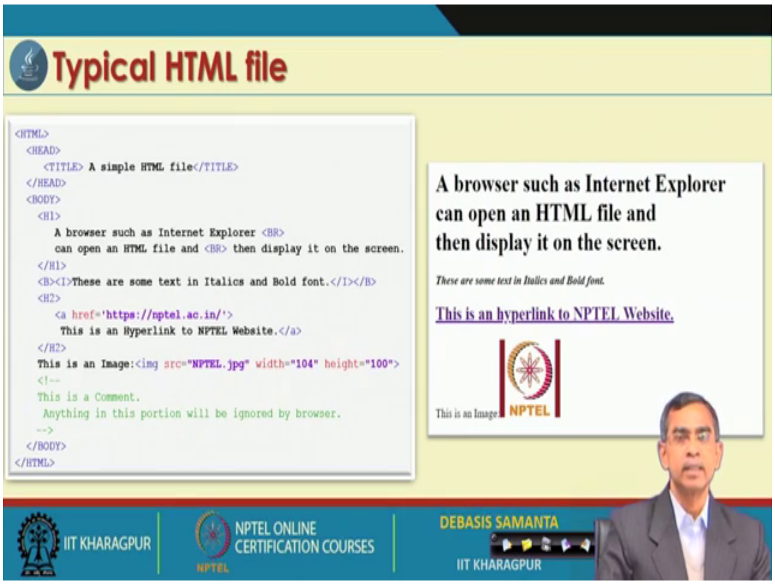
Programming in Java
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 36
Applet Programming – III

So, we are discussing about Applet. And in the last module, we have discussed the standard methods which can be included in an applet. Now, we will discussed about the HTML concept, as you know applet cannot be executed independently. So, in order to view an applet, we have to host this applet code into an HTML page. And then this HTML page will be browsed by means of an web browser like say Internet Explorer or Netscape navigator or the hot Java browser. And in Java JDK, there is a built-in viewer also called the applet viewer.

Now, whatever it is there, first the idea is that applet should be hosted by an HTML page embedded into an HTML page. So, in this module, we will see how this hosting is possible using HTML contact. Actually, one specific tag that is there in HTML; HTML in Hyper Text Markup Language is called the applet tag. So, we have to discuss about applet tag here.

(Refer Slide Time: 01:30)



Typical HTML file

```
<HTML>
  <HEAD>
    <TITLE> A simple HTML file</TITLE>
  </HEAD>
  <BODY>
    <H1>
      A browser such as Internet Explorer <BR>
      can open an HTML file and <BR> then display it on the screen.
    </H1>
    <B><I>These are some text in Italics and Bold font.</I></B>
    <H2>
      <a href='https://nptel.ac.in/'>
       This is an Hyperlink to NPTEL Website.</a>
    </H2>
    This is an Image:<img src="NPTEL.jpg" width="104" height="100">
    <!--
    This is a Comment.
     Anything in this portion will be ignored by browser.
    -->
  </BODY>
</HTML>
```

A browser such as Internet Explorer
can open an HTML file and
then display it on the screen.

These are some text in Italics and Bold font.

This is an hyperlink to NPTEL Website.

This is an Image:

Now, so applet tag is basically let us first see exactly how a typically an HTML page look like. Now, here you see HTML is a basically tag based as we said, now this is basically HTML, and this is the end. So, these are the tag, it is called the HTML tag that means, this indicates that this is the HTML page 1.

So, if this is the tag and then browser use it no issue, browser will see the blank page here, because it in this tag nothing is there. Now, the next that we can do is that head. So, head is basically we can write, what is the title that will be there. So, head is basically with one HTML browser will show it, here the browser view is not there. So, a simple HTML file if you use it, browser will show that.

And then within this body, we have a text is a header 1, header 1 is a very big font large font. So, this will display here in the header font. Now, you see there the different tag we are using, so this is the another tag B. So, B tag indicate that, it will display some font in a bold as well as I tag indicate that italics. So, this basically text will be displayed both in bold and italics form.

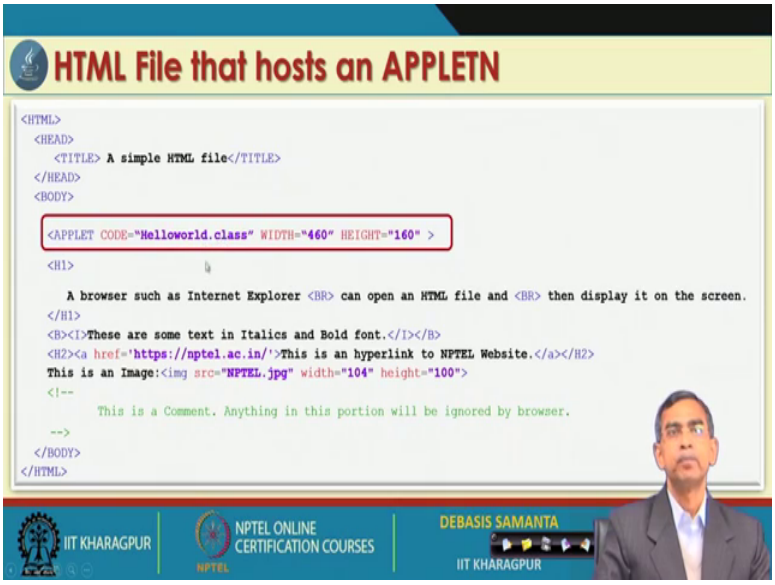
The next tag is H 2, H 2 is the text is basically small tag. And here not H 1 is the very bigger font, H 2 is a little bit smaller medium size font, so this will display here. And you see there is another a ref is basically link, so this is the another tag a, and then closing tag. So, every tag is here starting and then closing syntax. And then these basically see the link that means, from there this is the hyper link if we do it, so this and in this basically hyper link. Hyper link means if we click it, it will go to this link actually that means, it move to browser moves the focus from this link this page to some other distant page.

Then there are many other image also can be included in the text like. And in order to include an image, so image img is the tag for that. So, these basically display this kind of image on your screen. And the comment if you want to include in HTML, it basically comment will be included within this angular, then exclamation then dot, and then it is a closing is this one.

So, these are the few tags that I have shown here, so that how the HTML page can be designed using the tag only. And the browser it is basically run and HTML page. So, browser running an HTML page, this is basically in an interpretative manner. So, it will basically first tag, then second tag, next tag and so on. As it is interpretative not compilation nothing is there. So, if there is any mistake in these tags, say suppose H 1 is there, no closing this one, so this means that this will completely ignore. So, browser will not do anything ignore, it does not mean that the next portion will not be executed.

So, next portion will be interpreted and executed. So, it is an interpreter, and it will see every tag, and interpret every tag, and then display or paint on the screen. So, this basically the browser will paint on a window actually. So, this is basically the HTML page concept, and one more important thing that HTML tags whatever you can it is a case insensitive whether capital letter, small letter, it does not matter for the browser, the browser will do it correctly. So, is a case so is whether lower case on upper case that is not the issue there; so, this way the HTML page can be executed there.

(Refer Slide Time: 05:09)



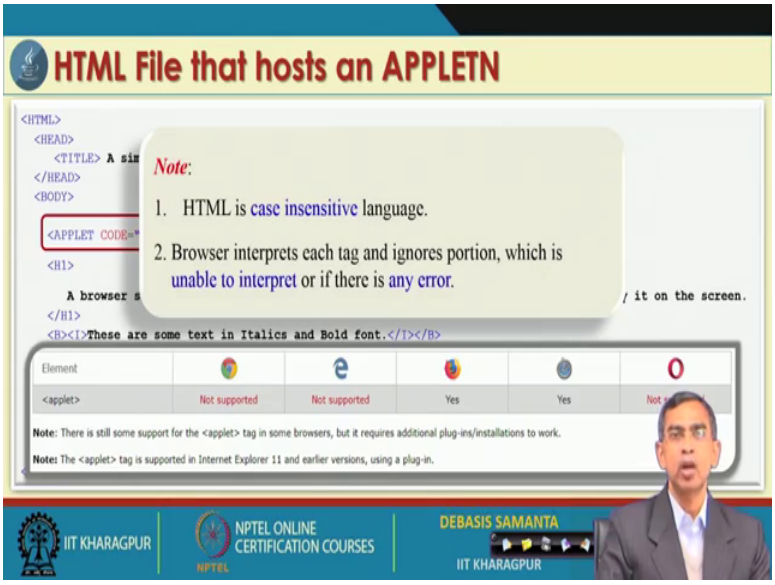
HTML File that hosts an APPLETN

```
<HTML>
  <HEAD>
     <TITLE> A simple HTML file</TITLE>
  </HEAD>
  <BODY>

    <APPLET CODE="Helloworld.class" WIDTH="460" HEIGHT="160" >

    <H1>

       A browser such as Internet Explorer <BR> can open an HTML file and <BR> then display it on the screen.
    </H1>
    <B><I>These are some text in Italics and Bold font.</I></B>
    <H2><a href='https://nptel.ac.in/'>This is an hyperlink to NPTEL Website.</a></H2>
    This is an Image:<img src="NPTEL.jpg" width="104" height="100">
    <!--
           This is a Comment. Anything in this portion will be ignored by browser.
     -->
  </BODY>
</HTML>
```

Now, we have discussed about the different tag. There is one special tag that is called the applet tag. So, this special tag is basically to include the code that you want to include in an HTML page. So, the tag look like applet, and this the code indicate that what is the class file in the Java class compile version of the Java applet program.

And then this is basically width and height that mean, it shows that what how much size of the applet would be in the windows that the browser will take to display your applet. So, these are the very nominal applet tag concept that is there. And then you can use it, and you can so if you can do it, your browser can run your applet from the machine current machine.

(Refer Slide Time: 05:55)



HTML File that hosts an APPLETN

```
<HTML>
  <HEAD>
    <TITLE> A sim
  </HEAD>
  <BODY>

   <APPLET CODE="

   <H1>

     A browser s                                                    / it on the screen.
   </H1>
   <B><I>These are some text in Italics and Bold font.</I></B>
```

Note:

1. HTML is case insensitive language.
2. Browser interprets each tag and ignores portion, which is unable to interpret or if there is any error.

Element		e			O
<applet>	Not supported	Not supported	Yes	Yes	Not s

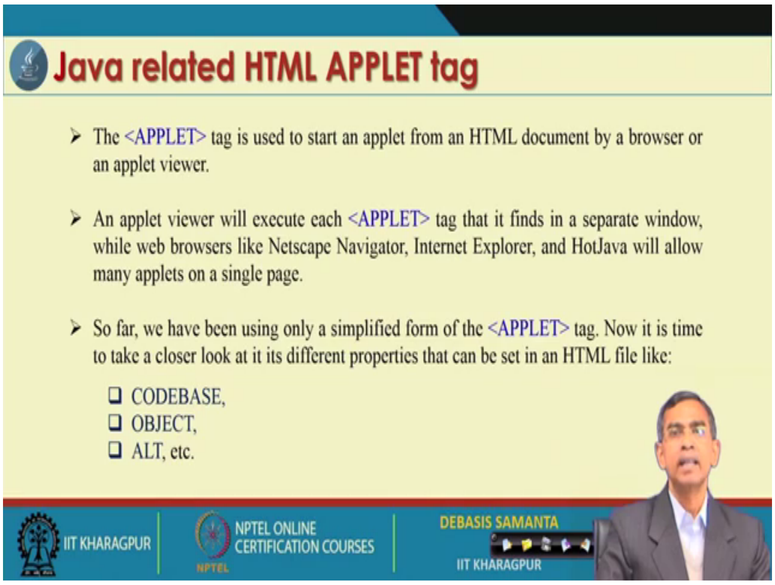
Note: There is still some support for the <applet> tag in some browsers, but it requires additional plug-ins/installations to work.

