



Session 05.02

Applet Basics

Dr. Ahmed ElShafee

Agenda

1. *
2. *
3. *

Introduction

- An applet is a Java program that runs in a Web browser.
- An applet can be a fully functional Java application.
- important differences between an applet and a standalone Java application:
 - An applet is a Java class that extends the `Java.applet.Applet` class.
 - applet class will not define `main()`.
 - Applets are designed to be embedded within an HTML page
- When a user views an HTML page that contains an applet, the code for the applet is downloaded to the user's machine then executed on his local machine

-
- A JVM is required to view an applet. The JVM can be either a plug-in of the Web browser or a separate runtime environment.
 - The JVM on the user's machine creates an instance of the applet class and invokes various methods during the applet's lifetime
 - Applets have strict security rules that are enforced by the Web browser. The security of an applet is often referred to as sandbox security (like a child playing in a sandbox).
 - Other classes that the applet needs can be downloaded in a single Java Archive (JAR) file.

Life Cycle of an Applet

- Four methods in the Applet class give you the applet framework :
- **init:** This method is intended for whatever initialization is needed for your applet. It is called after the param tags inside the applet tag have been processed.
- **start:** This method is automatically called after the browser calls the init method. It is also called whenever the user returns to the page containing the applet after having gone off to other pages.
- **stop:** This method is automatically called when the user moves off the page on which the applet sits. It can, therefore, be called repeatedly in the same applet.

-
- **destroy:** This method is only called when the browser shuts down normally. Because applets are meant to live on an HTML page, you should not normally leave resources behind after a user leaves the page that contains the applet.
 - **paint:** Invoked immediately after the start() method, and also any time the applet needs to repaint itself in the browser. The paint() method is actually inherited from the java.awt.

My first applet

- **A "Hello, World" Applet**
- The following is a simple applet named HelloWorldApplet.java

```
import java.applet.*;
import java.awt.*;
public class HelloWorldApplet extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString ("Hello World", 25, 50);
    }
}
```

- java.applet.Applet.
- java.awt.Graphics.
- Without those import statements, the Java compiler would not recognize the classes Applet and Graphics, which the applet class refers to.

Invoking an Applet

- An applet may be invoked by embedding directives in an HTML file and viewing the file through an applet viewer or Java-enabled browser
- The <applet> tag is the basis for embedding an applet in an HTML file. Below is an example that invokes the "Hello, World" applet

```
<html>
<title>The Hello, World Applet</title>
<hr>
<applet code="HelloWorldApplet.class" width="320" height="120">
If your browser was Java-enabled, a "Hello, World"
message would appear here.
</applet>
<hr>
</html>
```

Notes

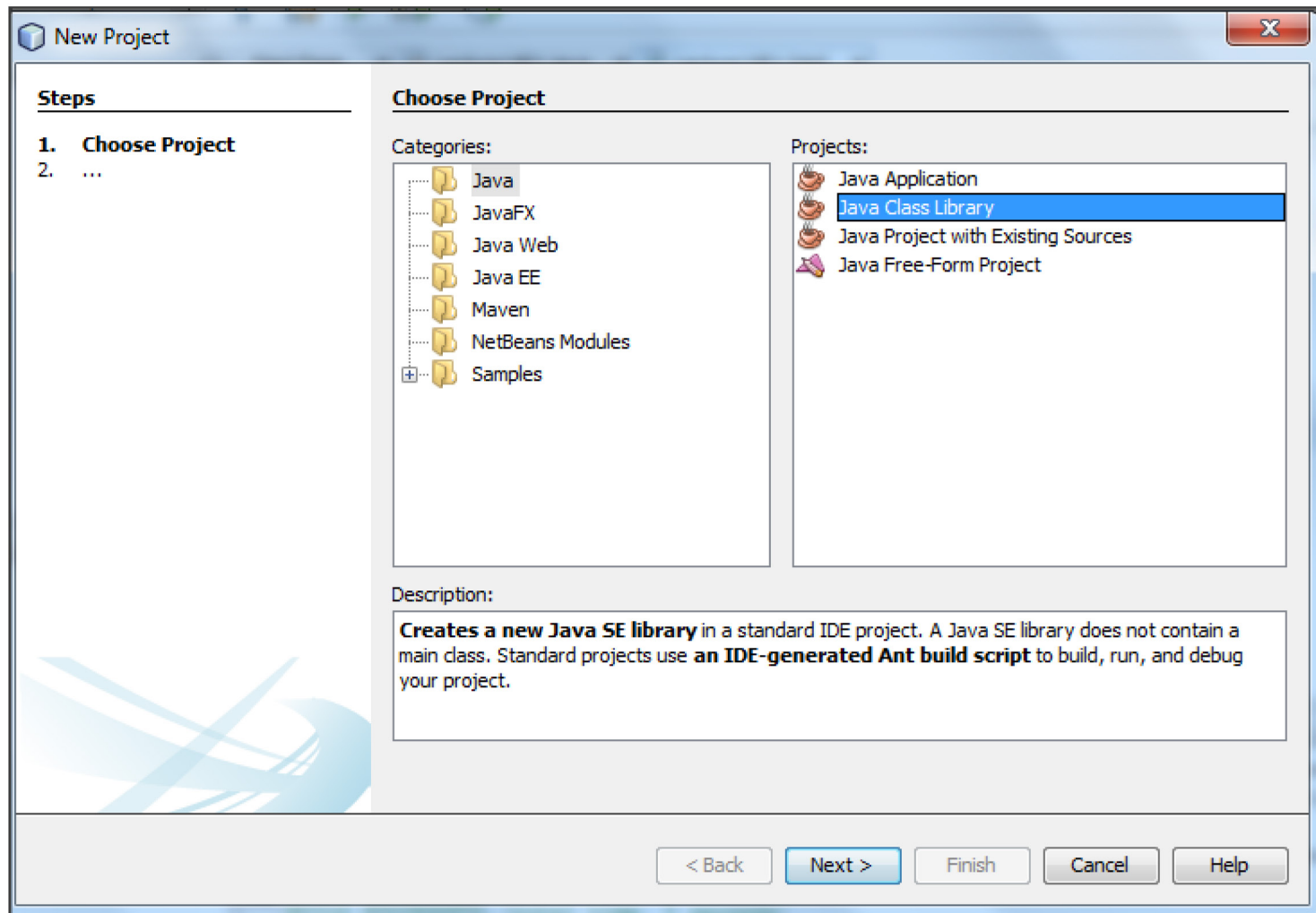
- The code attribute of the `<applet>` tag is required. It specifies the Applet class to run. Width and height are also required to specify the initial size of the panel in which an applet runs.
- The applet directive must be closed with a `</applet>` tag.
- If an applet takes parameters, values may be passed for the parameters by adding `<param>` tags between `<applet>` and `</applet>`.
- The browser ignores text and other tags between the applet tags.
- Non-Java-enabled browsers do not process `<applet>` and `</applet>`.

