get. So this interesting output or the code we will run in.

java compiler so when we run the code it's a login form yeah here you go so we type something bala iitr and then some password so login yeah you can see the see the output right welcome bala iitr press ok maybe you can try something else right let us say pradeep iitr and then some password right and when i login when i click login yes welcome pradeep iitr, we are getting the output right So, in fact, I mean you can think of, all right, formulating this graphical user interface. And we will see one more program, the counter application in Java, all right.

So, we will use the array and then we will do the similar one. So, now you are mostly familiar, right. So, swing, porting swing, counter app is your driver class and here you have a main program. I have a object frame under JFrame, class JFrame. So, JFrame you have a counter that means in the frame counter will be displayed and the counter label, right, it is upcasting with yeah not upcasting JLabel only right.

So, your class is JLabel and your constructor is JLabel, now count is equal to 0, count is 0, initial count colon 0 and then we are going to have increment all right. So, 0 will be incremented by 1, 1 will be incremented by 2 and so on all right and then you are having a reset also. So, increment button reset button. So, here you have an array counter. So, you are initializing to 0 first right.

So, initializing to 0 and then here you have event listener. So, what is your event listener? The increment button. So, that is invoking addActionListener right the event listener. So, the event listener what you are doing the count of 0 right the counter of 0 right.

Building a Counter Application in Java (Swing)

```
1  import javax.swing.*;
2
3  public class CounterApp {
4      public static void main(String[] args) {
5          JFrame frame = new JFrame("Counter");
6          JLabel counterLabel = new JLabel("Count: 0");
7          JButton incrementButton = new JButton("Increment");
8          JButton resetButton = new JButton("Reset");
9
10         final int[] counter = {0};
11         incrementButton.addActionListener(e -> {
12             counter[0]++;
13             counterLabel.setText("Count: " + counter[0]);
14         });
```

So, counter is an array. So, that is what we have put, and counter of 0 will be incremented by 1. All right, that is exactly what it will do. And counter label in line number 6, that object is invoking setText, and it is printing count with whatever the initial count is. Assume that initially it is 0; it will be incremented by 1 because of this, right. So, suppose it is 0 now; it will be 1. Suppose it is 1 now; it will be 2, and so on, right.

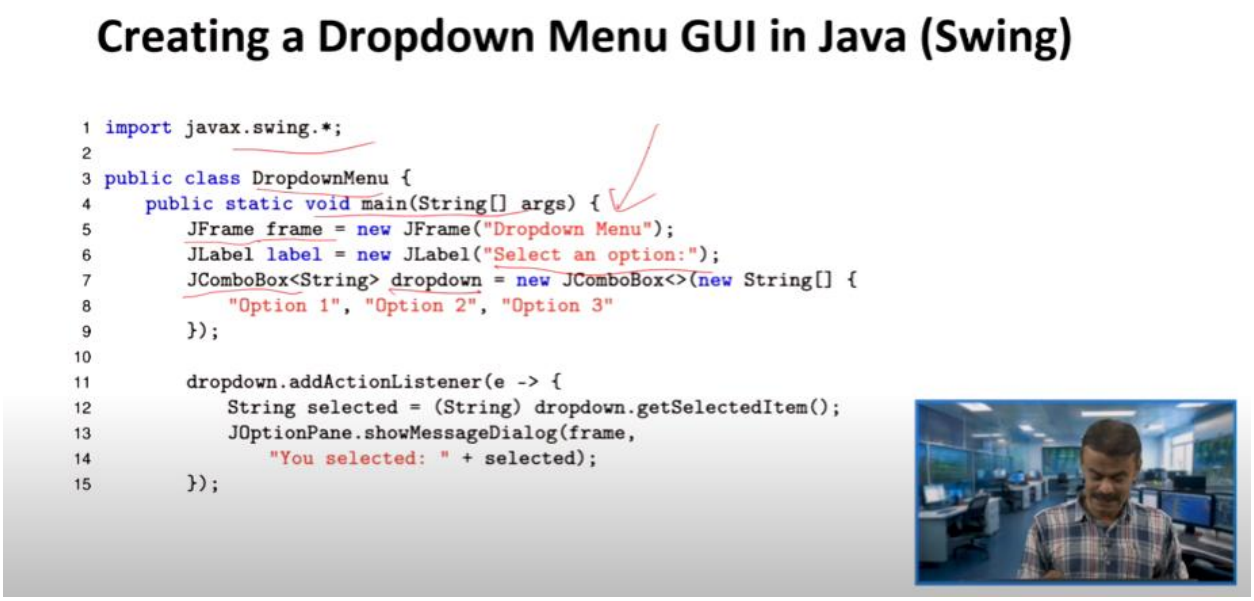
So, similarly, we can reset, right? Another event listener: reset. So, these two are lambda expressions, right? Anyway, we have seen them several times. So, this is one lambda expression, and the event listener is addActionListener for the increment button, and here you have the reset button, right. So, this is the second lambda expression, correct. So, here you have

