

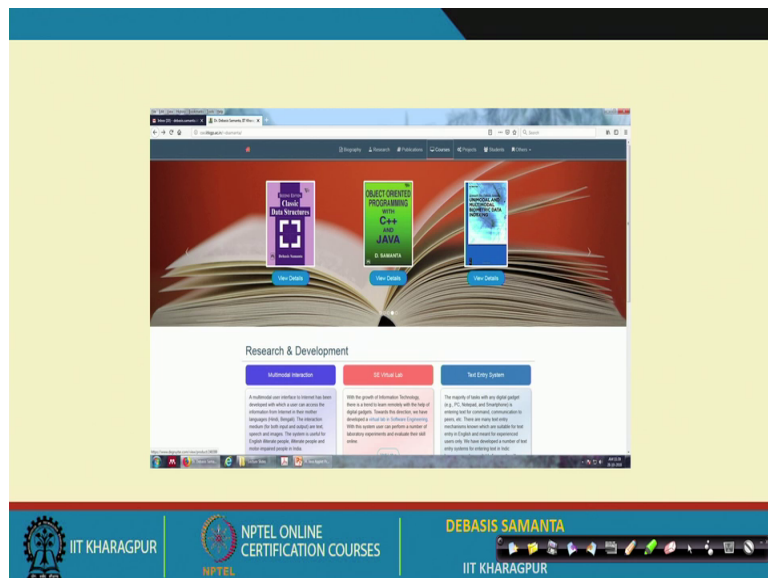
**Programming in Java**  
**Prof. Debasis Samanta**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture – 35**  
**Applet Programming – II**

We are discussing about Applet. We have some basic idea about, how to write an applet using Java syntax. So, there are many more things also to be learned so for that applet is concerned. So, today will discuss few more concepts and in the next module also we will discussed more advance concepts.

So, the applet is very important one what is call the concept in Java programming and it has many applications rather and so, we will exactly how the more organized way the applet can be designed.

(Refer Slide Time: 00:47)



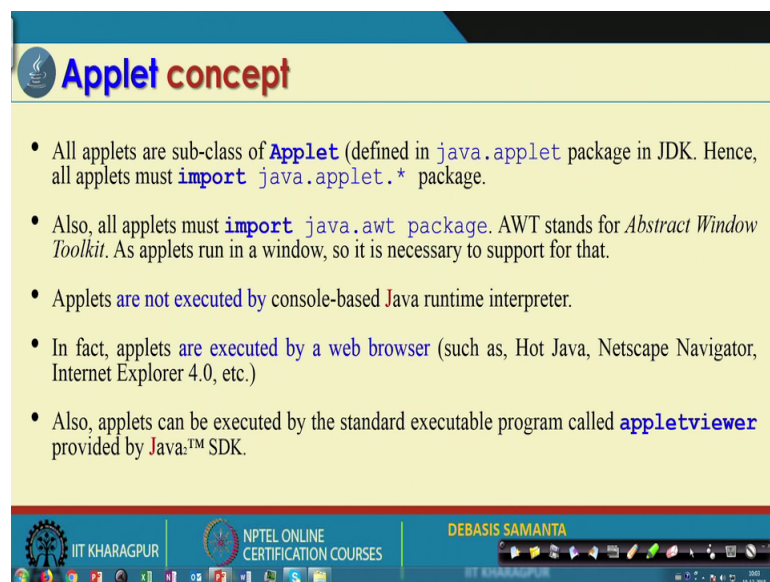
Now, before doing that we should have some basic idea about applet. So, basic idea means, what exactly an applet it is and what its usefulness. Actually, when Java introduced into the concept programming arena that time, it was a highly requirement that, how a programming language can supports internet programming. Now, so far the internet programming is concerned the basic requirement is that how a browser can browse web page. So, usually the web page written in HTML page, HTML what called the HyperText Markup Language.

Now, HTML can include many things. Here for an example this is the one figure that we have shown here this is basically web page, right. And then in this web if we see it contains many elements. For example, this is an image and this is also an image and there are some text box and there are may be some link like as you see here. So, this is the link those are the many links, this is basically menu we can say like this one.

So, there are many elements are there. Now, out of these many elements one elements may be an applet. So, applet maybe we can consider say suppose, this is also an applet look like ok. So, this applet has its own contents and its basically serve some purpose, it can if we click this applet or rather you use this applet to interact to the internet this kind of concept are there.

So, basically the main purpose of the applet is for internet programming and for this internet programming, we have to create an HTML page; and to create the HTML page we may use some code to be executed; some programs to be executed and that code that program is called the applet program actually. So, essentially applet is for this purpose only.

(Refer Slide Time: 03:13)



**Applet concept**

- All applets are sub-class of **Applet** (defined in `java.applet` package in JDK. Hence, all applets must **import** `java.applet.*` package.
- Also, all applets must **import** `java.awt` package. AWT stands for *Abstract Window Toolkit*. As applets run in a window, so it is necessary to support for that.
- Applets **are not executed** by console-based Java runtime interpreter.
- In fact, applets **are executed** by a **web browser** (such as, Hot Java, Netscape Navigator, Internet Explorer 4.0, etc.)
- Also, applets can be executed by the standard executable program called **appletviewer** provided by Java™ SDK.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA

Now, so this is the idea about this applet. Now, for this applet programming let us see how Java favors us, what are the supports from the Java that we can have. Now, in Java API, there is one package with the package is defined in applet package. So, this java dot applet package we rather say and there is one class called applet class.