-
- Therefore, anything that appears between the tags, not related to the applet, is visible in non-Java-enabled browsers.
 - The viewer or browser looks for the compiled Java code at the location of the document.
 - To specify otherwise, use the codebase attribute of the <applet> tag as shown:

```
<applet codebase="http://amrood.com/applets"  
code="HelloWorldApplet.class" width="320" height="120">
```

Build a website using netbeans

- Select java class library



- Select project name/path

New Java Class Library

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

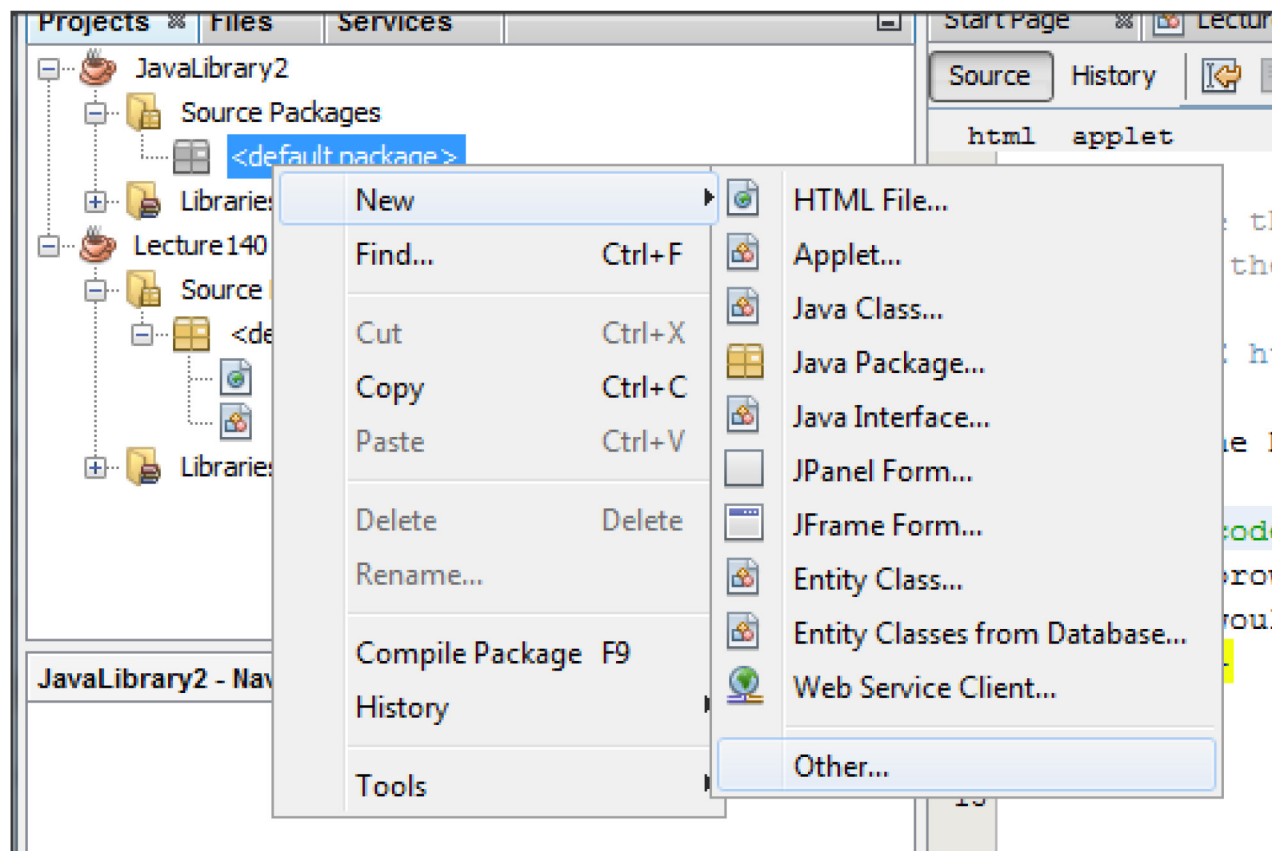
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