and now we are resetting to 0, right? We are resetting to 0, and then we are printing that counter of 0. So, counter of 0 is 0 now. and panel is invoking add with the reset button, so increment button and reset button. So, frame is invoking add panel; the panel height is 150, and width is 300. As usual, you have the close operation and setVisible, passing true, right. So, the last lines, 27 and 28, we keep on seeing in all the programs, if you recall, right. So, when I run the code, I should get the output like this, right.

So, counter, right. So, that is what line number 5, counter is displayed, right. And counter z, count colon 0, count colon 0 and line number 6 displayed, increment reset, right. So, here you have increment reset. So now we will run this interesting code.

So we will go to the Java compiler counter app. So counter we are seeing count colon 0 increment reset. So now we will click increment 1 2 3 4 right. So you are getting everything. Now reset it will be resetting to 0 again put increment right maybe up to 10 we will go right.

So keep on incrementing one by one ok. So you can change your program suppose you want to do increment twice. increment thrice or I am giving an exercise. So, try to print the prime numbers every time next prime, next prime, next prime. So, this you can take it as an exercise.



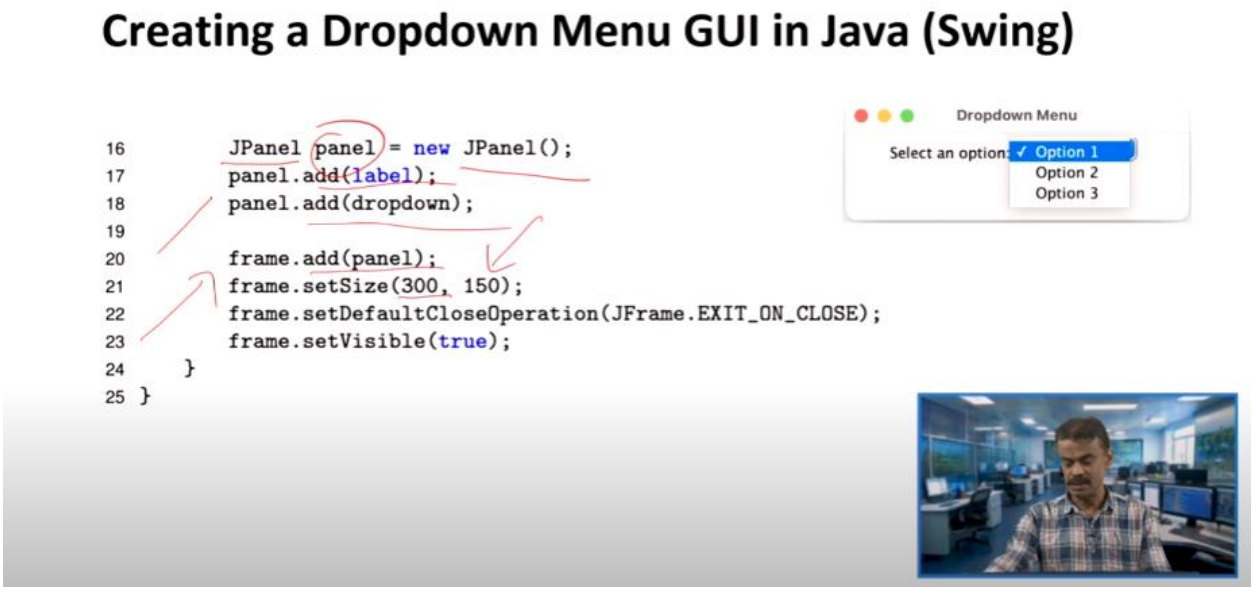
Initially first if you if you start with the first prime 2 right, then you write your code right in the back end and then what is the next prime, next prime, next prime. So, these are all the exercise that you can think of all right. So, we have seen several interesting programs with the help of graphical user interface maybe one more program we can see the drop down menu right so you we have used in several a software so is it possible that can you create the top right drop down menu right so as usual swing is being imported you have a drop down menu class and this is your main

right, so DropdownMenu is a driver class and here you go frame is an object under JFrame all right So, that means in that case you have dynamically allocated memory with the constructor right. So, you are having string the string contains this option 1, option 2, option 3 that means it will come as a drop down menu right. Suppose you are choosing country in some of the portal country. So, you will get then you are choosing

right that kind of menu you call it as option 1, option 2 and option 3 this we are going to see that.

right and then you have the event listener right so here you have drop down is invoking add action listener selected is an object under string right So that will be printed over here. You selected option 2. How this works? Show message dialog will invoke.