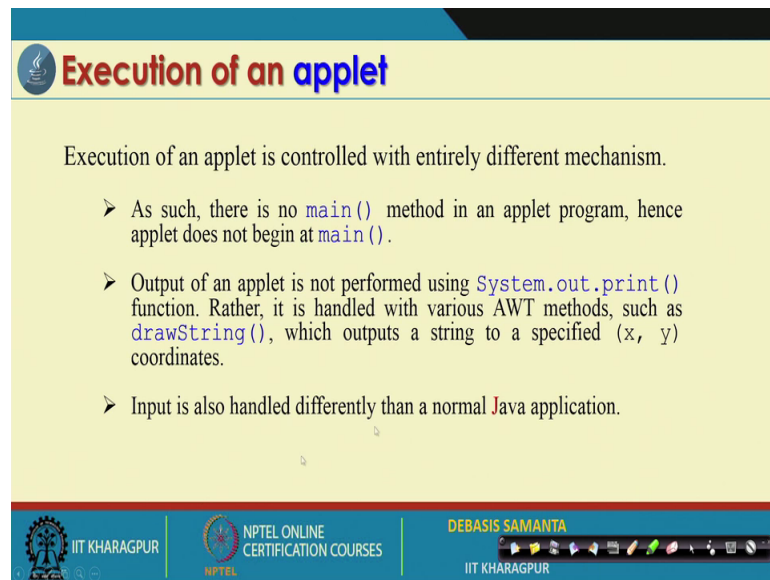
So, in java applet package there is a class called the applet class. So, we can use this applet class to create our applet program actually. Actually, basic idea about applet class is what? Exactly, applet class is an abstract class and it has certain predefined method, those methods are also the abstract methods.

So, if we have to create an applet essentially this will be basically inherited from this applet class. Inheritance in the sense here is that all the methods which are there in applet, that means abstract methods are to be overwritten here. Anyway, so this is the basic idea about writing and applet program. So, we have to import this package first, this package needs to be import whenever we have to write this applet and then once this package is imported, then we have to overwrite this method. And here another one thing is that, applets and the general program has different context in many sense.

The different in the sense that in case of applet as we see we need some separate program. For example, for the Java application, we need the Java command whereas, for applet we need the separate program called appletviewer. So, the Java runtime interpreter called the Java command is not actually applicable to the applet. And again another important thing is that, Java needs to be opened or the applet program needs to be executed with a help of browser, like say Hot Java, Netscape Navigator, Internet Explorer etcetera.

So, we need in order to run the applet, a browser or an appletviewer. So, this is the basic idea about, the applets and then obviously, the difference between the applets and application.

(Refer Slide Time: 05:37)



**Execution of an applet**

Execution of an applet is controlled with entirely different mechanism.

- As such, there is no `main()` method in an applet program, hence applet does not begin at `main()`.
- Output of an applet is not performed using `System.out.print()` function. Rather, it is handled with various AWT methods, such as `drawString()`, which outputs a string to a specified (x, y) coordinates.
- Input is also handled differently than a normal Java application.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

Now, how the applet is executed as we say there, in case of applet program there is no main method or rather there is no main class as it is there in Java application.

And again, the outputting form an applet is also totally different. So, for example, in case of Java application in order to produce an output on the screen or in some file we use say `System.out.print`, `print ln` like. But in case of applet, we need `drawString` method, which basically draw something. Essentially applet is a graphical windows or GUI based window themes, where we can draw many things. So, that is why is call the painting an applet.

(Refer Slide Time: 06:25)

**Applet window**

An applet is a **window-based** program.

- So called normal, console-based programs, we call them as application (also called Java core programs) are different than applets.
- As in window programs, applets are **event driven**. An applet waits until an event occurs. The AWT notifies the applet about an event by calling an event handler that has been provided by the applet. Once this happens, the applet must take appropriate action and then quickly return control to the AWT.
- Here, an event execution implies **the start of a thread**.
- User should initiate interaction with an applet. It is unlike a non-windowed program, where program prompts for an input, then read it and execute (either in single or multiple threads).

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA  
IIT KHARAGPUR

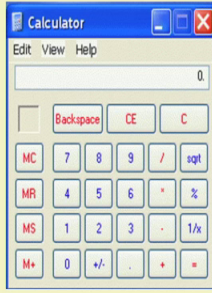
Anyway, regarding this painting and applet we will discuss many things later time. And one important thing is that apart from this applet being an window based program one important thing is that applet always an event driven program. So, event driven means, applet is ready to view something and it can interact with the user. So, it is basically the user interact with the internet, user can interact with a computer, but this is interaction through applets. So, this interaction is possible in applet using one another concept AWT, it is called Abstract Window Toolkits.




So, basically applet and AWT works together. So, we cannot impact for the actual applet programs AWT and then applet cannot be things in a separate way, both the things can be consider jointly. So, that is why we have to import the java dot applet dot star, that means, all the applet class facilities as well as, we have to import java dot awt dot star. That mean whatever the classes those are defined in AWT has to be there. Regarding AWT we will discuss in details once the applet is covered.

(Refer Slide Time: 07:39)

## Applet events

- In applets, **events are triggered** by a key space, mouse click, mouse drag, etc.
- More precisely, applet contains various controls, such as **buttons, text fields, checkboxes, labels, scroll bars**, etc.
- User **can interact** with these controls to generate events.