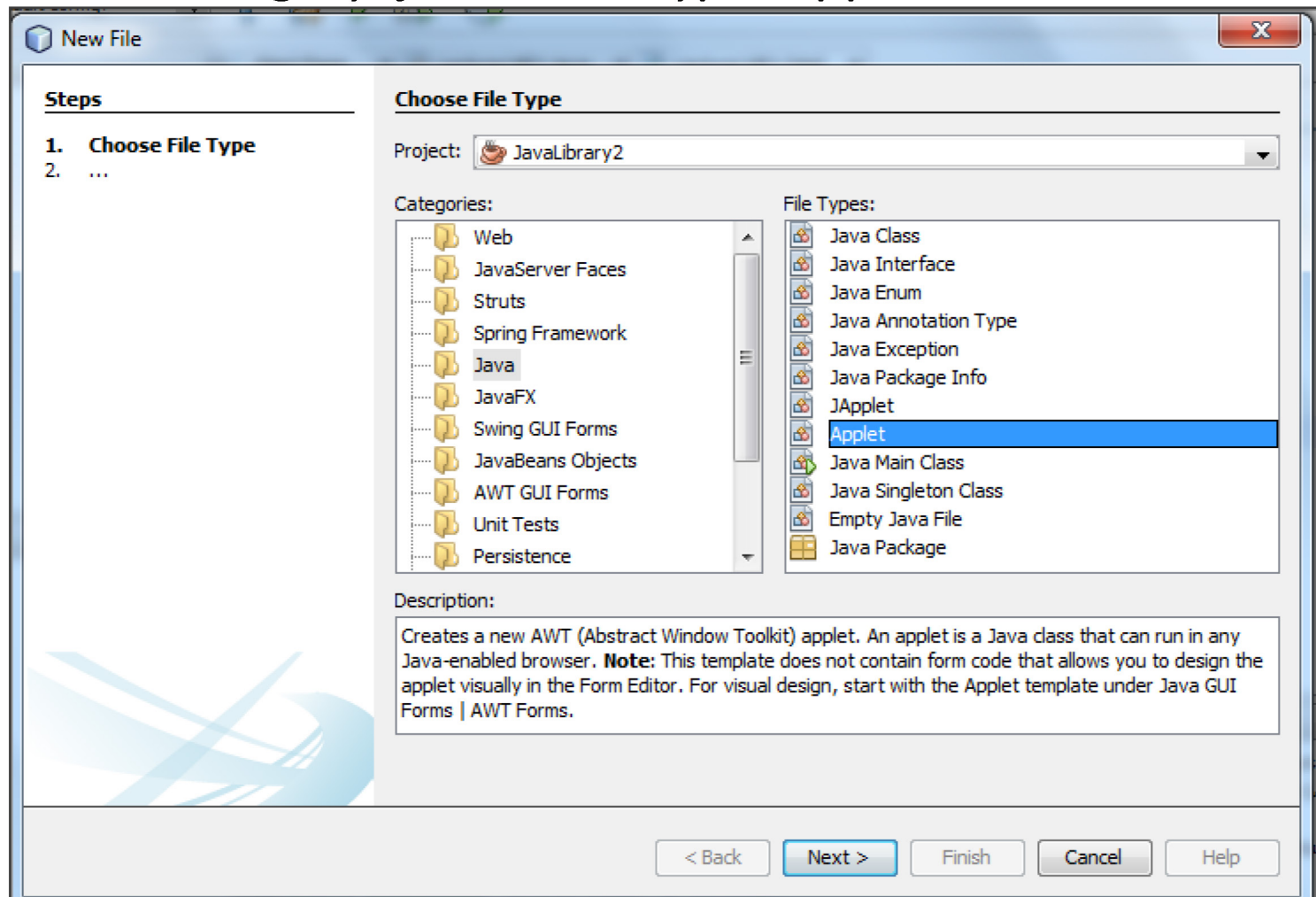
Different users and projects can share the same compilation libraries (see Help for details).