Note: The <applet> tag is supported in Internet Explorer 11 and earlier versions, using a plug-in.

Now, so this is the idea about, but there are few hinge is there is that now applet is little bit disregarded in the sense that many browser, they do not support until unless you uses special plugin for that. However, the chrome totally ignore any applet tag in the HTML page. Internet explorer till version 4.0, it was supporting and after that, it stop supporting.

Then Mozilla it can supports, if you use one special plugin for that. And this is the Netscape navigator, it always favor the applet code applet tag in an HTML page that means, Netscape navigator browse successfully. And the opera is another browser also which does not support it. So, many browser they do not support many browser support conditionally under the usage of plugin, whatever it is there. So, these are the basically restriction, and restriction there are certain reason for this kind of restriction, mainly it is from the security point of view. So, many browser they have stopped supporting applet code in an HTML.

(Refer Slide Time: 07:19)

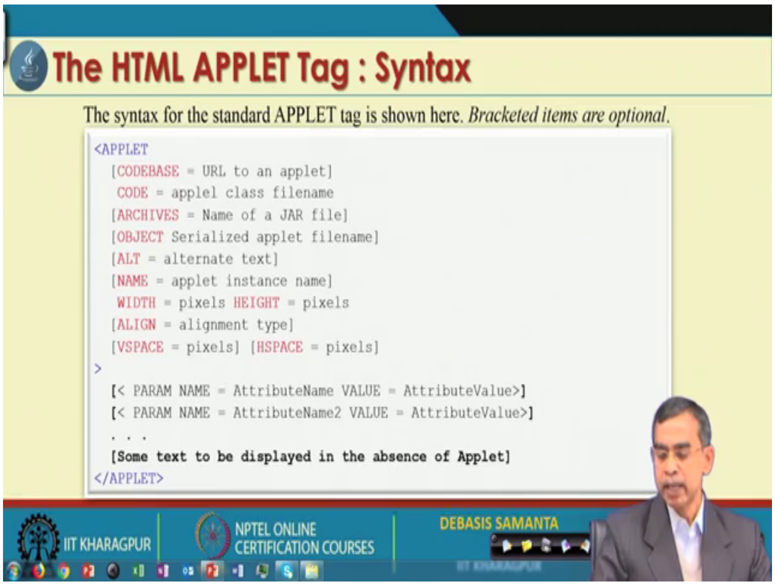


Java related HTML APPLET tag

- The <APPLET> tag is used to start an applet from an HTML document by a browser or an applet viewer.
- An applet viewer will execute each <APPLET> tag that it finds in a separate window, while web browsers like Netscape Navigator, Internet Explorer, and HotJava will allow many applets on a single page.
- So far, we have been using only a simplified form of the <APPLET> tag. Now it is time to take a closer look at it its different properties that can be set in an HTML file like:
  - ❑ CODEBASE,
  - ❑ OBJECT,
  - ❑ ALT, etc.

Now, see applet code; whenever it was initiated initially, applet code has many other parameters for the complete specification of applet.

(Refer Slide Time: 07:33)



The HTML APPLET Tag : Syntax

The syntax for the standard APPLET tag is shown here. *Bracketed items are optional.*

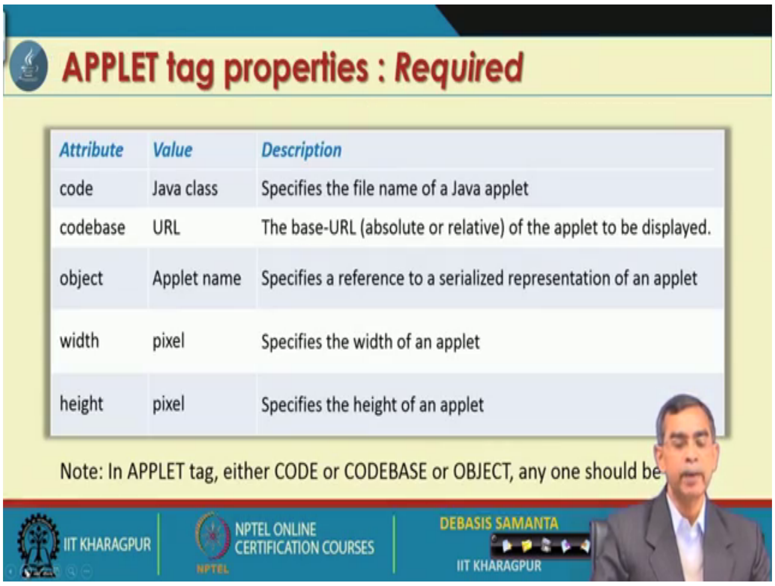
```
<APPLET
  [CODEBASE = URL to an applet]
   CODE = applel class filename
  [ARCHIVES = Name of a JAR file]
  [OBJECT Serialized applet filename]
  [ALT = alternate text]
  [NAME = applet instance name]
   WIDTH = pixels HEIGHT = pixels
  [ALIGN = alignment type]
  [VSPACE = pixels] [HSPACE = pixels]
>
  [< PARAM NAME = AttributeName VALUE = AttributeValue>]
  [< PARAM NAME = AttributeName2 VALUE = AttributeValue>]
  . . .
  [Some text to be displayed in the absence of Applet]
</APPLET>
```

The parameters those are there in a applet code are like which are some specification optional, and some are necessary. Those are the optional, which are written in the square bracket look like. So, for example, this is the applet tag initial, and this is the starting of the applet. And there if you see within this there is a few portion which you can include, and then this is in the applet portion; so, this is very important. So, this and then curly brackets is basically the different parameter that the applet tag will allow to specify for your special viewing of your applet. So, this is there, and what are the special thing that can be here you see.

And the codebase is the one parameter, archives is there code already we have discussed, then object, alt, name, width, height, align, vspace, and hspace are the few things that we have. And apart from this param name also somethings are there. If the param name is there, then separate angular brackets should be used for each parameter name. Param name probably for passing parameter or passing an input from HTML page to applet, we have discussed you can recall.

Now, here we can see out of this different specification that is possible for an applet, few things are mandatory. These are code, width, and heights, these are the mandatory that means, you have to give it. If you do not give it, your applet the tag will not be executed by a browser successfully. Whereas, the other the code based, archives, object, alt, name, and align, all these things are optional. And then param name also optional, you can have you also may not have it. So, these are the different tags specification; so, far the applet tag is concerned are there.

(Refer Slide Time: 09:37)



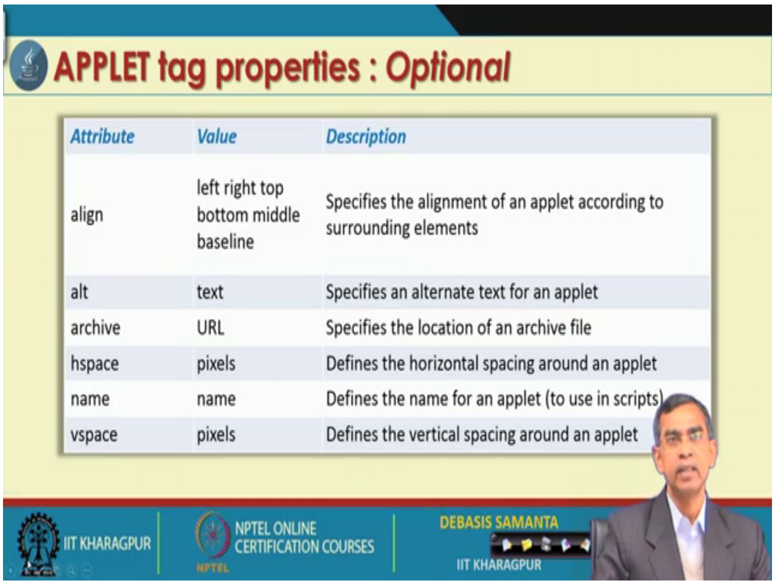
APPLET tag properties : Required

Attribute	Value	Description
code	Java class	Specifies the file name of a Java applet
codebase	URL	The base-URL (absolute or relative) of the applet to be displayed.
object	Applet name	Specifies a reference to a serialized representation of an applet
width	pixel	Specifies the width of an applet
height	pixel	Specifies the height of an applet

Note: In APPLET tag, either CODE or CODEBASE or OBJECT, any one should be

Now, in our next few slides, we will see exactly what is the meaning of each specification in the applet tag?

(Refer Slide Time: 09:39)

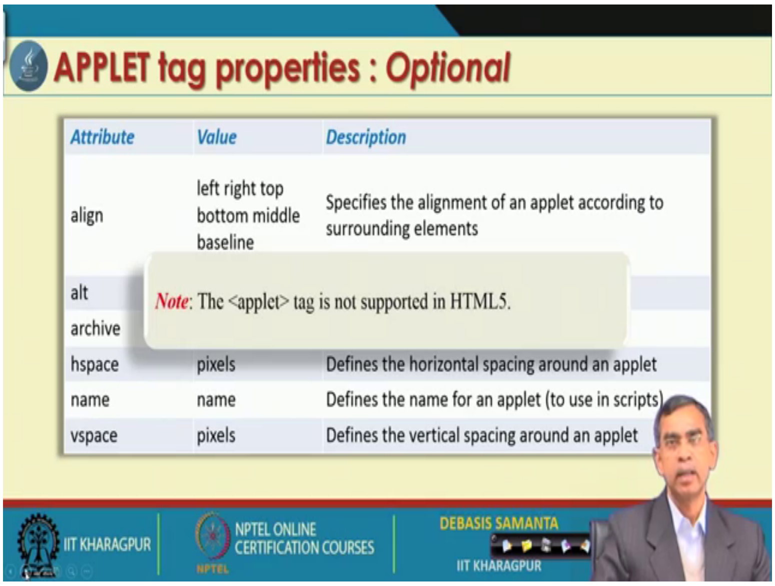


APPLET tag properties : Optional

Attribute	Value	Description
align	left right top bottom middle baseline	Specifies the alignment of an applet according to surrounding elements
alt	text	Specifies an alternate text for an applet
archive	URL	Specifies the location of an archive file
hspace	pixels	Defines the horizontal spacing around an applet
name	name	Defines the name for an applet (to use in scripts)
vspace	pixels	Defines the vertical spacing around an applet

Now, these are the ok, I have listed the few things that you can take your own time to go through, and then understand about it, but I will give the detail description in next slides.