Anyways so, this is the idea about that applet is basically an event driven one program. Now, what is the event driven program that can be better understood using an example, let us have the quick look to this image here, and we see this is basically is an applet. And actually, if you see care carefully this kind of applets you see in your mobile or whatever it is there. And this basically is a calculator program, that means, if we can click 7 and then 8 then plus then 1 2. So, this means that the result will be there.

So, here you see if the click, I said the click. So, mouse click 7 and 8. So, if we click 7, then basically what will happen, the 7 will be selected or it basically appear here. Similarly, 8 8 after the 7 8, so 7 and 8 will be appear here, plus and then 1 and then 2 if we select, then also those will be there. And then say suppose, the plus this means that, it is an event. That means we want to add whatever the previous selected, plus the next will that will be going to select. So, these basically call the event driven.

So, here the idea is that, here the different components. These are the basically the different elements in these applets can be triggered and then once the applets that elements are triggered and event will be generated. This is the concept of event based programming. So, applet is basically an event base programming concept.

(Refer Slide Time: 09:17)

**Basic applet Structure**

```
import java.applet.*;
import java.awt.*;

public class <APPLET-NAME> extends Applet{
    public void init() {
        //Code for applet initialization.
    }
    public void start() {
        //This method is called after init()
        //Also called whenever applet is restarted.
        //This method contains code to start or resume execution.
    }
    public void stop() {
        //This method is called when the applet is stopped.
        //This method contains code to suspend the execution of applet.
    }
    public void paint() {
        //This method contains code to paint the window of an applet.
    }
    public void destroy() {
        //This method is called when the applet is terminated.
        // This is the last method in the execution of an applet.
        /*
        This method includes the code to perform shutdown activities.
        Usually, the codes are the methods defined in:
        java.awt package
        */
    }
}
```

Import Statements

init() method

start() method

stop() method

paint() method

destroy() method

Class declaration

IIT KHARAGPUR NPTEL DEBASIS SAMANTA IIT KHARAGPUR

Now, so we have learned about that applet it is basically the window program graphically user based windows program and event driven con program actually. So now, we have to learn about, how this graphical user interface face windows can be created and then how the event can be generated. So, these are the basic concepts, if we can learned it and then our concept of learning applet will be completed.

Now, before going to this things let us see, what is the structure an applet usually it can be. Now so, applet has the some essential components. Now, as I already told you that, these are the 2 packages to be included, whenever you have to do applet programming. And then the next thing is that, we have to create your own class, that is your program and it should be inherited from the applet class. So, our next syntax is basically this one public class this is your name I mean, user define name and then extents applet. So, that can be an extent that mean inherited program.

Now, next part the different methods. Now, again there are many methods are there. So, as we see there is a init method and then the start method and the stop method then paint method destroy method. So, these are the 5 standard methods that that usually occurs whenever a program are wants to design their own applet. Now, all these method again as I told you, these are the abstract method, which are defined in class applet, which is an abstract class. This means that we have to overwrite this method if we want to do our own applet program.

So, this basically if you want to write applet program this means that you have to implement the init methods, start methods, stop method, paint method and destroy method. Not necessarily that all methods are mandatory, all this methods are basically optional you can use it if you need it.

(Refer Slide Time: 11:31)

**Applet with blank methods**

```
import java.awt.*;
import java.applet.*;

public class AppletSkeleton extends Applet {
    public void init() { }
    public void start() { }
    public void stop() { }
    public void destroy() { }
    public void paint(Graphics g) { }
}

<html>
<body>
  <applet width="300" height="300" code="AppletSkeleton.class">
  </applet>
</body>
</html>
```

Applet Viewer

Applet

Applet started

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA

Now, so for, so these are methods optional for an example as we see here, in this example this is the one AppletSkeleton class that we have declare here. And here you see init, start, stop and destroy are these are the methods basically we have just overwritten, but the there is no code, is a null code actually so, but method as over written even if we do not write all these method also, that is also not a problem.

Anyway, so this methods as we see, if we run this applet as you know how to run this applet using an html applet code this is the applet code to run this applet and these basically, so an output here. So, this output if you see this output if you see here, this basically is a applet area and here there is no content and applet is just like a (Refer Time: 12:21) applet actually, it is not doing anything for us, but it is just viewed only.

So, appletviewer, browse it and then display it on your screen and showing that nothing is there, no event, no other content, no other elements etcetera. So, this way want I want to emphasis here is that the methods which we have used, they are mandatory; they are not mandatory they are optional, but preferably if you want to have them then you should implement them properly according to your own requirement.



(Refer Slide Time: 12:55)

**Order of invocation of Applet methods**

- These are abstract methods defined in abstract class `Applet` and they need to be overridden in applet programs. It is important to note the order in which the various methods are called.
- When an applet begins, the AWT calls the method in the sequence  
`init()` → `start()` → `paint()`
- When an applet is terminated, the following sequence of methods call takes place  
`stop()` → `destroy()`
- **Note:** All these methods are optional.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA

Now, next thing is that, whether all these methods should be written here and there in the applet program. Actually answer is yes, we can write it is not necessary to write `init` first then `start` then `paint` whatever it is, but we can write in any order. However the execution of all these methods follows a specific order. Here for example, as we have seen here, the applet will be executed if you define all these methods there then `init` first, once the `init` is executed then, `start` will be invoked and then `paint` in method will be invoked.