< Back Next > **Finish** Cancel Help

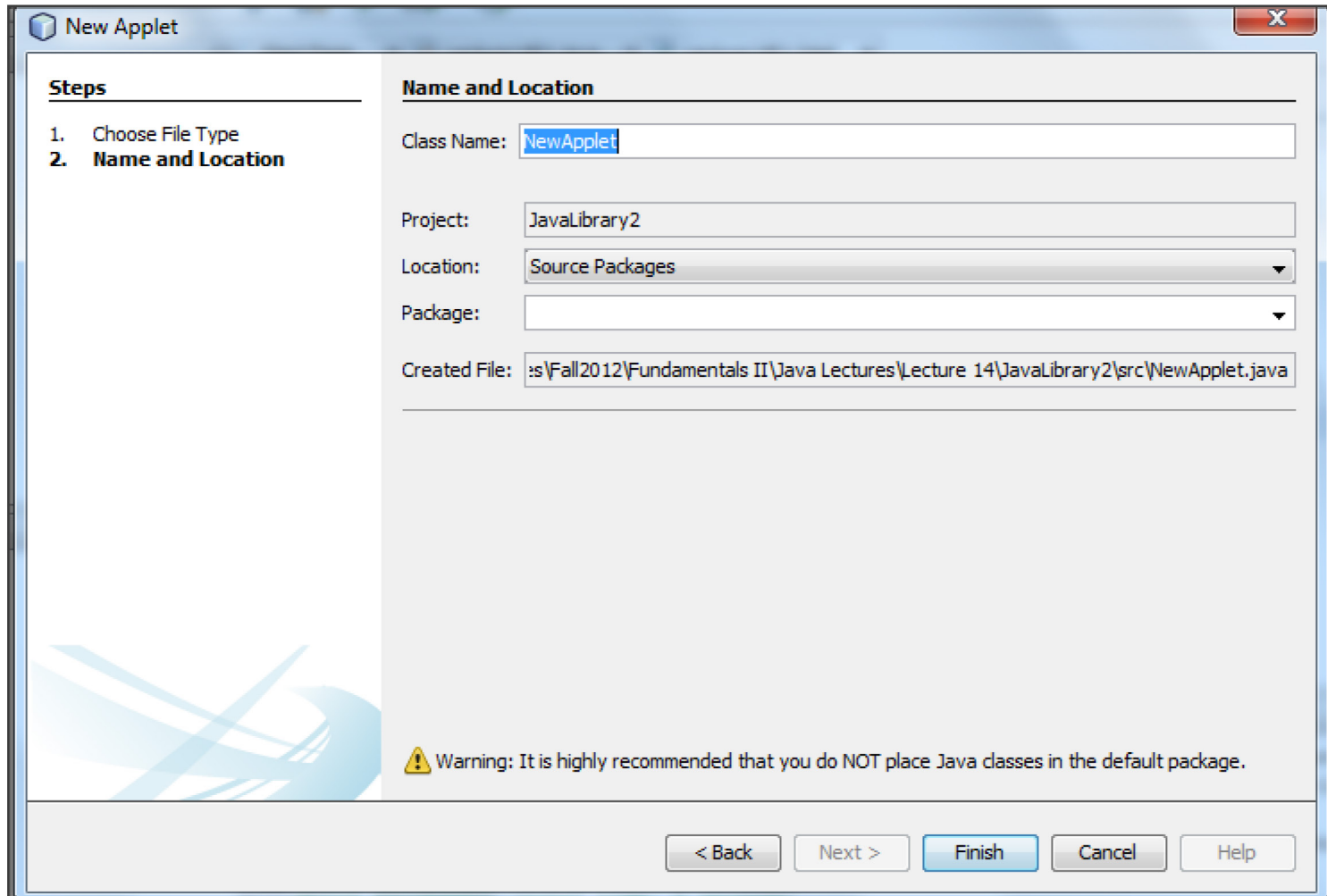
- Add new file to project. Select other



- Select category: java, & file type: Applet



- Type applet name



New Applet

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

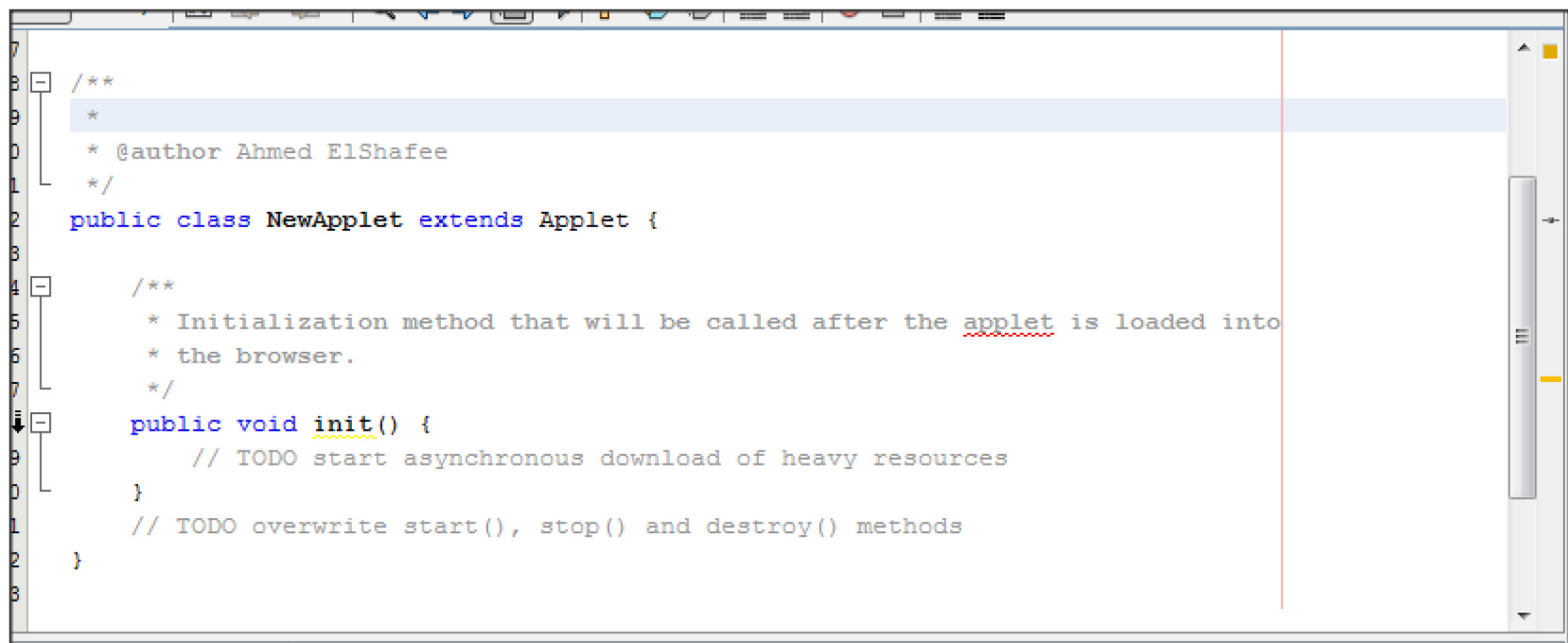
Package:

Created File:

 Warning: It is highly recommended that you do NOT place Java classes in the default package.