(Refer Slide Time: 09:47)



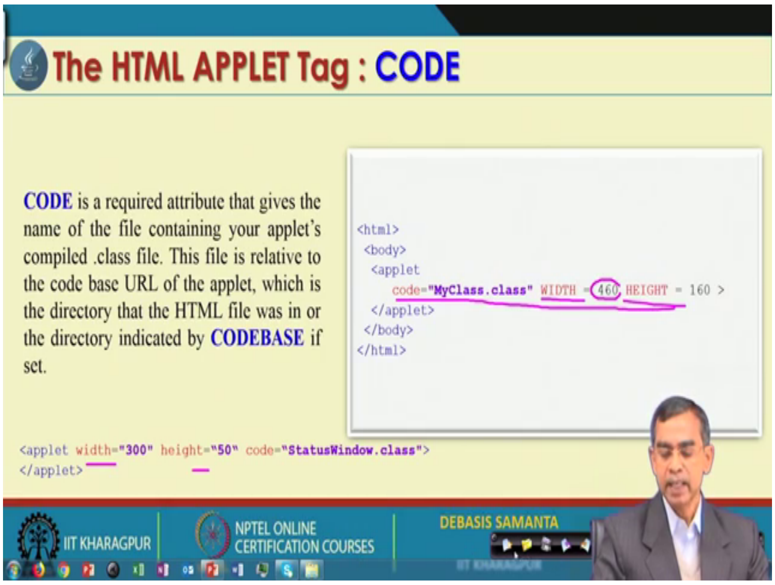
APPLET tag properties : Optional

Attribute	Value	Description
align	left right top bottom middle baseline	Specifies the alignment of an applet according to surrounding elements
alt		
archive		
hspace	pixels	Defines the horizontal spacing around an applet
name	name	Defines the name for an applet (to use in scripts)
vspace	pixels	Defines the vertical spacing around an applet

Note: The <applet> tag is not supported in HTML5.

And as I told you HTML also has the different versions. And HTML5 onwards, the applet tag is not supported. Rather instead of applet tag, they supported some other concept, it is called the object, I will discuss this things.

(Refer Slide Time: 10:07)



The HTML APPLET Tag : CODE

CODE is a required attribute that gives the name of the file containing your applet's compiled .class file. This file is relative to the code base URL of the applet, which is the directory that the HTML file was in or the directory indicated by **CODEBASE** if set.

```
<html>
 <body>
  <applet
     code="MyClass.class" WIDTH = 460 HEIGHT = 160 >
  </applet>
 </body>
</html>
```

```
<applet width="300" height="50" code="StatusWindow.class">
</applet>
```

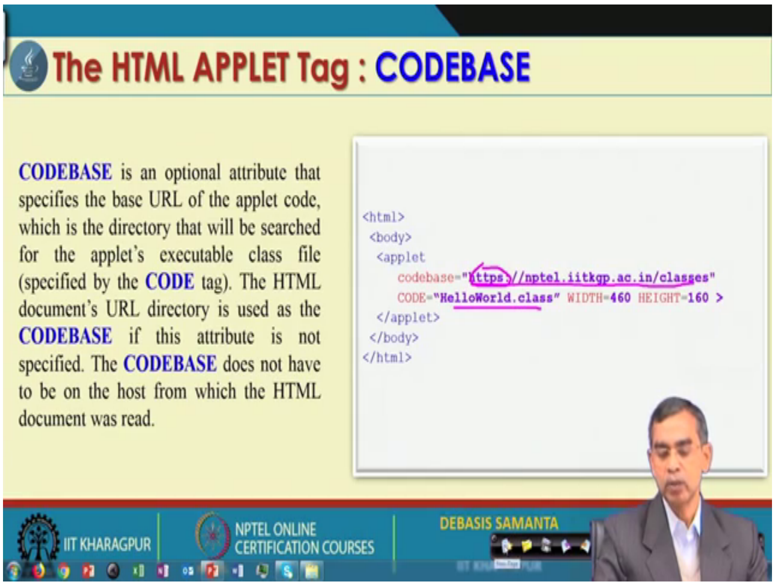
Now, first is basically the code. Code as it is the as you have learned it, within this applet. The code indicates that what is the class file that you want to include width and height specification. So, this is the minimum portion that it should have in your applet tag to run an applet.

So, this is basically code is there. And then code is basically here one important point that you should note. Whenever you run an HTML file hosting a class file, it basically require two things. The HTML file itself, it should be saved somewhere, and then the dot class file should be saved somewhere. So, if you do not specify anything explicitly, then your browser assume that the dot class file and HTML file are belongs to the same directory.

If you want to explicitly mention that it is not in the same directory, but somewhere else, then we should use the codebase. So, codebase usually accompanied with code, if you want to tell that the HTML file is here, and then code is somewhere else. Regarding codebase we will discuss details. ah

So, and another thing is that, so for the width and height whenever is the applet code, it is not necessary to maintain in this order. It also can be in any order like here width, height. And also another thing is that this is the width in pixel, those things also not necessary to put in a double quote, but is a customary to for the improving the viewer view (Refer Time: 11:51). So, we usually put in the double quote, but this needs to be put in the double quote always. So, here is an example like, so this is also correct, and this is also correct. So, these are the code and this code and with height and width are the mandatory things. Mandatory means, you have to use it without using you cannot do that.

(Refer Slide Time: 12:07)



The HTML APPLET Tag : CODEBASE

CODEBASE is an optional attribute that specifies the base URL of the applet code, which is the directory that will be searched for the applet's executable class file (specified by the CODE tag). The HTML document's URL directory is used as the CODEBASE if this attribute is not specified. The CODEBASE does not have to be on the host from which the HTML document was read.

```
<html>
 <body>
  <applet
     codebase="https://nptel.iitkgp.ac.in/classes"
     CODE="HelloWorld.class" WIDTH=460 HEIGHT=160 >
  </applet>
 </body>
</html>
```

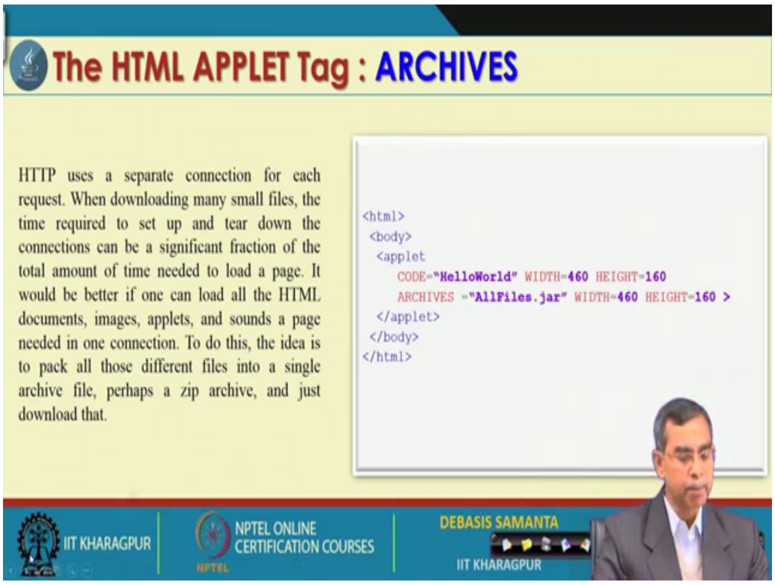
Now, let us come to the concept of code base. So, codebase is basically showing the relative address of your or base address of your applet. Now, what exactly the idea it is that if you want to I mean import your class file from a distance machine from your local other than your local. So, browser from which machine it is browsing, it is called the local to that browser. And if you want to have access from some distance machine remote machine, then the through browser that can be connected.

Here is an example say https indicates that it is the location of the remote repository, where this class file is there. So, this means that your browser will connect to this size, and then check this that class file import it into to your local machine, and then this applet code will be executed. So, applet code is not necessary to be executed from your own machine from the distance.

Usually, in case of internet programming, we are we have to do that. You can import your HTML page from the remote machine, and importing means all the class file applets, all the image, all the audio, video files, whatever it is required. Needs to be fetched into your browser machine, and then it will be executed there in your browser machine, so the codebase is for that purpose. We can explicitly mention that which is the location where the your class file is there.

Even if you use the same thing in your own local machine also, but it is that the HTML file is in belongs to one directory, whereas the class file belong to the separate directory. So, the base is basically here, which is the base directory and with respect to this base, which is the class file that you want to use. So, in the codebase also that we can use it, but in that case we have to use the file command here. So, file and then the location of the directory, and then the name of the class file that you want to use it. So, this is the use of the tag the codebase.

(Refer Slide Time: 14:29)



The HTML APPLET Tag : ARCHIVES

HTTP uses a separate connection for each request. When downloading many small files, the time required to set up and tear down the connections can be a significant fraction of the total amount of time needed to load a page. It would be better if one can load all the HTML documents, images, applets, and sounds a page needed in one connection. To do this, the idea is to pack all those different files into a single archive file, perhaps a zip archive, and just download that.

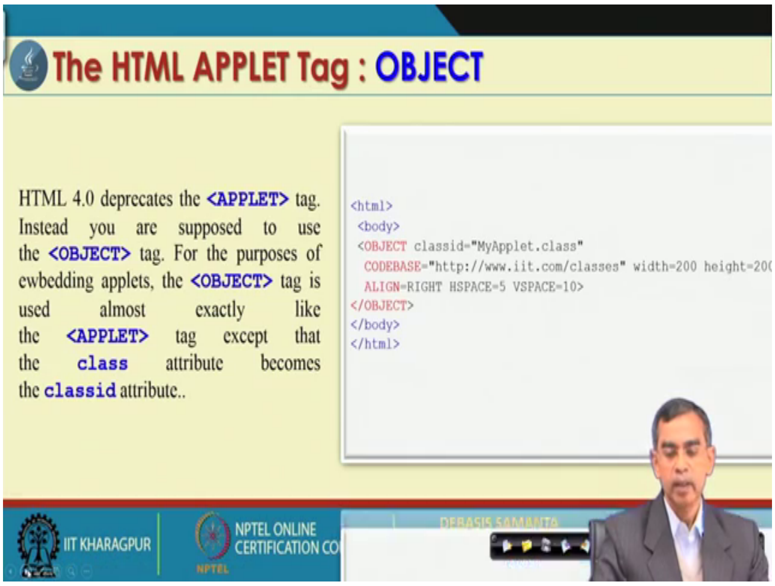
```
<html>
 <body>
  <applet
     CODE="HelloWorld" WIDTH=460 HEIGHT=160
     ARCHIVES ="AllFiles.jar" WIDTH=460 HEIGHT=160 >
  </applet>
 </body>
</html>
```

And archive is there for this purpose is that say suppose, as I told you applet can be viewed from your local machine containing its elements from the distance machine. If you want to do that, then every times you want some things to be included in your applet, it needs to be connected to your remote machine and then import it, and then store it, and then display.