So you are passing the frame and then this has to be printed. Selected already there. Assume that I have selected option 2. So java option pan will invoke this show message dialog. So this is your lambda expression.



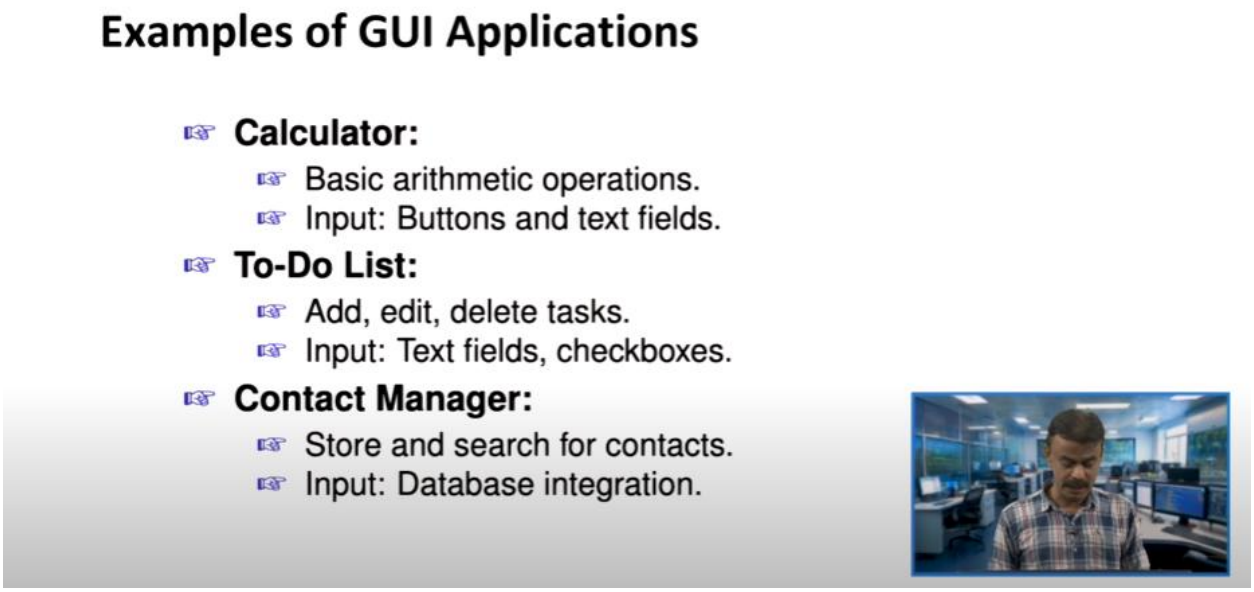
So I have one lambda expression here. Line number 11 to 15 is a lambda expression. right so then as usual panel is an object right under the class JPanel so here you have constructor default constructor the panel is invoking add label add dropdown under frame right which is adding panel panel height is 150 and width is 300 as usual close operation and set visible method right close operation and set default close operation and set visible methods right so set visible method you are passing true so when i am running the code i have to get the output like this right

so you are seeing drop down menu select an option so let us say what option we are selecting accordingly it will be printing right so it will print like this suppose assume that option one i selected it will print you selected option one line number 14 okay so this we will check in the program So we will run the code drop down so here you go select an

option right let us say I select option 2 right I click option 2 we will try one more also yes you selected option 2 is being printed. So let us see now I select option 3 you selected option 3 maybe one more left we select option 1 right you are getting you selected option 1 right. So these kind of interesting programs you can run right so this is what we have.

so some of the applications that we had seen i think four or five applications we had seen in gui so we started with the calculators like basic arithmetic operators in fact the plus minus if you look right so you can extend further right so these are all some of the gui applications so you can make a basic calculator right so the basic calculator that contains arithmetic operations right. So, you can see what are all the operations are there right.