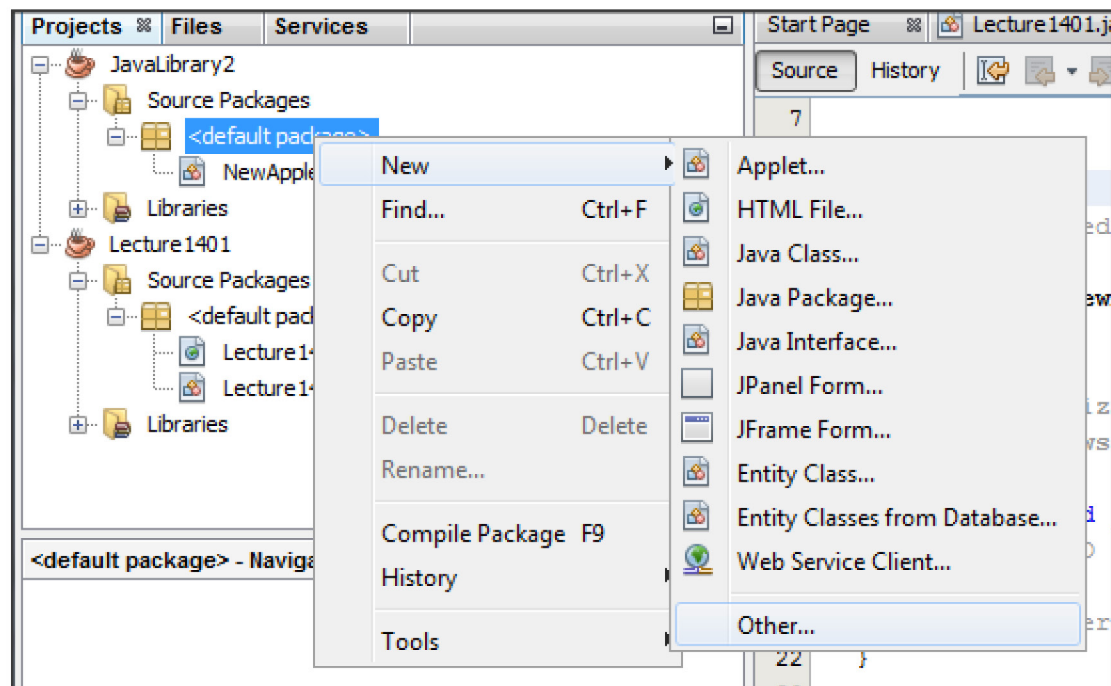
< Back Next > **Finish** Cancel Help

- Add applet code

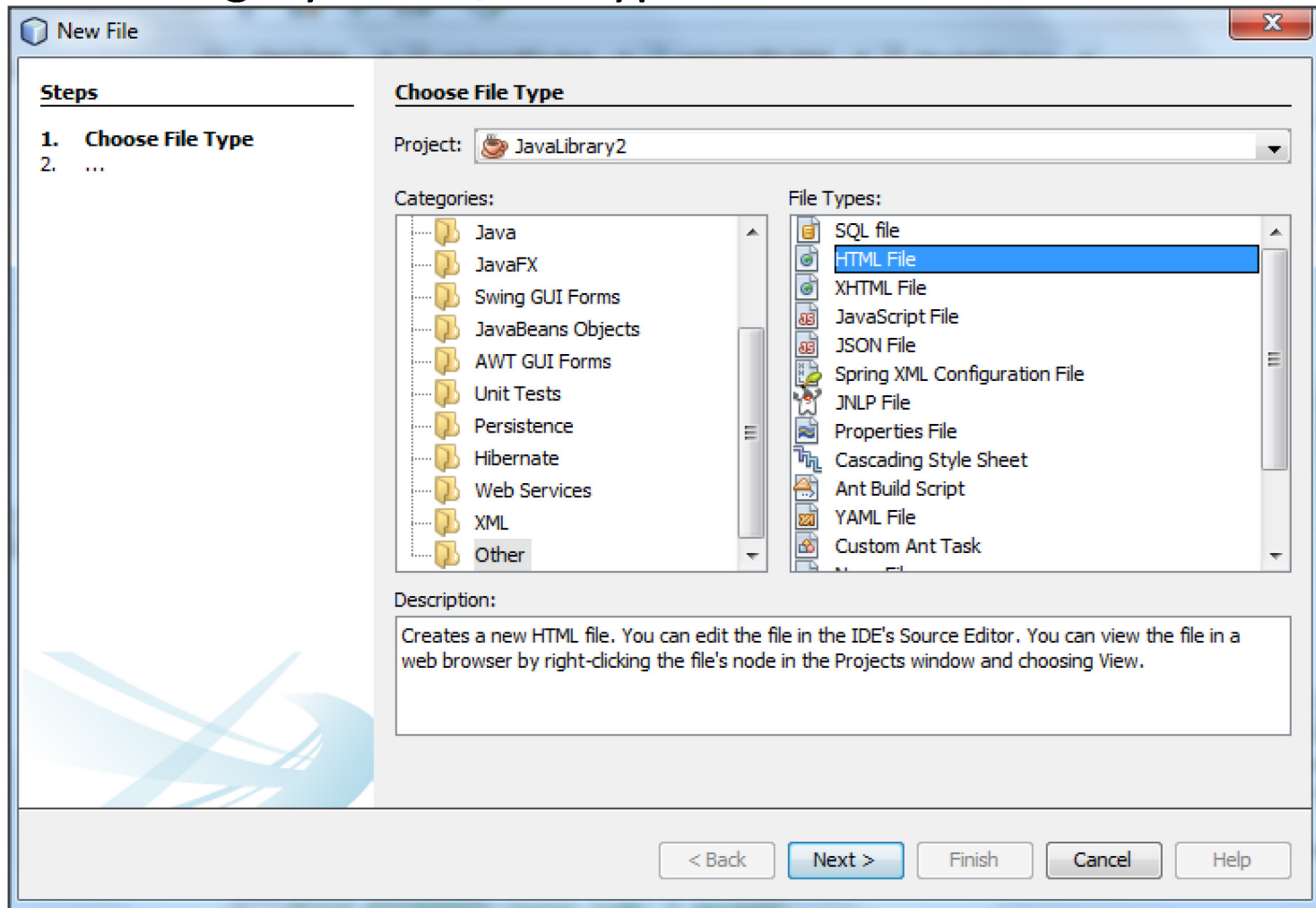
A screenshot of a code editor window with a toolbar at the top. The code is written in Java and defines a class named 'NewApplet' that extends 'Applet'. The code includes a multi-line comment at the top identifying the author as 'Ahmed ElShafee'. Below this, the 'init()' method is defined, with a comment indicating it's for asynchronous download of heavy resources. The 'start()', 'stop()', and 'destroy()' methods are mentioned in a comment as being to be overwritten. The code is color-coded: keywords like 'public', 'class', 'extends', 'void', and 'init' are in blue; the class name 'NewApplet' is in red; and the method name 'init' is in yellow. The editor has a vertical scrollbar on the right and a horizontal scrollbar at the bottom.

```
7
8 /**
9  *
10 * @author Ahmed ElShafee
11 */
12 public class NewApplet extends Applet {
13
14     /**
15      * Initialization method that will be called after the applet is loaded into
16      * the browser.
17      */
18     public void init() {
19         // TODO start asynchronous download of heavy resources
20     }
21     // TODO overwrite start(), stop() and destroy() methods
22 }
23
```

- Add html file to your project. Select new/other



- Select category: other, file type: html



- Type file name and location

New HTML File

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name:

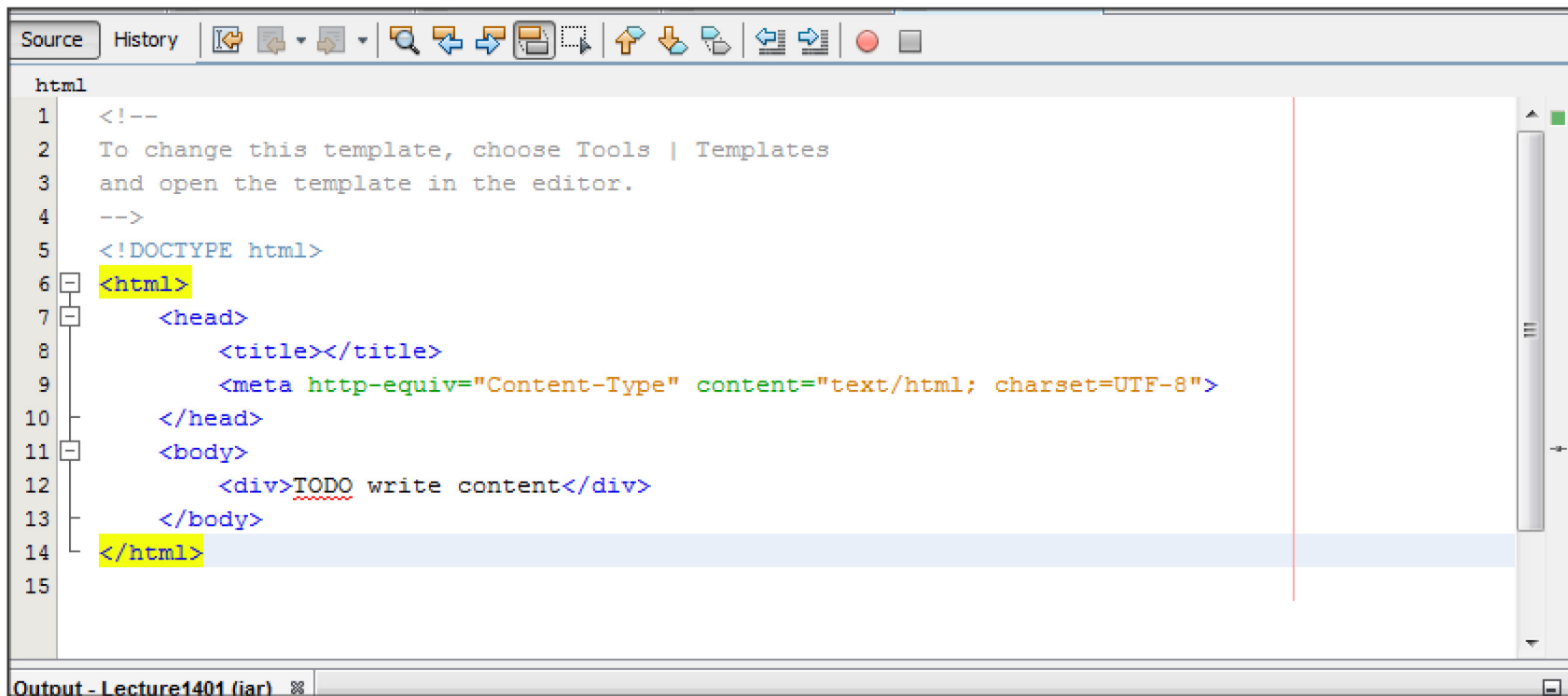
Project:

Folder:

Created File:

< Back Next > **Finish** Cancel Help

- Type code

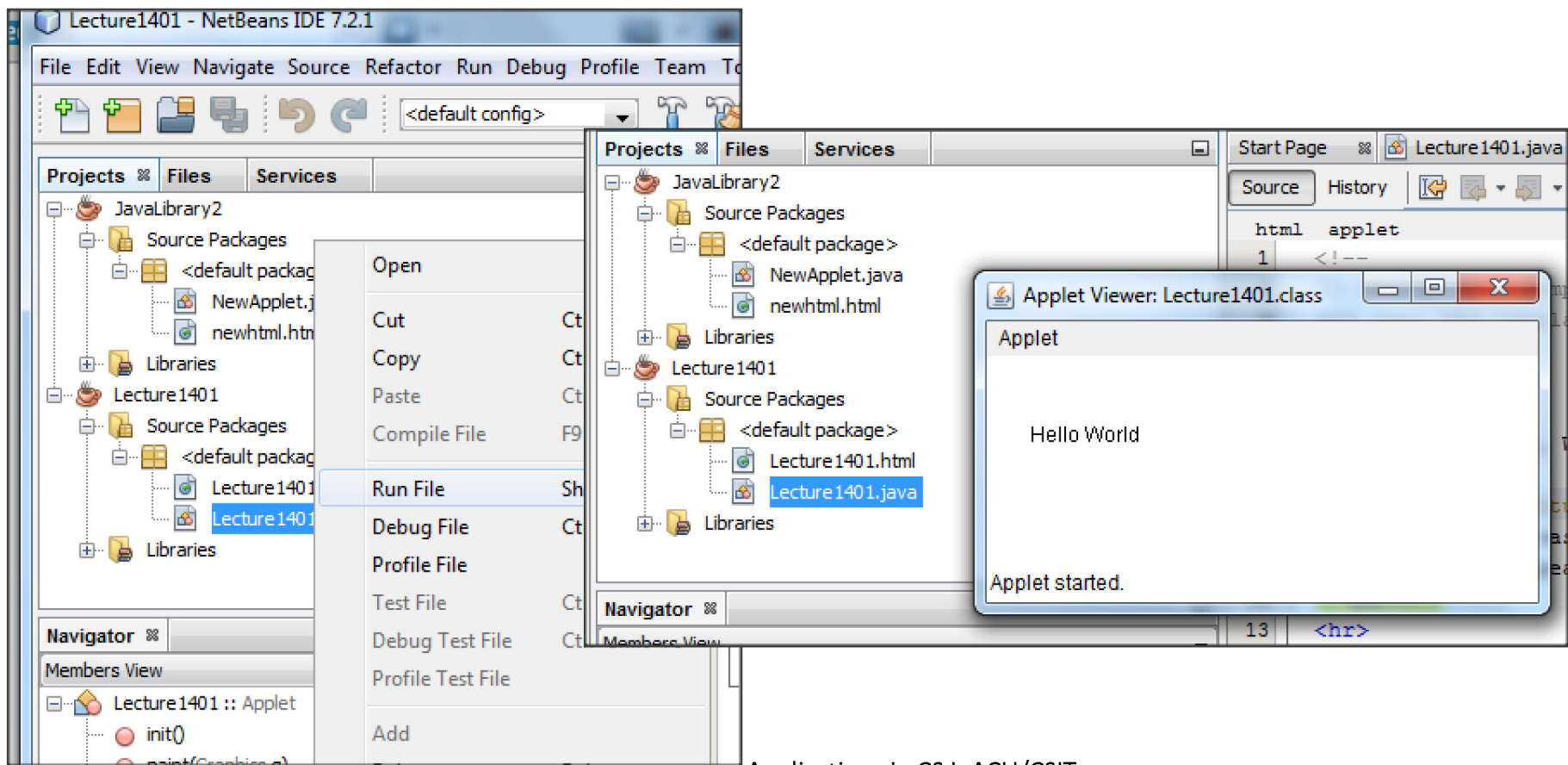


The screenshot shows an IDE window with a toolbar at the top containing icons for file operations (open, save, print, etc.) and editing (undo, redo, search, etc.). The main editor area displays an HTML file named 'html'. The code is as follows:

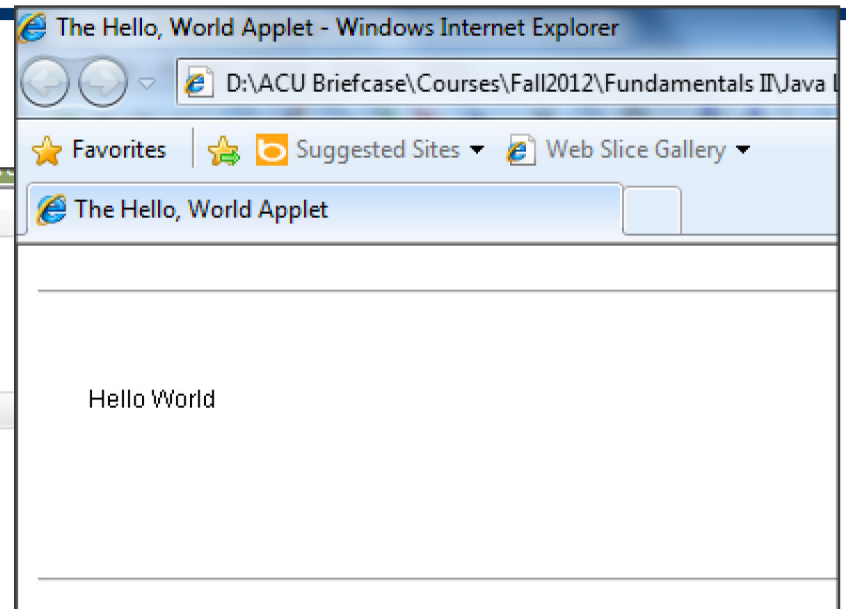
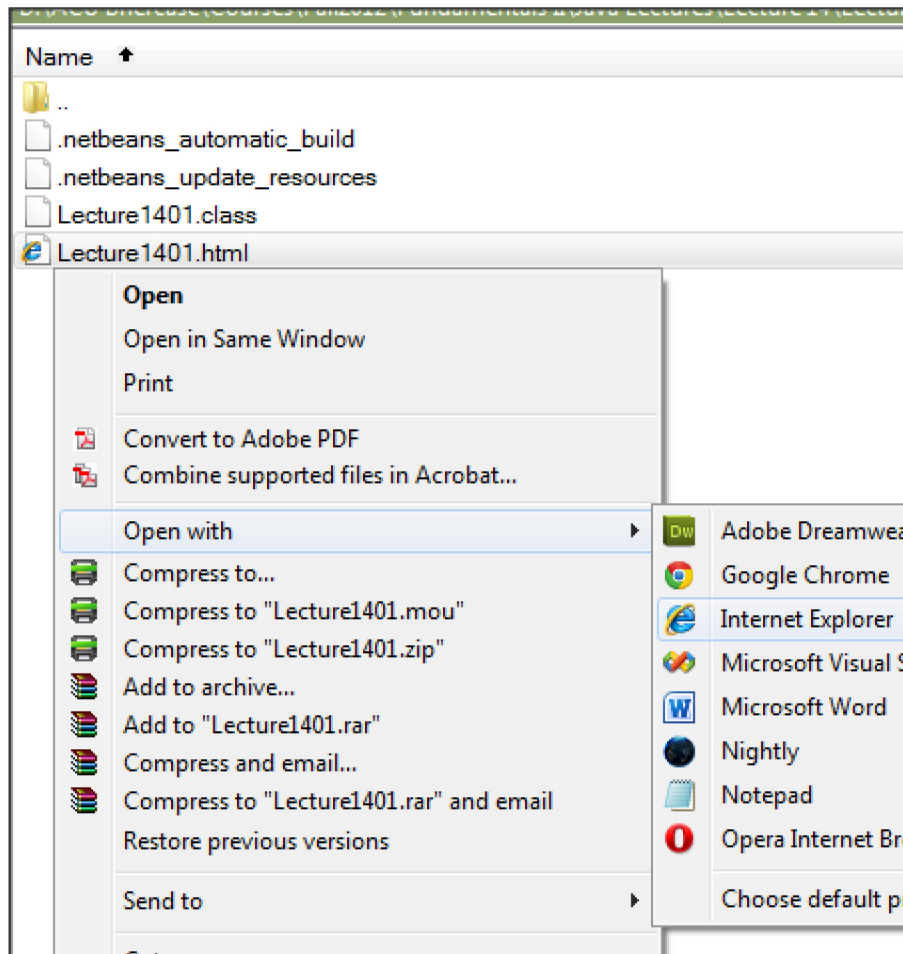
```
1  <!--
2  To change this template, choose Tools | Templates
3  and open the template in the editor.
4  -->
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <title></title>
9          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
10     </head>
11     <body>
12         <div>TODO write content</div>
13     </body>
14 </html>
```

The bottom status bar indicates the current file is 'Output - Lecture1401 (iar)'.