So, instead of whatever the files that is required for that HTML page, you can put into a archive in a jar file it is called here for example all files those are required for an HTML page can be put into one jar file. Then browser what it will do is that browser will import only once with only single connection to the remote server, and then fetch it into a local machine.

And then from this jar file whatever the file that is required image or video, audio or any other class file will be extracted from there and will be referred to this one. So, this archive commands will basically gives an idea about that which is the jar file that needs to be fetched only once in a lifetime of the viewing. So, this is idea about the archive tag specification in the applet code.

(Refer Slide Time: 15:54)



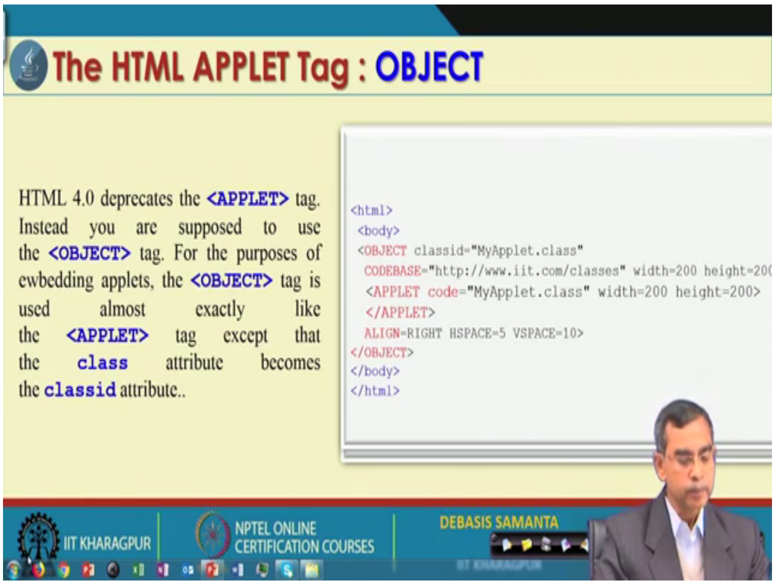
The HTML APPLET Tag : OBJECT

HTML 4.0 deprecates the **<APPLET>** tag. Instead you are supposed to use the **<OBJECT>** tag. For the purposes of ewbedding applets, the **<OBJECT>** tag is used almost exactly like the **<APPLET>** tag except that the **class** attribute becomes the **classid** attribute..

```
<html>
 <body>
 <OBJECT classid="MyApplet.class"
  CODEBASE="http://www.iit.com/classes" width=200 height=20(
  ALIGN=RIGHT HSPACE=5 VSPACE=10>
</OBJECT>
</body>
</html>
```

And then object. Object as I told you HTML5, and upwards version of the HTML, they have stopped supporting applet code. So, in that case is basically allow to use the object code. So, object code is same as the applet tag actually; so applet is not there. So, here basically the same instead of applet, you use the object that is all; so, this basically the idea about. And other code based, then width, height everything is applicable to that only. So, object is an alternative to applet tag that is the newer version in the HTML page.

(Refer Slide Time: 16:32)



The HTML APPLET Tag : OBJECT

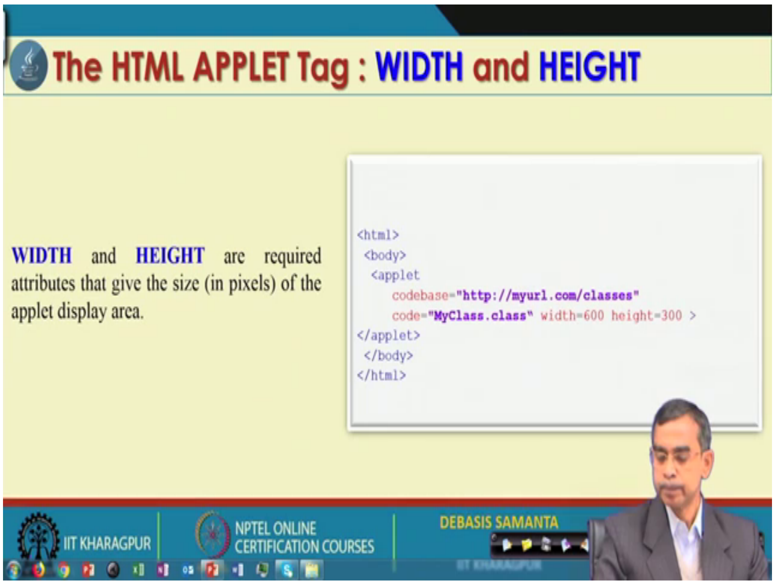
HTML 4.0 deprecates the **<APPLET>** tag. Instead you are supposed to use the **<OBJECT>** tag. For the purposes of ewbedding applets, the **<OBJECT>** tag is used almost exactly like the **<APPLET>** tag except that the **class** attribute becomes the **classid** attribute..

```
<html>
 <body>
 <OBJECT classid="MyApplet.class"
  CODEBASE="http://www.iit.com/classes" width=200 height=20(
   <APPLET code="MyApplet.class" width=200 height=200>
   </APPLET>
  ALIGN=RIGHT HSPACE=5 VSPACE=10>
</OBJECT>
</body>
</html>
```

And here again if you want to suppose, you do not know whether your browser will support, whether HTML4 or HTML2 or HTML5 or upwards so, sometimes to make it more robust for version independent or compatible to any versions; so some HTML designer used both object as applet code here. For example, object is there here, and then applet code is also there. So, if it is fetches for the for example HTML4, you want to your browser supporting HTML4, then object will not be applicable to that HTML browser. So, in that case, you can use the this one; so all these will be ignored, but applet will be considered in that case.

On the other hand, another version which support HTML5, then the only object; so, this will be ignored by that browser, only this portion will be considered, and the applet that file will be successfully browsed. So, this is the idea about that both object and applet can be merged together, to make it more compatible to your browser system.

(Refer Slide Time: 17:44)



The HTML APPLET Tag : WIDTH and HEIGHT

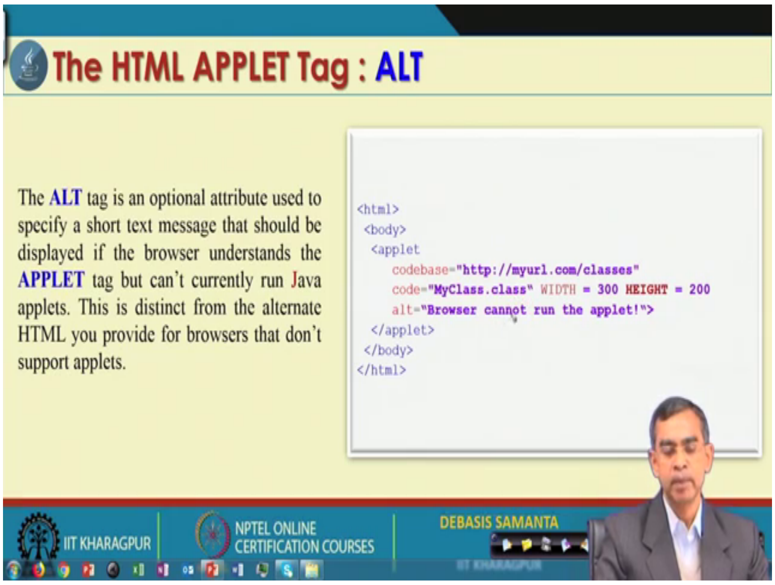
WIDTH and HEIGHT are required attributes that give the size (in pixels) of the applet display area.

```
<html>
 <body>
  <applet
     codebase="http://myurl.com/classes"
     code="MyClass.class" width=600 height=300 >
</applet>
 </body>
</html>
```

And width and height as I told you specification of the height and width by changing this parameters, we can resized your applet view, and that can be done from your applet from the HTML page itself, so that is why the ok. Before, viewing the HTML, so HTML will be footed in front of within set of text in a multimedia document like, it can include audio, image, and then video also, in addition to the hyper link text, hypertext we can say.

And then the applet is basically an windows within this text also, it appears as a living on things for the browser. So, in that case we can specify the width and height, how much space of your total view area of your display that it will take to display your applet. So, this is the width and height is an essential one parameters ok.

(Refer Slide Time: 18:49)



The HTML APPLET Tag : ALT

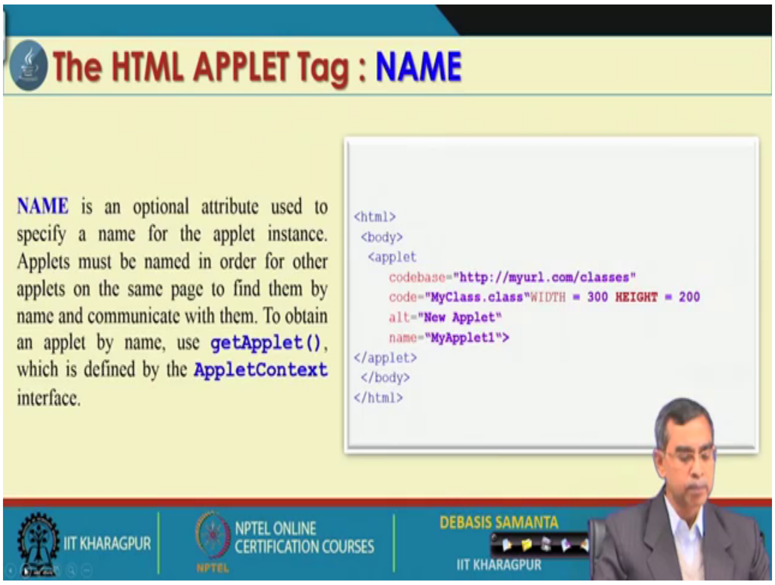
The ALT tag is an optional attribute used to specify a short text message that should be displayed if the browser understands the APPLET tag but can't currently run Java applets. This is distinct from the alternate HTML you provide for browsers that don't support applets.

```
<html>
 <body>
  <applet
     codebase="http://myurl.com/classes"
     code="MyClass.class" WIDTH = 300 HEIGHT = 200
     alt="Browser cannot run the applet!">
  </applet>
 </body>
</html>
```

And then the alt. Say suppose your applet, there are some mistakes in your class file or there is a problem that it is not able to load it successfully, then browser can give some message to the user. So, here the message that can be given by means of all statement. In case, suppose this is this is a problem right, your browser face the problem of fetching this class or executing this class file, then it will be there; otherwise, this will not be executed.

So, this is basically for giving user good interaction or impression about viewing a page. So, alt statement is used there. So, alt is basically an alternate text within this double quote can be used. And you see one thing here all these things will be within this applet tag, whatever this codebase, code, alt, width, height, name whatever it is there. Those are the within this applet state and finally, this is called the final applet tag.