These are the 3 methods if it is defined in your applet program they should be executed in that order. And finally, the `stop` and then `destroy` if they are declared in your program they should be executed in these orders. So, order is very important as we learn here and then we will give an example so that we can understand that how the order is important here.

(Refer Slide Time: 13:53)

**Applet initialization and start : Methods**

The **init()** method is the first method to be called. This is where you should initialize an applet. This method is called only once during the run time of your applet.

The **start()** method is called after **init()**. It is also called to restart an applet after it has been stopped. Whereas **init()** is called once—the first time an applet is loaded—**start()** is called each time an applet's HTML document is displayed onscreen. So, if a user leaves a web page and comes back, the applet resumes execution at **start()**.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

Now ok, now let us see: what is the purpose of this method; init method it basically used to initialize the applet. Initialization means, if we want change the background color of the applet, style of the applet or if you want to do something prior to start the applet. So, it is basically initialization. Anyway, we will have the detail demonstration, detailed illustration of this method how they are used; init has very important many important applications in many applet design actually.

So, we will discuss in details whenever we will consider some case studies. And then finally the stop start method. So, after the init; that mean, init method is for initialization it will not do only the initialize the applet and then start is basically starting and; starting and execution of an applet. So, it is basically usually called after init if it is defined here and, in fact it is basically call to restart an applet whenever it has been stop. For example, you are browsing an web page where than applet is there, you just leave these web page and then come back. Whenever you leave this so, applet basically stop, and when you come back again the same applet started to execution.

So, it is basically the idea about this method. So, start method.

(Refer Slide Time: 15:19)

### Applet paint method

The `paint()` method is called each time the applet's output to be redrawn. This situation can occur for several reasons. For example, the window in which the applet is running may be overwritten by another window and then uncovered. Or the applet window may be minimized and then restored. `paint()` is also called when the applet begins execution. Whatever the cause, whenever the applet must redraw its output, `paint()` is called. The `paint()` method has one parameter of type `Graphics`. This parameter will contain the graphics context, which describes the graphics environment in which the applet is running. This context is used whenever output to the applet is required.

**paint()**

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And then finally, the paint now you know whatever the things you want to include if you want. So, applet also help allow a user to draw something on the screen using the pin or cursor whatever it is. So, using this pin paint; paint is in that sense very versatile one method with which user can interact with the screen interface and then draw it, even paint can also displays on screen from the user, the also it can be include many other graphical component into your applet. So, paint is also very important, init is very important, paint is also very important in that sense.

(Refer Slide Time: 16:03)

### Order of execution of methods : An example

```
import java.awt.*;
import java.applet.*;
public class Sample extends Applet{
    String msg;
    // set the foreground and background colors.
    public void init() {
        setBackground(Color.cyan);
        setForeground(Color.red);
        msg = "Inside init() --";
    }
    // Initialize the string to be displayed.
    public void start() {
        msg += " Inside start() --";
    }
    // Display msg in applet window.
    public void paint(Graphics g) {
        msg += " Inside paint( ).";
        g.drawString(msg, 10, 30);
    }
}
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

And here is an example; this example basically illustrates the execution of order, order of execution of init, start and paint method. Now, let us look at this program first as we see here we declare one class, that is the sample the name of our applet, it basically extends applet. And here, msg is basically the message that we want to use it, may be a string, and here you see init method. This is very important here as we see what we have done here, set background and these the color dot cyan. That means, the back ground of this applet is set as a cyan color and then foreground, the text if you want to display within this applet.

It basically selected as a red color. And the message is initialized as in that is the with the screen. So, message is initialize at this thing in side init. So, this is basically the init method. Now, here the start method, whenever the method starts is execution. It basically doing nothing only just message is basically updating that insides start. So, that mean, message is augmented with these plus this one. And then paint method is if we see; the paint method doing exactly again message inside paint and then g dot drawstring, this the message that already been write obtained in the store in this message fill, and these basically print the message.

Now, what is the message right now? So, whenever it is initialized. So, message is initially in inside init and then whenever start is executed, the same method will appended with these method. And finally, whenever the paint is called, the method will be again appended and the message become inside init, then insides start and then inside paint and finally it draw there.

Now, having this is the basically, i mean type of the methods that we have dis discussed here. If we run this, so init, start and then paint and automatically this way you can see the output. So, that means, it shows that all other execution is first init then, start and then the paint method. So, this way the execution will takes place in case of applet, whenever it starts its execution.

(Refer Slide Time: 18:29)

**Applet termination : Methods**

**stop()**

The **stop()** method is called when a web browser leaves the HTML document containing the applet—when it goes to another page, for example. When **stop()** is called, the applet is probably running. You should use **stop()** to suspend threads that do not need to run when the applet is not visible. One can restart them when **stop()** is called if the user returns to the page.

**destroy()**

The **destroy()** method is called when the environment determines that your applet needs to be removed completely from memory. At this point, you should free up any resources the applet may be using. The **stop()** method is always called before **destroy()**.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

Now, further termination there are 2 methods are used, stop and destroy method. So, as i told you, stop method means if you want to leave this browser and then actually the applet will stop its execution. So, this is the basically the idea of the stop. Now, stop is basically correctly stop its view, but not the key completely. So, to do it basically the destroy method is there. Once the destroy method is used in your applet program, and whenever this method is invoked.

This means, that it will completely remove from the memory and whenever you open the browse again we have to reload it, but in case if you do not use the destroy. Stop means it will be in the memory, but not completely remove from the memory or destroyed it. So, destroy is basically completely kill the execution of an applet.

(Refer Slide Time: 19:19)

The slide is titled "Use of stop(): An example". It contains the following Java code:

```
class MyWindowAdapter extends WindowAdapter {
    SampleFrame sampleFrame;
    public MyWindowAdapter(SampleFrame sampleFrame) {
        this.sampleFrame = sampleFrame;
    }
    public void windowClosing(WindowEvent we) {
        sampleFrame.setVisible(false);
    }
}

// Create frame window.
public class AppletFrame extends Applet {
    Frame f;
    public void init() {
        f = new SampleFrame("A Frame Window");
        f.setSize(250, 250);
        f.setVisible(true);
    }
    public void start() { f.setVisible(true); }
    public void stop() { f.setVisible(false); }
    public void paint(Graphics g) { g.drawString("This is in applet window", 10, 20); }
}
```

The code is annotated with a pink circle around the `f.setVisible(true);` line in the `init()` method and a pink arrow pointing from it to the `stop()` method. To the right of the code is a small window titled "Applet" with the text "This is in applet window" and "Applet started". At the bottom right, there is a video inset of a man in a suit, identified as DEBASIS SAMANTA. The slide footer includes logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

Now, here is an example which basically explains, how the stop method works here, but lot of things basically at this movement will assume. Now, here we discussed one program or class basically, this class we have discussed here although, we do not have any idea about, how this frame and everything. This frame is defined in a AWT, let us assume that we clear one class called the sample frame, which extends a frame.

Basically frame is basically one separate window, that can be included in the applet or a in a separately it can be viewed. So, frame is basically like this, it is just we can say one set up applet window like; it is an window we can say. And these frame we will basically there is an event concept it will takes place ok.

So, without knowing much about these things we assume that, there is a frame will be created and this frame will just wait for something's to happen. So, this basically code will do this. And this is the paint method, as we see in this class, this paint method whenever it is called, this paint method will draw a string, this is in the window in the title of the frame actually.

Now, so this is the first part of this program, and then the next part as we see, we have to define the method that we have declare my window. So, this basically the method that is declare. Again this method is basically will not doing this thing only whenever a applet is started its execution, it will basically display the window on the screen.

And then finally, this is basically the applet class program and this is the main program we can say, here we declare the init method. Init method create a frame, this is the f is the frame. And then this frame is initialized A frame window and then its set size; what is the size? For example, this; if this is the frame and this is the applet, this is the size of the frame. And A frame window it will basically show here like this, and here in this is the string basically paint within the applet, this is the; this is in the applet window.

Now, here in this program if we start, so whenever this applet is run by means of applet viewer. Then it basically display this one and as we display this one also display frame. So, both the things will be displayed on your screen. And then so, start means it will do this things, then it stop (Refer Time: 21:52) for you.

Now, if we click this one then what will happened, it basically invoke the stop method. So, stop method means it will kill it. So, it will stop, that means this frame will go out and, but this applet will remain it there. On the other hand, if we stop this applet, then what will happen? So, this is basically the entire things will go there. So, stopping a frame means only frame will go out, but stopping an when applet means the entire things will go out.

Now, again so this is basically the idea about that stop means it will be basically close the view of this one. So, this idea says that, this is the use of the stop method that is there in the applet. And the destroy method is basically, completely removing this one.

(Refer Slide Time: 22:35)

The slide is titled "Use of destroy(): An example" and contains two code snippets. The first snippet shows the class structure and lifecycle methods: init(), start(), stop(), and destroy(). The second snippet shows the addItem() method and the paint() method, which draws a rectangle and displays the string inside it. The slide also features a video feed of the presenter, Debasis Samanta, and logos for IIT Kharagpur and NPTEL Online Certification Courses.

```
import java.applet.Applet;
import java.awt.Graphics;

public class AppletLifecycle extends Applet{
    StringBuffer strBuffer;
    public void init() {
        strBuffer = new StringBuffer();
        addItem("initializing the apple ");
    }
    public void start() {
        addItem("starting the applet ");
    }
    public void stop() {
        addItem("stopping the applet ");
    }
    public void destroy() {
        addItem("unloading the applet.");
    }
}

void addItem(String word) {
    System.out.println(word);
    strBuffer.append(word);
    repaint();
}

public void paint(Graphics g) {
    // Draw a Rectangle around the applet's display area.
    g.drawRect(0, 0,getWidth() - 1,getHeight() - 1);
    // display the string inside the rectangle.
    g.drawString(strBuffer.toString(), 10, 20);
}

<<applet code="AppletLifecycle.class" width="300" height="200"></applet>
```



Now, here is an example again if we see; so, this is a the definition of an applet class and this is that complete applet class include. Now, in this applet class if we see; we the init method, the start method, stop method and destroy method. For the illustration we make all this method very simple. It basically, use one string and the string will be initialize by this one and then add item one method is basically called this all this methods basically.

Now, addItem method is defined here, is basically take an argument of the string and it basically print the string on the consol, that means, use as the java consol window and then it basically whatever the strings are there will append it and then basically the just similar the message concept, that is shown in the last example. And repaint means it basically, clean it and then paint it actually.

So, this is the repaint; repaint method is basically clear showing the output all the time. Now, so addItem here for the start staring the applet it will, whenever this method is called or the applet is executed start will be invoke automatically. And then starting the applet will be showing, and then so; so then the stop and then destroy will come into the applet.

Now, here in the paint method, so for the applet is concern; so it will basically draw rectangle and within this rectangle it will basically draw a string. Now, let us see, the output of this program, how it look like then you can understand.

(Refer Slide Time: 24:15)

### Use of destroy(): An example