- Run applet



- Run html file



Lecture1301

- Check lab manual

Getting Applet Parameters

- Applet gets its parameters in the `init()` method. It may also get its parameters in the `paint()` method.
- However, getting the values and saving the settings once at the start of the applet, instead of at every refresh, is convenient and efficient.

```
import java.applet.*;
import java.awt.*;
public class CheckerApplet extends Applet
{
    int squareSize = 50; // initialized to default size
    public void init () {}
    private void parseSquareSize (String param) {}
    private Color parseColor (String param) {}
    public void paint (Graphics g) {}
}
```

```

public void init ()
{
    String squareSizeParam = getParameter ("squareSize");
    parseSquareSize (squareSizeParam);
    String colorParam = getParameter ("color");
    Color fg = parseColor (colorParam);
    setBackground (Color.black);
    setForeground (fg);
}
private void parseSquareSize (String param)
{
    if (param == null) return;
    try {
        squareSize = Integer.parseInt (param);
    }
    catch (Exception e) {
        // Let default value remain
    }
}

```

-
- The applet calls `parseSquareSize()` to parse the `squareSize` parameter. `parseSquareSize()` calls the library method `Integer.parseInt()`, which parses a string and returns an integer. `Integer.parseInt()` throws an exception whenever its argument is invalid.
 - The applet calls `parseColor()` to parse the color parameter into a `Color` value

Specifying Applet Parameters

```
<html>  
<title>Checkerboard Applet</title>  
<hr>  
<applet code="CheckerApplet.class" width="480" height="320">  
<param name="color" value="blue">  
<param name="squaresize" value="30">  
</applet>  
<hr>  
</html>
```

Lecture1402

- Check lab manual

Application Conversion to Applets

- It is easy to convert a graphical Java application (that is, an application that uses the AWT and that you can start with the java program launcher) into an applet that you can embed in a web page
- **steps for converting an application to an applet**
 - Make an HTML page with the appropriate tag to load the applet code.
 - Supply a subclass of the JApplet class. Make this class public. Otherwise, the applet cannot be loaded.
 - Eliminate the main method in the application. Do not construct a frame window for the application.
 - Your application will be displayed inside the browser.

- Move any initialization code from the frame window constructor to the init method of the applet.
- You don't need to explicitly construct the applet object. the browser instantiates it for you and calls the init method.
- Remove the call to setSize; for applets, sizing is done with the width and height parameters in the HTML file.
- Remove the call to setDefaultCloseOperation. An applet cannot be closed; it terminates when the browser exits.
- If the application calls setTitle, eliminate the call to the method. Applets cannot have title bars. (You can, of course, title the web page itself, using the HTML title tag.)
- Don't call setVisible(true). The applet is displayed automatically.

Lecture1303

- Check lab manual

Event Handling

- Applets inherit a group of event-handling methods from the Container class. The Container class defines several methods, such as `processKeyEvent` and `processMouseEvent`, for handling particular types of events, and then one catch-all method called `processEvent`.
- Inorder to react an event, an applet must override the appropriate event-specific method

```
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.applet.Applet;
import java.awt.Graphics;
String evt;
    public void init() {
        addMouseListener(this);
        printEvt("initializing the applet ");
    }
    public void start() {
        printEvt("starting the applet ");
    }
    public void stop() {
        printEvt("stopping the applet ");
    }
    public void destroy() {
        printEvt("unloading the applet");
    }
}
```

```

void printEvt(String word) {
    evt=word;
    repaint();
}
public void paint(Graphics g) {
g.drawRect(0, 0,
            getWidth() - 1,
            getHeight() - 1);
g.drawString(evt, 10, 20);
}
public void mouseEntered(MouseEvent event) {
    printEvt("mouse entered");
}
public void mouseExited(MouseEvent event) {
    printEvt("mouse exit");
}
public void mouseClicked(MouseEvent event) {
    printEvt("mouse clicked! ");
}
}

```

-
- Html file

```
<html>  
<title>Event Handling</title>  
<hr>  
<applet code="ExampleEventHandling.class"  
width="300" height="300">  
</applet>  
<hr>  
</html>
```

Lecture1304

- Check lab manual

Displaying Images

- An applet can display images of the format GIF, JPEG, BMP, and others. To display an image within the applet, you use the `drawImage()` method found in the `java.awt.Graphics` class.

```
import java.applet.*;
import java.awt.*;
import java.net.*;
public class NewApplet extends Applet {
    private Image image;
    private AppletContext context;
    public void init() {
        context = this.getAppletContext();
        String imageURL = this.getParameter("image");
        if (imageURL == null) {
            imageURL = "java.jpg";
        }
    }
}
```

```
try {
    URL url = new URL(this.getDocumentBase(), imageURL);
    image = context.getImage(url);
} catch (MalformedURLException e) {
    e.printStackTrace();
// Display in browser status bar
    context.showStatus("Could not load image!");
}

}

public void paint(Graphics g) {
    context.showStatus("Displaying image");
    g.drawImage(image, 0, 0, 200, 84, null);
    g.drawString("www.javalicense.com", 35, 100);
}
}
```

Lecture1305

- Check lab manual

Playing Audio

- An applet can play an audio file represented by the AudioClip interface in the java.applet package. The
- AudioClip interface has three methods, including:
 - **public void play():** Plays the audio clip one time, from the beginning.
 - **public void loop():** Causes the audio clip to replay continually.
 - **public void stop():** Stops playing the audio clip.

```

import java.applet.*;
import java.awt.*;
import java.net.*;
public class NewApplet extends Applet {
    private AudioClip clip;
    private AppletContext context;
    public void init() {
        context = this.getAppletContext();
        String audioURL = this.getParameter("audio");
        if (audioURL == null) {
            audioURL = "default.au";
        }
        try {
            URL url = new URL(this.getDocumentBase(), audioURL);
            clip = context.getAudioClip(url);
        } catch (MalformedURLException e) {
            e.printStackTrace();
            context.showStatus("Could not load audio file!");
        }
    }
}

```

```
public void start() {  
    if (clip != null) {  
        clip.loop();  
    }  
}  
  
public void stop() {  
    if (clip != null) {  
        clip.stop();  
    }  
}  
}
```

```
<html>
<title>The ImageDemo applet</title>
<hr>
<applet code="ImageDemo.class" width="0" height="0">
<param name="audio" value="test.wav">
</applet>
<hr>
</html>
```


Lecture1306

- Check lab manual



Thanks,
See you next Lecture, isA

Dr. Ahmed ElShafee, Practical Applications in CS I, ACU/CSIT
Fall 2013