(Refer Slide Time: 19:55)



The HTML APPLET Tag : NAME

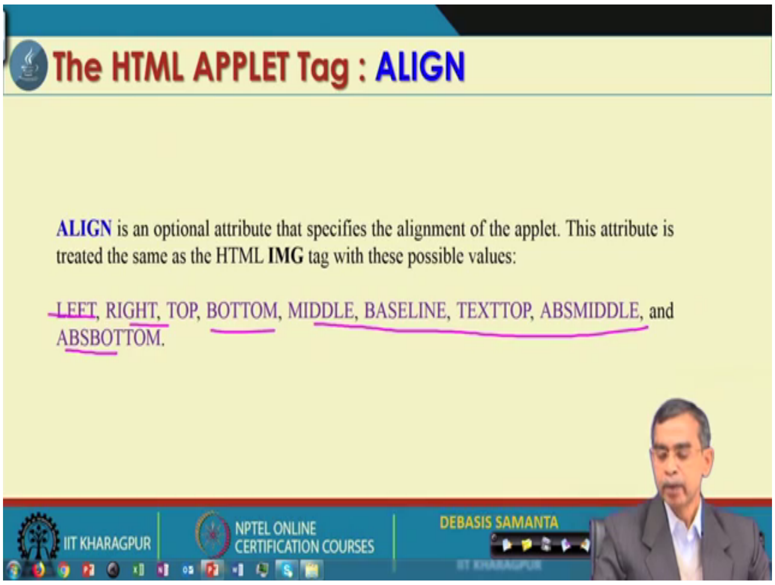
NAME is an optional attribute used to specify a name for the applet instance. Applets must be named in order for other applets on the same page to find them by name and communicate with them. To obtain an applet by name, use **`getApplet()`**, which is defined by the **`AppletContext`** interface.

```
<html>
 <body>
  <applet
     codebase="http://myurl.com/classes"
     code="MyClass.class"WIDTH = 300 HEIGHT = 200
     alt="New Applet"
     name="MyApplet1">
</applet>
 </body>
</html>
```

Next is the name. And in your HTML page, you can use many applets into the same HTML page or sometimes applet can be used in a distributed programming system also. From your machine, you are running an applet. From another machine, another user running, another applet and both the applets needs to be communicated to each other. And for this communication, so that actually one specification that is required that which applets communicate to which one that mean, each applets should have their own identity; this identity can be mentioned by means of name.

So, name is the one specification that is there by which we can specify that if each applet has its own unique name. And then by this name the applet in your applet programming using the method get name the name of the applet can be understood, and using this under say some communication can be established or some communication protocol can be implemented, so that is why the name that specification has been there in the applet. And then so name can be obtained by the getApplet method not get name ok.

(Refer Slide Time: 21:17)



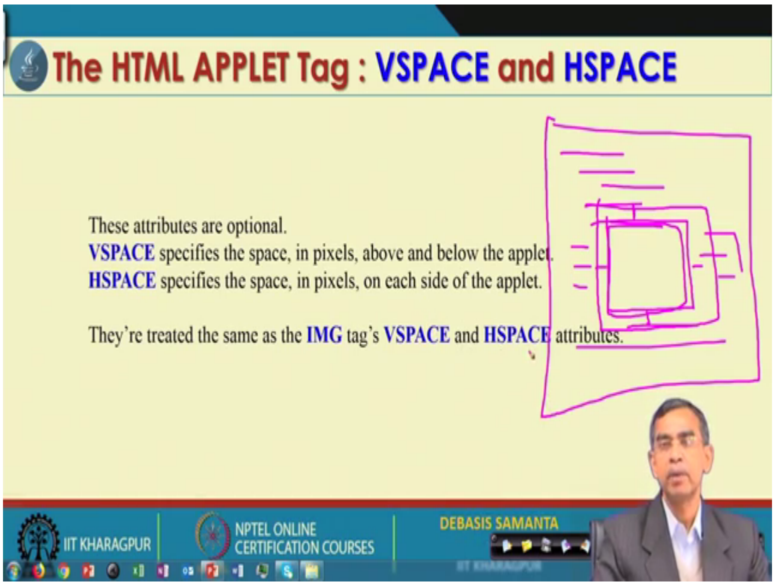
The HTML APPLET Tag : ALIGN

ALIGN is an optional attribute that specifies the alignment of the applet. This attribute is treated the same as the HTML **IMG** tag with these possible values:

LEFT, RIGHT, TOP, BOTTOM, MIDDLE, BASELINE, TEXTTOP, ABSMIDDLE, and ABSBOTTOM.

So, these are the there are few more method tag align. Align means you can view your applet in a particular position. For example, these are so many position left, right, top, bottom, and absolute bottom, whatever. There are many other specification that you can find you can mention, so that applet will be placed accordingly whether left side, right side, top, bottom absolute top, top of some text like this one. So, these are the basically placing one applet in an HTML page HTML area view windows that is browsed by the applets.

(Refer Slide Time: 21:51)



The HTML APPLET Tag : VSPACE and HSPACE

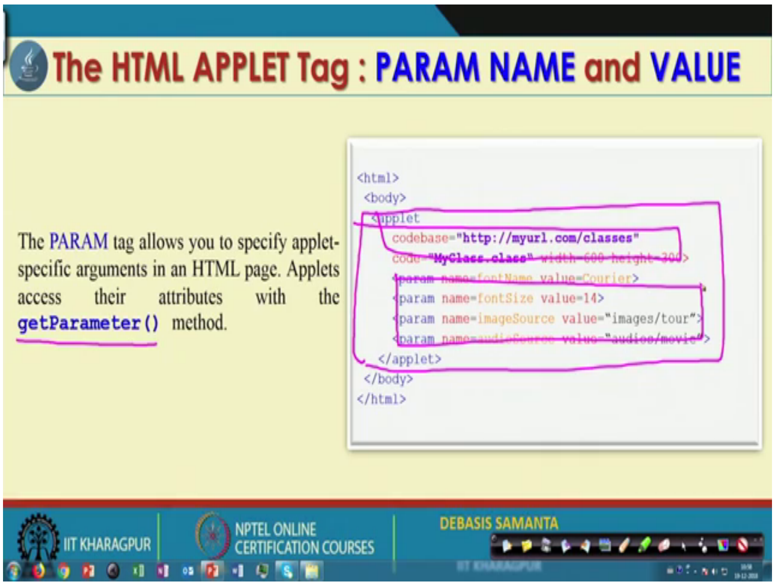
These attributes are optional.
VSPACE specifies the space, in pixels, above and below the applet.
HSPACE specifies the space, in pixels, on each side of the applet.

They're treated the same as the **IMG** tag's **VSPACE** and **HSPACE** attributes.

And VSPACE and HSPACE is basically the use of this say suppose, this is the total HTML view of your total HTML page. And there some text like, and here also some text like and here you want to display your applet like suppose. Now, so VSPACE, HSPACE is basically how much gap that you want to give between the text and this applet. And so it is basically VSPACE.

Similarly, if there are some other text, it is there or some other applet is there you want to place it, then this basically HSPACE means what is the gap that you want to allow between the text or this one. So, is basically VSPACE, HSPACE totally is say that other than this is the applet this is the applet view, this is the width and height is basically how much space that you want to do it. Sometimes it is basically used for good impression of the applets, and then artistical presentation of this applets that in there is HTML that can be controlled using this VSPACE, HSPACE commands.

(Refer Slide Time: 22:59)



The HTML APPLET Tag : PARAM NAME and VALUE

The PARAM tag allows you to specify applet-specific arguments in an HTML page. Applets access their attributes with the getParameter() method.

```
<html>
 <body>
  <applet
     codebase="http://myurl.com/classes"
     code="MyClass.class" width=600 height=300>
     <param name=fontName value=Courier>
     <param name=fontSize value=14>
     <param name=imageSource value="images/tour">
     <param name=audioSource value="audios/movie">
   </applet>
 </body>
</html>
```

And parameter name param is the another important parameter, extensively used one parameter that is required to specify the input to an applet as you know applet cannot take the direct input from the browser machine or cannot take direct input from the distance machine also. So, if you want to pass any input to your applet, then in the HTML page the parameter names to be mentioned there.

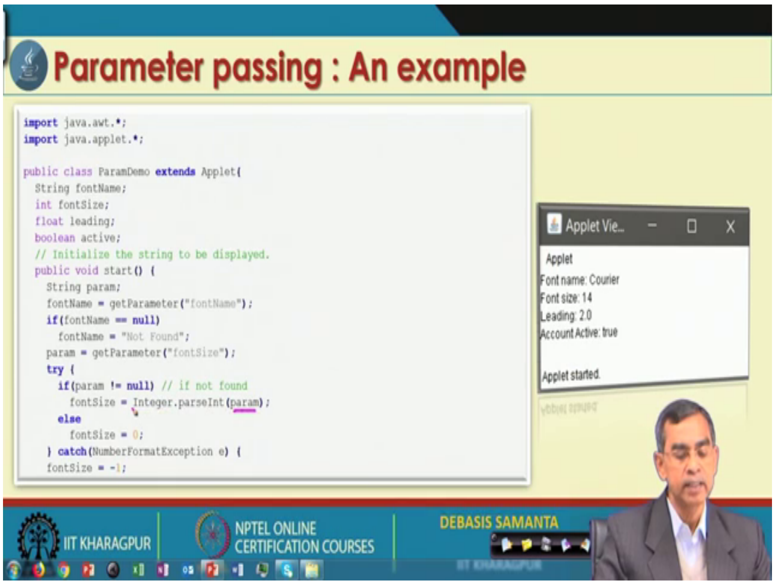
Now, the syntax it is like this. The tag specification is called the param, the name of the parameter. For example, in this example as we see font name is the name of a parameter, and value is basically the courier, so that means, this is the parameter the font name the value is courier. Then font size is another parameter, whose value is 14 at present. The name is image source, the value is basically the file location.

And then name is another is that audio source, the value is this one. So, here we can see these are the different parameters, whose values are like this which we want to pass from this HTML page to this applet, when it is in execution. So, whenever the applets starts its execution, it will refer whatever the parameter that is mentioned there in this HTML applet code will obtain the value from here, and it will use it.

So, for this purpose the parameters concept param tag is there defined there. And this tag you see this is the applet, applet is applet is there, this is the applet part. Yeah this is basically the applet tag, but within this applet tag a separate parameter tags are to be mentioned there. So, and then whole the things will be basically the applet tag for the HTML page.

Now, so param name, so how an applet can get this values. So, for this purpose, the get parameter methods are there defined in applet class, and which can be called for this. And then can be can get this values from the HTML itself. So, if you change this values from your machine, and then this will display with the different value to your applets. Now, let us see one example to illustrate the use of this one. And this is suppose the HTML page that you have defined, and the my class dot class is our applet, which basically want to take input from this value. And then how the applet code will be there.