```
import java.applet.Applet;
import java.awt.Graphics;

public class AppletLifecycle extends Applet{
    StringBuffer strBuffer;
    public void init() {
        strBuffer = new StringBuffer();
        addItem("initializing the apple ");
    }
    public void start() {
        addItem("starting the applet ");
    }
    public void stop() {
        addItem("stopping the applet ");
    }
    public void destroy() {
        addItem("unloading the applet ");
    }
    void addItem(String word) {
        System.out.println(word);
        strBuffer.append(word);
        repaint();
    }
    public void paint(Graphics g) {
        // Draw a Rectangle around the applet's display area.
        g.drawRect(0, 0, getWidth() - 1, getHeight() - 1);
        // Draw the rectangle.
        g.drawString(strBuffer.toString(), 10, 20);
    }
}
<applet width="300" height="200"></applet>
```

appletviewer AppletLifecycle.java  
initializing the apple  
starting the applet  
stopping the applet  
unloading the applet

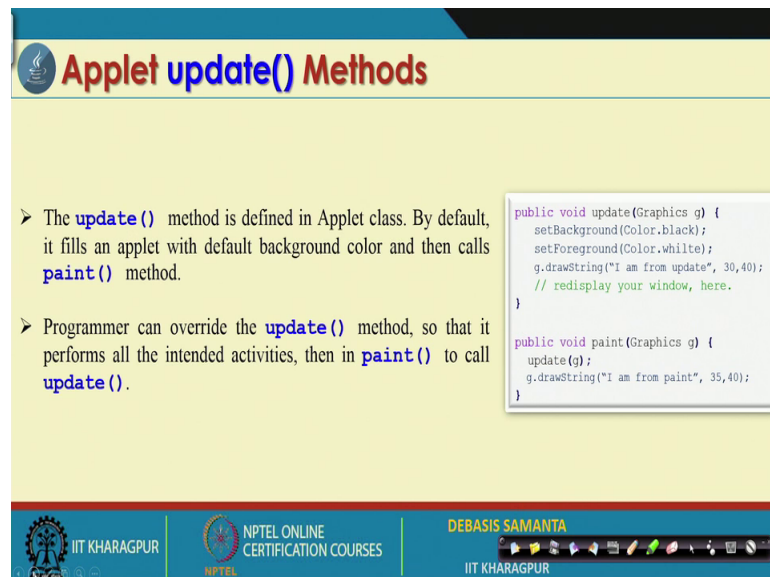
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR



So, is basically whenever we start this it basic initialize the applet and starting the applet, because of this addItem concept it is there go to the consol. And also applet will be there, which is not show here. Anyway, so applet will come there and applet will display the string on the applet area. And then whenever you stop it, then it will basically stop the applet and then unloading the applet means, completely clearing from the memory.

So, this way the destroy method will clear it. Now, these are the methods has many application, whenever we have to have a very complex the web base browsing application these are there.

(Refer Slide Time: 24:55)



**Applet update() Methods**

- The **update()** method is defined in Applet class. By default, it fills an applet with default background color and then calls **paint()** method.
- Programmer can override the **update()** method, so that it performs all the intended activities, then in **paint()** to call **update()**.

```
public void update(Graphics g) {
    setBackground(Color.black);
    setForeground(Color.white);
    g.drawString("I am from update", 30,40);
    // redisplay your window, here.
}