So, in fact, you have the application in several operating system. So, here what you can do whatever we have done the first program we have done calculator maybe we have done plus and minus right. So, you can extend the program and then see the graphical user interface for calculator by adding the buttons and text fields right and what are all the to do list. So, what you can do you can do addition right addition right you can do editing right and the deletion tasks



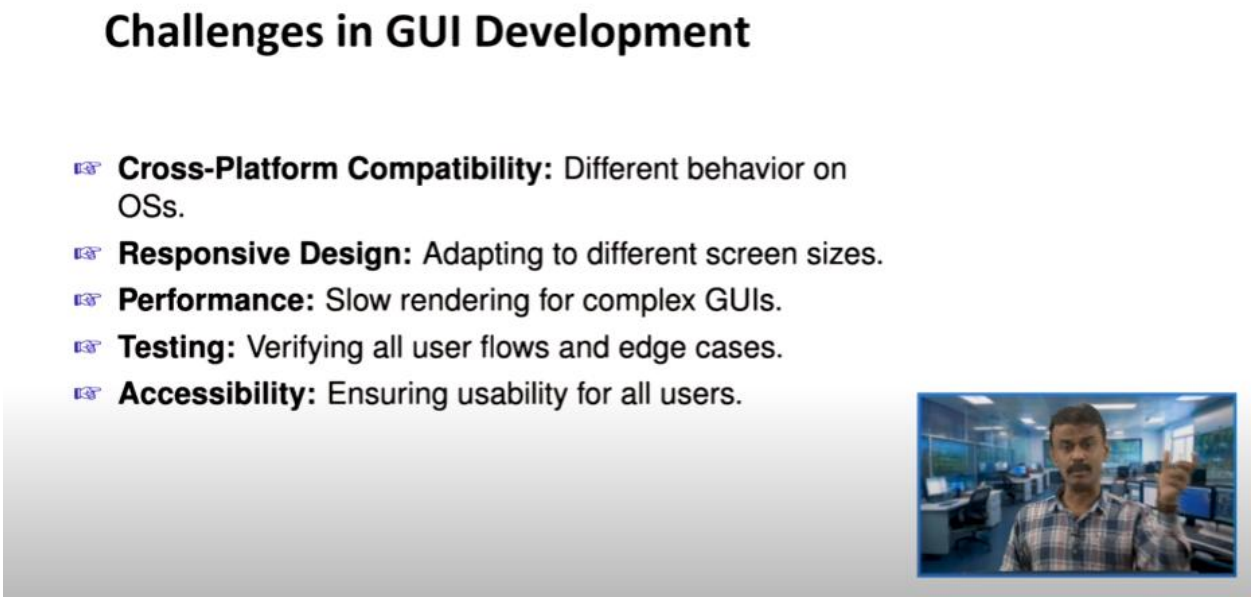
so these are all the things you can do so the input is the text field and check boxes and also contact manager right suppose you are the second point is to do list if you want to do something today right so that also you can develop all right and third one is contact

manager So the contact manager, you can store and search for contacts. All right. So you can store the person or you can store the information and then you can search.

All right. So these operations you can do. Right. With the help of graphical user interface. And also.

in this case the input should be database integration so whatever the input that you are giving for example one particular person you are giving certain information it has to be integrated for example first time you have added one phone number and let us assume after 10 days you are adding for that particular person so we are doing this integration so these things can be done with the help of gui in fact and many challenges we can face all right so the first one is the cross-platform compatibility right somebody may use one operating system and you are using another operating system so is it possible right to do the class across platform right so in that case it may have a different behavior on operating systems

responsive design right so adapting to different screen sizes right so somebody may use in the mobile and somebody may use in the right big screen desktop or laptop right so in that case so that is a very i mean difficult issue even if you are making a website so this problem you can i mean some people may ask so how it looks in the right desktop and how it looks like in the mobile right the mobile version they call it as a mobile version they call it as a desktop version all right so that is very challenging and performance all right so sometimes right since you are using a graphics the rendering problem right so these are basic problems in graphics so slow rendering can happen right



So, for complex graphical user interfaces, slow rendering can happen, and you have to verify all user flows and edge cases, right? So, testing and then accessibility—ensuring usability for all users—these are all the major challenges. One can face these when using a graphical user interface. So, with this, I am completing the entire chapter—the advanced concepts—I hope, right? So, you have learned so many things, right? Starting from network programming, etcetera, and I ended with the graphical user interface.

Thank you.