(Refer Slide Time: 25:40)



Parameter passing : An example

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

public class ParamDemo extends Applet{
  String fontName;
  int fontSize;
  float leading;
  boolean active;
  // Initialize the string to be displayed.
  public void start() {
    String param;
    fontName = getParameter("fontName");
    if(fontName == null)
      fontName = "Not Found";
    param = getParameter("fontSize");
    try {
      if(param != null) // if not found
        fontSize = Integer.parseInt(param);
      else
        fontSize = 0;
    } catch(NumberFormatException e) {
    fontSize = -1;
```

```
Applet
Font name: Courier
Font size: 14
Leading: 2.0
Account Active: true

Applet started.
```

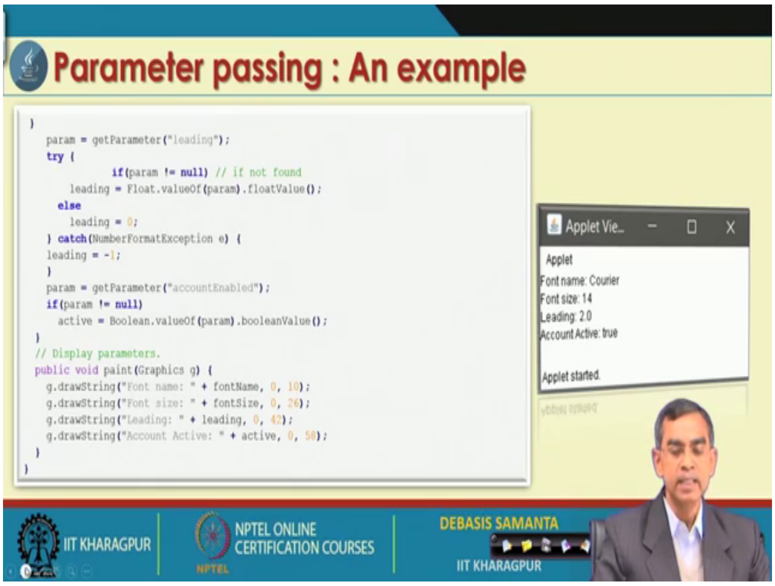
Now, here is the applet code which look like this here if you see the paramdemo is the name of the applet, which you are going to demo it, and these are the different parameters called the field of this class like fontname, fontsize, leading, active. All these are the variable as we have declared in a class; so this is the class members.

And here you see the methods start is basically to be either it is in the start method or initialize method, you can read the value. So, here in this case we can read the value from the start method or like this one. Now, here as you see all the parameter value that will be read from the HTML page, they will read from a string. So, let the name of the string be param and read. And then from this parameter string, we can convert into the different form.

So, is a get parameter font name, which is there in a HTML page, you will get this and then font name will be there. So, this is the basically parameter that is there font name, font name from a courier. So, the value will be there, so font name is currently the courier. And it is basically the try-catch is mentioned, if in case suppose, this value is not present in the HTML page like this one.

And then so there is another param is if you want to convert in integer also, because it is read as a string, so we can change into integer. So, integer class, then parseint, it will convert the string to integer, so font size, and then this one ok. Now, so this way as the different parameters, there are few more parts of the text is there, code is there. The different parameters can be read using this get parameter method is there.

(Refer Slide Time: 27:31)



Parameter passing : An example

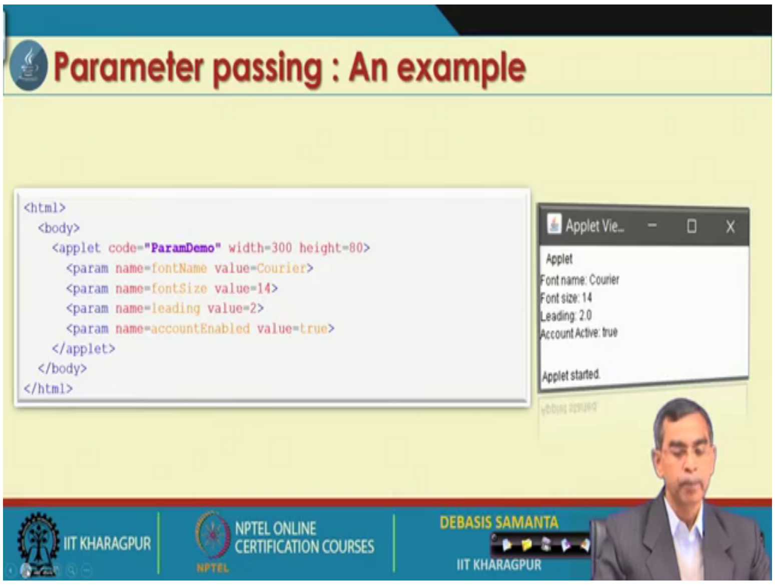
```
}
  param = getParameter("leading");
  try {
              if(param != null) // if not found
      leading = Float.valueOf(param).floatValue();
    else
      leading = 0;
  } catch(NumberFormatException e) {
  leading = -1;
  }
  param = getParameter("accountEnabled");
  if(param != null)
    active = Boolean.valueOf(param).booleanValue();
 }
 // Display parameters.
 public void paint(Graphics g) {
  g.drawString("Font name: " + fontName, 0, 10);
  g.drawString("Font size: " + fontSize, 0, 26);
  g.drawString("Leading: " + leading, 0, 42);
  g.drawString("Account Active: " + active, 0, 58);
 }
}
```

```
Applet
Font name: Courier
Font size: 14
Leading: 2.0
Account Active: true

Applet started.
```

Now, other parameter also get parameter leading, and then get parameter account enable. All these value will be read from the HTML page. And then the once the value is loaded into your applet, and then we can use them in your paint method; so, here is the paint method. And you see whatever the value that we could read, it from the HTML page like font name, font size, reading, and active.

And then they can be printed on the screen like in this case, and then if you run this applet using the HTML file that we have previously shown, it will basically like this one. So, what we want to say here is that whatever the parameters, we want to pass from an HTML page to an applet, they can be embedded into the HTML page itself in within the applet tag by the special tag specification called the param. So, this way we can pass the parameter and we can run this like this. So, usually the HTML page designer have this provision to pass the different values or there is a many other way also, it can be there.

(Refer Slide Time: 28:46)



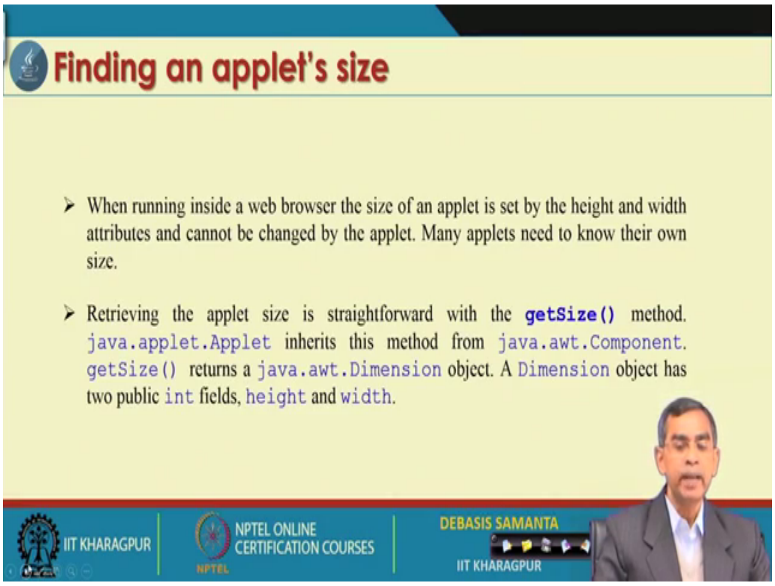
Parameter passing : An example

```
<html>
  <body>
    <applet code="ParamDemo" width=300 height=80>
      <param name=fontName value=Courier>
      <param name=fontSize value=14>
      <param name=leading value=2>
      <param name=accountEnabled value=true>
    </applet>
  </body>
</html>
```

```
Applet
Font name: Courier
Font size: 14
Leading: 2.0
Account Active: true

Applet started.
```

One good example that we can have it, then we can see the applet is ok, it is just a windows program. And this is the HTML page for this applet, it will look like.

(Refer Slide Time: 28:53)

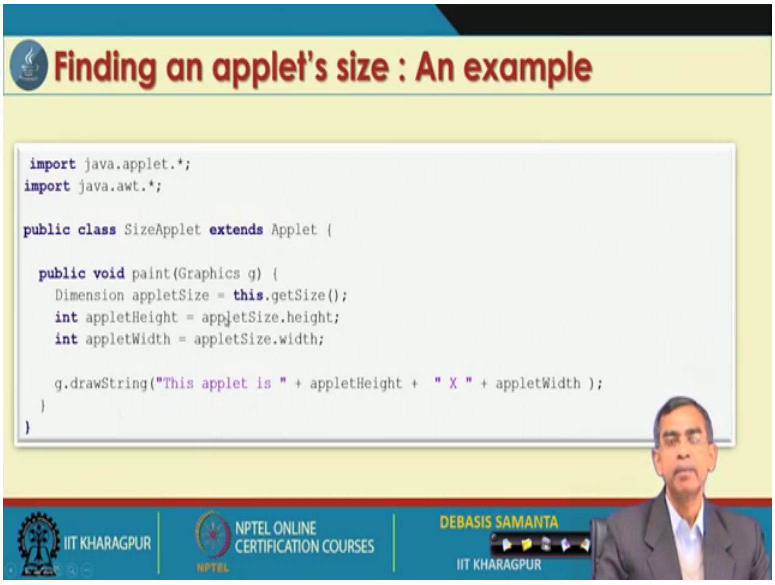


Finding an applet's size

- When running inside a web browser the size of an applet is set by the height and width attributes and cannot be changed by the applet. Many applets need to know their own size.
- Retrieving the applet size is straightforward with the `getSize()` method. `java.applet.Applet` inherits this method from `java.awt.Component`. `getSize()` returns a `java.awt.Dimension` object. A `Dimension` object has two public `int` fields, `height` and `width`.

The applets finding an applet size; now sometimes it is required to interact with the applet size which some situation, then we need to know what is the current size of the applet. So, the current size of the applet can be understood using another method called the get size, which is defined in applet methods are there. So, this basically get the applet size like.

(Refer Slide Time: 29:18)

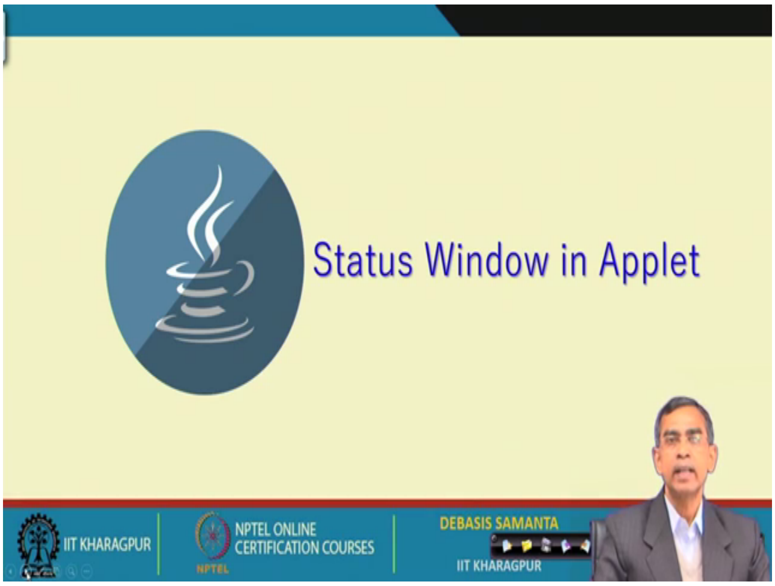


Finding an applet's size : An example

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

public class SizeApplet extends Applet {

  public void paint(Graphics g) {
    Dimension appletSize = this.getSize();
    int appletHeight = appletSize.height;
    int appletWidth = appletSize.width;

    g.drawString("This applet is " + appletHeight +  " X " + appletWidth );
  }
}
```

And this example says for a current applet, what is this width and height parameter that can be read here.