public void paint(Graphics g) {
    update(g);
    g.drawString("I am from paint", 35,40);
}
```

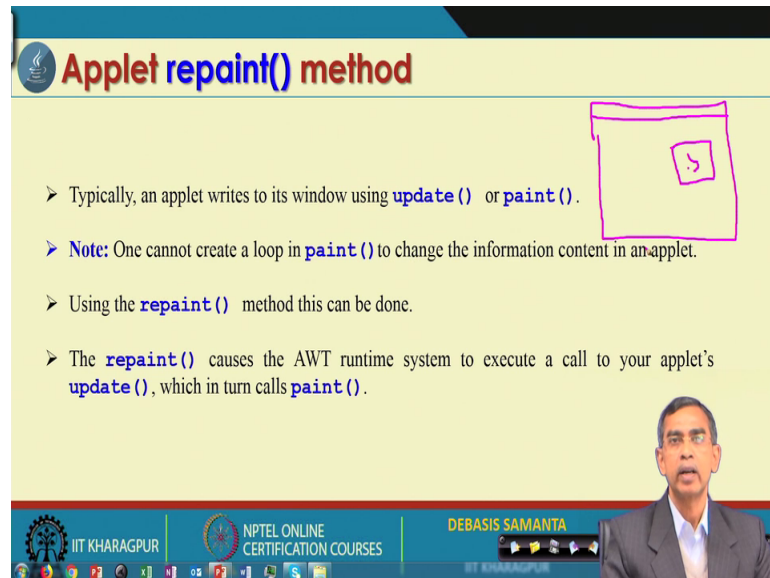
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA | IIT KHARAGPUR

Now, other than this update destroy and stop there is another two very important methods are there. There called the update and then paint method. So, update method is basically, just opposite it is on set of init method is basically reinitialization we can say updating the applet class. So, we can change it as an example as we see, if we define an update method and then we can change the background color, foreground color we can do something like this.

So, if we have initialization with some settings by means of updating we can resetting sort of things are there. And then these update method if we have do it, it cannot be invoked automatically. In the paint method you have to call it in order to update this one. So, it is basically update in the paint methods, so it will update it and then after the

update, whatever you can do in the paint method you will do it. So, update is basically setting changing the settings in your applet

(Refer Slide Time: 25:57)



**Applet repaint() method**

- Typically, an applet writes to its window using `update()` or `paint()`.
- **Note:** One cannot create a loop in `paint()` to change the information content in an applet.
- Using the `repaint()` method this can be done.
- The `repaint()` causes the AWT runtime system to execute a call to your applet's `update()`, which in turn calls `paint()`.

The slide includes a diagram of a rectangular window with a smaller square inside it, containing a right-pointing arrow. The bottom of the slide features a video feed of a man in a suit, along with logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

Now, repaint method is another example here, you want to update it. Now, for an example say suppose, this is your the entire applet view, these are entire applet. Now, a particular portion I want to change it, say suppose this portion I want to change it, but other portion will remain index. So, whenever you drawing some graphics or something using like paintbrush and like this one, I can select it; and the selected portion I can repaint it and then there something we can use it.

So, the repaint method is basically, a not the may be entire applets view or a particular portion of the applet view can be redrawn or repaint it. So, this is the idea about the repayment method. Now, I can have an example, so that we can understand, how this update and repaint method works it, there is very interesting program. This program is called Banner example.

(Refer Slide Time: 26:49)

## Applet repaint() methods

This version causes the entire window to be repainted and is the simplest version

```
void repaint() {...};
```

This version specifies a region that will be repainted

```
void repaint(int left, int top, int width, int height) {...};
```

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

Now, again before going to this repaint has the 3 different constructors. This is the one constructor without any arguments, so these are another constructor, it shows that which is the portion. So, it is basically left, top, width and height. So left, top is the one portion and then weight and then height. That means, this amount of area that we want to do the repaint.

(Refer Slide Time: 27:21)

## Applet repaint() Methods

```
void repaint(long maxDelay);
```

Here, **maxDelay** specifies the maximum number of milliseconds that can elapse before **update()** is called.

```
void repaint(long maxDelay, int x, int y, int width, int height);
```

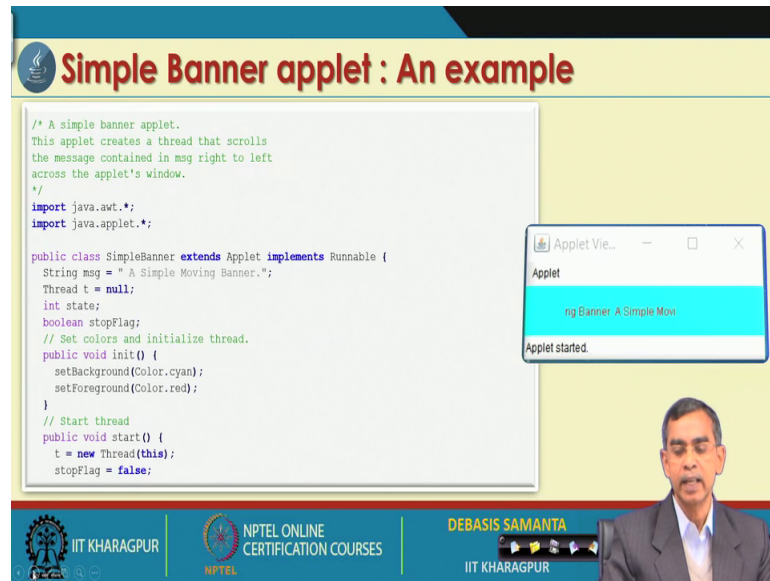
**Note:** If the time elapses before **update()** can be called, it isn't called. There's no return value or exception thrown, so you must be careful.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES DEBASIS SAMANTA IIT KHARAGPUR

And then another constructor, it is basically with some delay, that means repaint and update should be used together. So, once the repaint is there, then the following by the

update is there. So, a time delay we can specify, repaint long maxDelay it indicates that, what is the maximum number of time in milliseconds, that is should be that applet should wait before the update is called. So, it is basically, now if you do immediately the effect cannot be seen. So, that is why some delay should be then repaint. So, that some view, some impression can be obtained here.

(Refer Slide Time: 27:55)



The slide features a title "Simple Banner applet : An example" in red text. Below the title is a code editor window containing the following Java code:

```
/* A simple banner applet.
This applet creates a thread that scrolls
the message contained in msg right to left
across the applet's window.
*/
import java.awt.*;
import java.applet.*;

public class SimpleBanner extends Applet implements Runnable {
    String msg = " A Simple Moving Banner.";
    Thread t = null;
    int state;
    boolean stopFlag;
    // Set colors and initialize thread.
    public void init() {
        setBackground(Color.cyan);
        setForeground(Color.red);
    }
    // Start thread
    public void start() {
        t = new Thread(this);
        stopFlag = false;
    }
}
```

To the right of the code is a screenshot of a Java Applet Viewer window titled "Applet Vie...". The window shows a cyan background with the text "ng Banner: A Simple Mov" in red, which is partially cut off. Below the text, it says "Applet started." In the bottom right corner of the slide, there is a small video feed of a man, identified as Debasis Samanta, with the text "DEBASIS SAMANTA" and "IIT KHARAGPUR" next to it.

Now, this can be better understand with an example, it is call the banner example the idea it is there. So, suppose this is the applet, it can be viewed here but this is fine. We can run it; once you run it, then you will see this basically the banner will scroll it from right to left. So, it will change there. So, simple moving banner it will go there and then it will come here and then. So, base; basically this text will move. So, that is the banner text we can say.

Now, so do this things here you see we cannot simply use the paint and then redraw string. So, it is the static paint, but here actually painting will be always update, repaint and then display; update, repaint and display; update, repaint and display is a basically in a iterative method. So, that it will appear that this text is moving.

So, that can be understood by means a simple program here let us have a quick look of the program. First we declare one applet, this is the SimpleBanner extends Applet and it require set to be executed. So, we implement Runnable. So, it is a (Refer Time: 29:00) program also. So, applet combined you threading, both the things together here. Now

here, so a thread class is created t is the thread class and this is the state int. Now, so this is one boolean variables stopFlag.

Now, here the init method that we have declare; init methods is doing nothing is basically changing the background and foreground. So, as the background foreground we change like this one. And then here is basically the start method that we have declare here. In the start method we create the thread of this own applet.

So, this thread means, it will run independently until you stop it like. So, this basically, defining the state thread is basically created, which is the thread means these are thread of this applet, so applet is thread. And then stopFlag is false, means it is run. So, if we click some like, so this one stopFlag will be true line.

(Refer Slide Time: 30:05)

```
t.start();
}
// Set up point for the thread that runs the banner.
public void run() {
    char ch;
    // Display banner
    for( ; ; ) {
        try {
            repaint();
            Thread.sleep(250);
            ch = msg.charAt(0);
            msg = msg.substring(1, msg.length());
            msg += ch;
            if(stopFlag)
                break;
        }
        catch (InterruptedException e) { }
    }
}
// Pause the banner.
public void stop() {
    stopFlag = true;
}
```

So, this is the first part of the applet and then; then the next stage basically, the next portion is basically run as you know this is the thread program. So, thread should be accompanied with say run method. This is the run method here and t dot start you know; you see this start is basically starting the thread not that the applet start method. So, that is the totally different as you have implements runnable. So, start will automatically; that mean, we start the execution of thread. So, in the start method of the applet, we start the thread.

Then run method if you see, this is the more intelligent one code actually here. This run method what it will do is basically, here initially it will repaint, whatever repaint means initialize repaint is there, then thread will sleep for 250, because some weight to be there. So, that user cannot, otherwise if you constantly change you cannot find that change right.

So, delay it is there. So, this is the sleep means, the thread will just wait without doing anything and then the character will be corrected from the message; that means, fast corrector from the strings of the characters and then message it will basically next is the message will be obtained by competing the substring of the next corrector of 1 to the maximum 1.

So, basically each time, say suppose welcome; so first character is wel then the next substring basically, after proponing w so e l c o m e like this. So, this will basically the run method as the thread will execute. So, it will go on, one by one, one by one. So, it will take 1 character at a time from the front 1, and then get the substring, and then substring will be displayed the next and then again it basically update it and repaint it.

So, this way update and repaint will be go on go on and the idea it is there it first is A and then A will move here, then simple moving will come here, then this will come here and then other portion will be there and this way that that applet will start moving. So, if we run it and then we can see this actually applet gives a life to this and then is a beautiful on program that you can check it. Anyway, whenever we will give the demonstration, we will show its execution. Then stop here, so whenever stop means say, suppose user select this one. So, the stop will be invoke and then stopFlag is true then thread will stop and then the applet that the banner will stop its execution.

Anyway, so basic idea that different methods, which are there in an applet like init, start, stop then paint, update, repaint all this methods we have try to give an idea about, how these methods are there. And whenever you run different application, different program you can innovate also, different way of viewing the different applet you can see. And more over, all those things also will come into great effect whenever we develop a calculator. So, will see exactly in our demonstration, how all those advance applet can be designed and then those methods will be used to do that.

(Refer Slide Time: 33:11)

The slide is titled "Simple Banner Applet : An example". It contains two code blocks and a screenshot of an applet window.

```
t = null;
}
// Display the banner.
public void paint(Graphics g) {
    g.drawString(msg, 50, 30);
}
}
```

```
<html>
<body>
  <applet width="300" height="50" code="SimpleBanner.class">
  </applet>
</body>
</html>
```

The screenshot shows a window titled "Applet Vie..." with a blue header and a white body. The body contains the text "Applet started" and "ng Banner A Simple Movi".

The footer of the slide includes the IIT KHARAGPUR logo, NPTEL ONLINE CERTIFICATION COURSES logo, and the name DEBASIS SAMANTA.

So, these are the methods we have discussed there and so fine. So, these are the methods last standard methods. Apart from these standard methods, there are many more method also can be any method also user can implement and add into the applet. And all those things, we will discuss whenever we will discuss more advance features about applet programming.

Thank you.