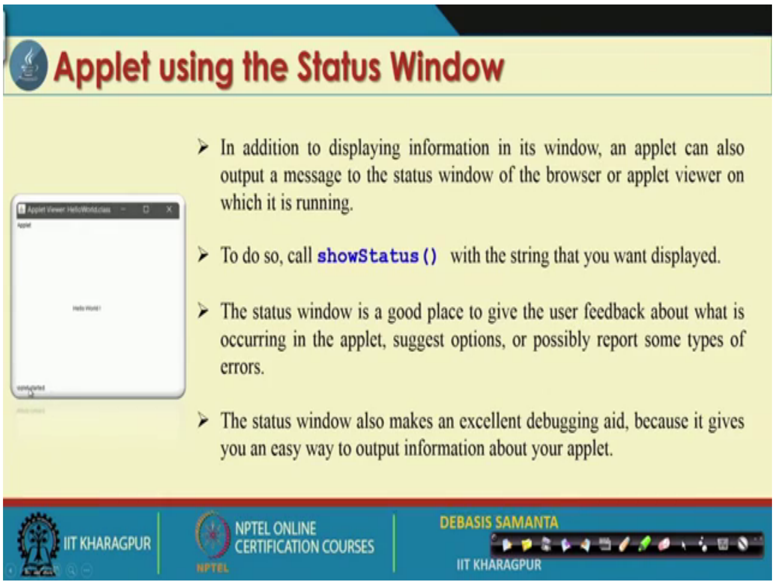
(Refer Slide Time: 29:26)



Status Window in Applet

Now, there is also another method is also in an applet is basically called the show status method.

(Refer Slide Time: 29:34)

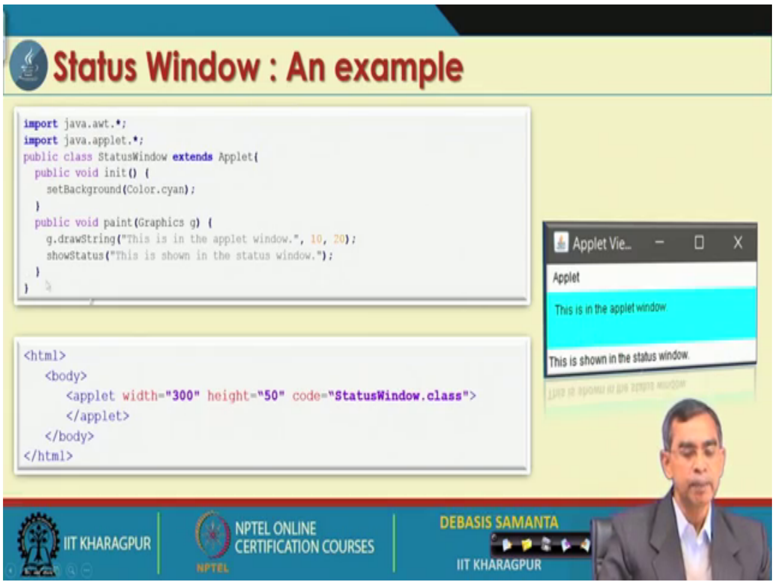


Applet using the Status Window

- In addition to displaying information in its window, an applet can also output a message to the status window of the browser or applet viewer on which it is running.
- To do so, call **showStatus()** with the string that you want displayed.
- The status window is a good place to give the user feedback about what is occurring in the applet, suggest options, or possibly report some types of errors.
- The status window also makes an excellent debugging aid, because it gives you an easy way to output information about your applet.

So, here basically idea is that this basically showing the status of the applet. If the applets started its execution, it is called the applet started. If it is destroyed, it is applet is stopped or killed or if it is applet is not yet started like not started this kind of. So, this is the called the status bar of the applet, and this is the title of the applet. Now, sometimes the showstatus, we can for some reason that we want to display our own, what is called the message on to this one. So, this can be done using the showstaus method.

(Refer Slide Time: 30:19)



Status Window : An example

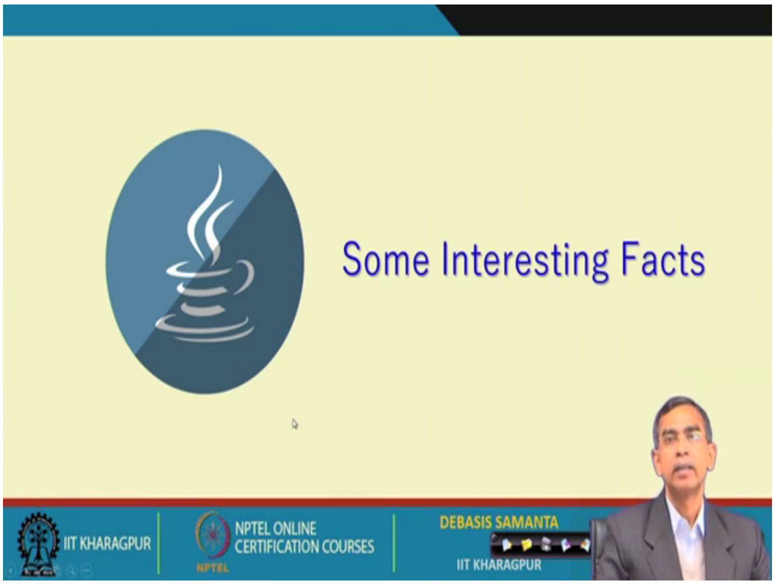
```
import java.awt.*;
import java.applet.*;
public class StatusWindow extends Applet{
  public void init() {
    setBackground(Color.cyan);
  }
  public void paint(Graphics g) {
    g.drawString("This is in the applet window.", 10, 20);
    showStatus("This is shown in the status window.");
  }
}
```

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

Applet
This is in the applet window.
This is shown in the status window.

And here is the one simple example, how this showstatus method can be utilized. Here you see this method can be used here showstatus, and this string. And if you run this applet, then it basically show in this status bar there that this is shown in the status window like.

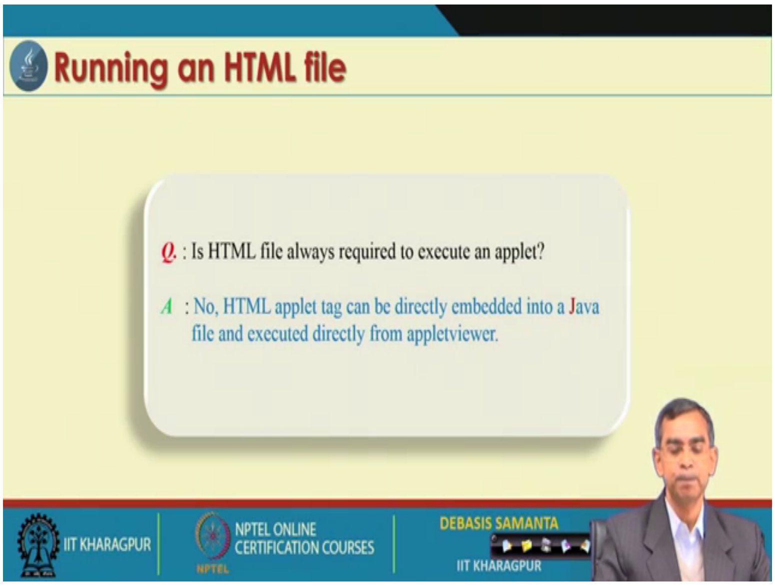
(Refer Slide Time: 30:40)



Some Interesting Facts

So, this is this is the different methods are there that can be used to display the status concept here. And there are few more interesting facts that I want to give, before concluding this module.

(Refer Slide Time: 30:53)



Running an HTML file

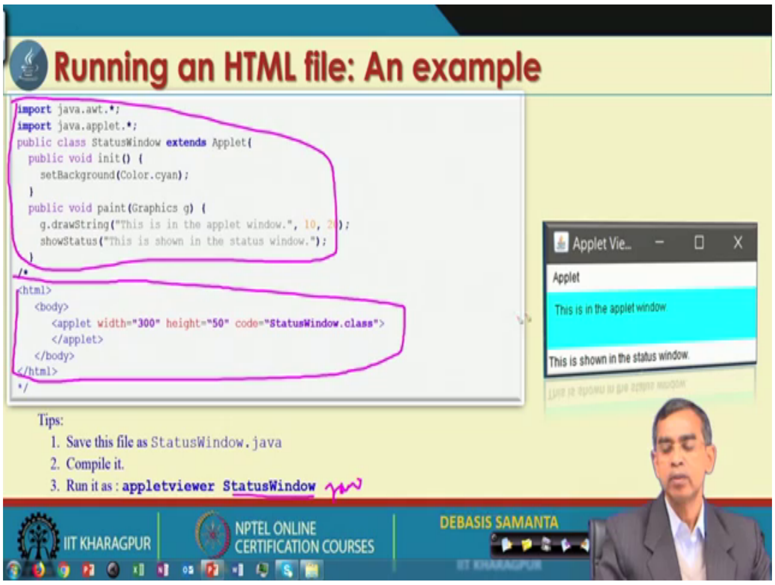
Q. : Is HTML file always required to execute an applet?

A : No, HTML applet tag can be directly embedded into a Java file and executed directly from appletviewer.

And let us see the first is HTML file is always necessary to execute an applet that means, as I have mentioned that HTML code should be there, HTML file is there, but there is a tricks that the tricks is basically HTML applet tag can be directly embedded into the Java program itself. And then that can be used by the applet viewer actually.

Now, what is the policy it is here that means, whatever the applet tag, it is there. You just write after the HTML file, either after or before whatever wherever it is there as shown no problem. Then same dot Java file can be used, so no need to maintain two files. One is the dot Java file or dot class, and then dot HTML, but only one dot java, and then corresponding class file should be there complied version.

(Refer Slide Time: 31:43)



Running an HTML file: An example

```
import java.awt.*;
import java.applet.*;
public class StatusWindow extends Applet{
  public void init() {
    setBackground(Color.cyan);
  }
  public void paint(Graphics g) {
    g.drawString("This is in the applet window.", 10, 20);
    showStatus("This is shown in the status window.");
  }
/*
<html>
   <body>
      <applet width="300" height="50" code="StatusWindow.class">
      </applet>
   </body>
</html>
*/
```

Tips:

1. Save this file as `StatusWindow.java`
2. Compile it.
3. Run it as : **`appletviewer StatusWindow`**

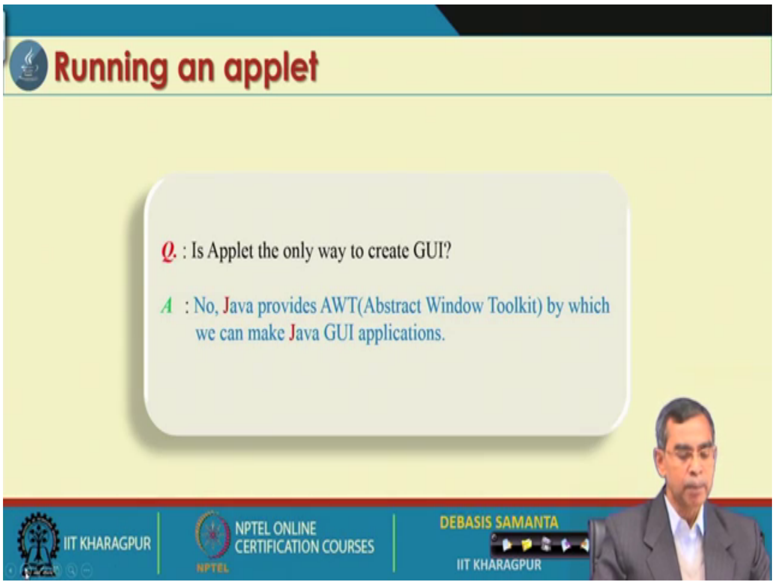
Applet Vie...
Applet
This is in the applet window
This is shown in the status window.
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
DEBASIS SAMANTA

Now, here is an example as you can see here, this is the applet program that we have already know; so, this is the dot Java file. And here is the HTML code, which basically to host it. Now, I can maintain only one file, and then let the name of the file is dot java. So, this is the dot Java may be status window dot Java is the name of the applet file compile it. Now, whenever you go for compilation, those are the things including within this comment that means, your compiler the Java compiler will ignore this one.

On the other hand, whenever applet viewer will come into their, applet viewer will ignore, all this paths because applet viewer cannot understand about this tag and simply ignore it is just interpreted. But, whatever the portions it is there, applet viewer can understand it, and then can interpreted.

So, if we run this using applet viewer for example, here run it applet viewer statuswindow dot java. So, here you have to write dot java, this is an original file right. So, then as the applet viewer will browse it, and then displayed. So, only one HTML file that you can plan only one dot file only dot Java file can be maintained, and then no need to maintain explicitly dot HTML file. So, this is the idea about and many programmer prefer to do that, what are the code HTML tag it is required, they usually use in the dot Java file within the comment lines only; so this is the one.

(Refer Slide Time: 33:25)



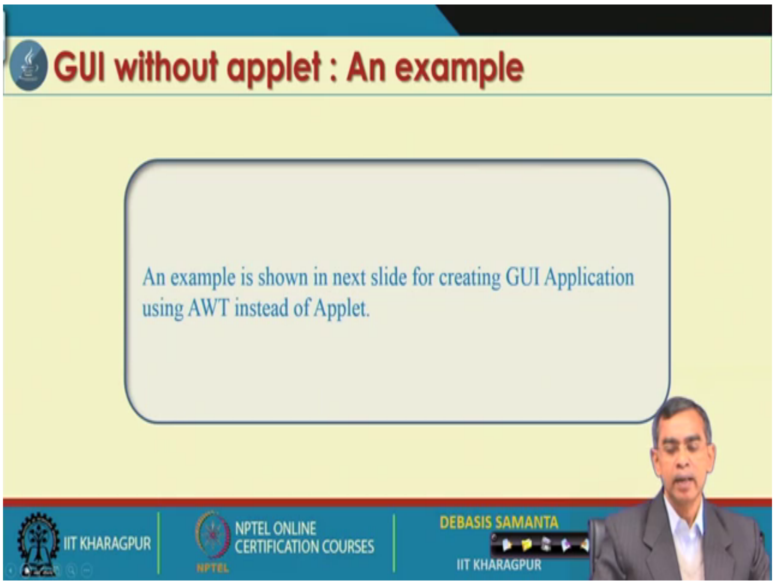
Running an applet

Q. : Is Applet the only way to create GUI?

A : No, Java provides AWT(Abstract Window Toolkit) by which we can make Java GUI applications.

And another is that is applet the only way to create graphical user interface or there is many other ways are there. In our next discussion, we will discuss about the AWT the Abstract Windowing Tool kit. The abstract windowing tool kit allows us to include our code into here. So, using this one also you can do it, but regarding this abstract window tool kit, we will discuss all this things in details there.

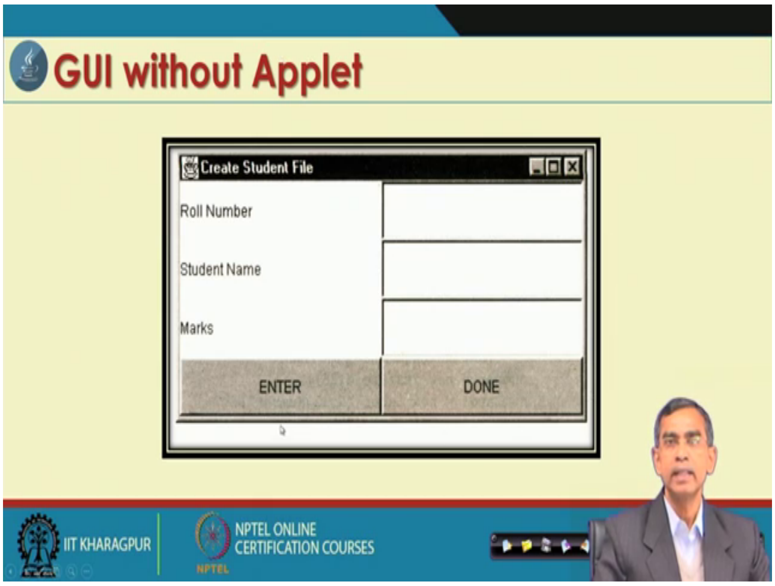
(Refer Slide Time: 33:52)



GUI without applet : An example

An example is shown in next slide for creating GUI Application using AWT instead of Applet.

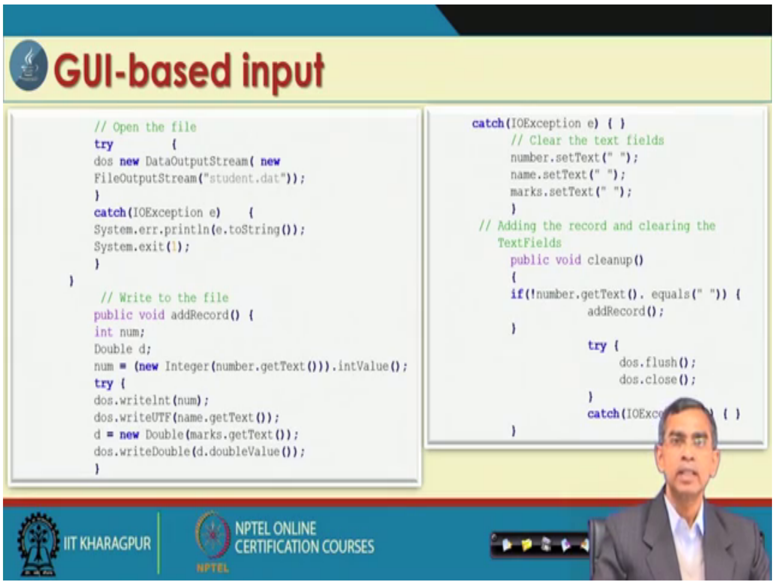
(Refer Slide Time: 33:53)



GUI without Applet

So, it means that in your program, you can write one application look like, it is not an applet actually. But, it looks like an applet, it is basically called the frame or panel, and that panel is basically an windows programmer. So, this are one another version of the windows program, which can come in the form of a AWT programming.

(Refer Slide Time: 34:07)



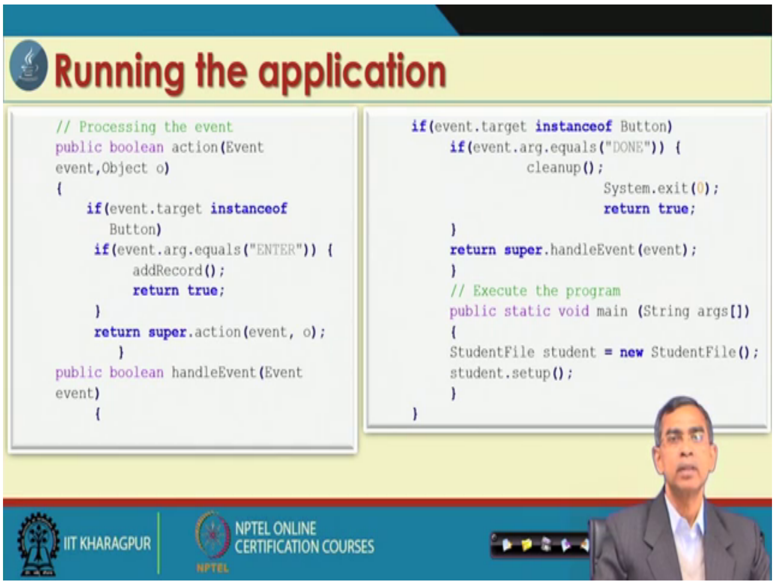
GUI-based input

```
        // Open the file
        try         {
        dos new DataOutputStream( new
        FileOutputStream("student.dat"));
        }
        catch(IOException e)     {
        System.err.println(e.toString());
        System.exit(1);
        }
    }
         // Write to the file
        public void addRecord() {
        int num;
        Double d;
        num = (new Integer(number.getText())).intValue();
        try {
        dos.writeInt(num);
        dos.writeUTF(name.getText());
        d = new Double(marks.getText());
        dos.writeDouble(d.doubleValue());
        }
```

```
    catch(IOException e) { }
          // Clear the text fields
          number.setText(" ");
          name.setText(" ");
          marks.setText(" ");
          }
     // Adding the record and clearing the
        TextFields
          public void cleanup()
          {
          if(!number.getText(). equals(" ")) {
                    addRecord();
          }
                    try {
                        dos.flush();
                        dos.close();
                    }
                    catch(IOExc          ) { }
          }
```

And this is the one AWT code that will be discussed in details, I just want to skip it.

(Refer Slide Time: 34:16)



Running the application

```
    // Processing the event
    public boolean action(Event
    event,Object o)
    {
        if(event.target instanceof
          Button)
         if(event.arg.equals("ENTER")) {
             addRecord();
             return true;
         }
         return super.action(event, o);
            }
    public boolean handleEvent(Event
    event)
         {
```

```
    if(event.target instanceof Button)
         if(event.arg.equals("DONE")) {
                   cleanup();
                             System.exit(0);
                             return true;
         }
         return super.handleEvent(event);
         }
         // Execute the program
         public static void main (String args[])
         {
         StudentFile student = new StudentFile();
         student.setup();
         }
    }
```

And so this way we can run another windows program, other than applets. So, AWT is an alternative to the applet programming actually. And applet regarding AWT programming, we will discussed in our next module onwards. And then will discuss another programming concept in Java the AWT concept.

Thank you very much.