

CIS 27P

Class 11

Java Overview

- Java 2 Standard Edition (J2SE)
 - Topics covered in CIS27P
- Java 2 Micro Edition
 - Java in small devices
- Java 2 Enterprise Edition
 - Server-side Java

J2ME

- Competing with WAP/IMODE
- Various JAVA VM's for small devices
- Midlets can be downloaded onto device or loaded directly onto it with a wire
- Development is done using emulators

```

/*
 * Copyright 2000-2001 by Sun Microsystems, Inc.,
 * 901 San Antonio Road, Palo Alto, California, 94303, U.S.A.
 * All rights reserved.
 */
package examples.helloworld;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class HelloWorld extends MIDlet implements
    CommandListener {
    private Command exitCommand;
    private TextBox tb;

    public HelloWorld() {
        exitCommand = new Command("Exit", Command.EXIT, 1);
        tb = new TextBox("Hello MIDlet", "Hello, World!", 15, 0);
        tb.addCommand(exitCommand);
        tb.setCommandListener(this);
    }

    protected void startApp() {
        Display.getDisplay(this).setCurrent(tb);
    }

    protected void pauseApp() {}
    protected void destroyApp(boolean u) {}

    public void commandAction(Command c, Displayable d) {
        if (c == exitCommand) {
            destroyApp(false);
            notifyDestroyed();
        }
    }
}

```

J2ME - To Get Started

- Download J2ME Toolkit

<http://java.sun.com/products/j2mewtoolkit/download.html>

- Also:
 - developers.motorola.com
 - www.nokia.com
 - www.nttdocomo.com

J2EE

- Servlets / JSP
 - Containers: Jakarta Tomcat, Allaire Jrun, iPlanet Web Server
- Other J2EE Technologies
 - Includes EJB, JNDI, JDBC, JMS
 - Use application servers: JBoss, BEA Weblogic, IBM WebSphere

J2EE - To Get Started

- Download J2EE Toolkit

<http://java.sun.com/j2ee/download.html>

- Also

- <http://www.theserverside.com>
- <http://jakarta.apache.org/tomcat/>
- <http://www.jboss.org>
- <http://www.bea.com/products/weblogic/server/index.shtml>
- <http://www-4.ibm.com/software/webservers/>

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SimpleCounter extends HttpServlet {

    int count = 0;

    public void doGet
    (
        HttpServletRequest req,
        HttpServletResponse res
    )
    throws ServletException, IOException
    {
        res.setContentType("text/plain");
        PrintWriter out = res.getWriter();
        count++;
        out.println("This servlet has been accessed " +
            count + " times.");
    }
}
```

```

<HTML>
<HEAD>
<TITLE>Form Example</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffcc">
<% if (request.getParameter("name")==null &&
    request.getParameter("email") == null) { %>
<CENTER>
<H2>User Info Request Form</H2>
<FORM METHOD="GET" ACTION="process.jsp">
<P>
Your name: <input type="text" name="name"
    size=26>
<P>
Your email: <input type="text" name="email"
    size=26>
<P>
<input type="submit" value="Process">
</FORM>
</CENTER>
<% } else { %>
<%! String name, email; %>
<%
name = request.getParameter("name");
email = request.getParameter("email");
%>
<P>
<B>You have provided the following info</B>:
<P>
<B>Name</B>: <%= name %><P>
<B>Email</B>: <%= email %>
<% } %>
</BODY>
</HTML>

```

Midterm Concepts

- Byte code
- Virtual Machine
- Object Oriented Design
 - Abstraction
 - Encapsulation
 - Inheritance
 - Polymorphism
- Class structure
- Variables
- Primitive types
- String class

Midterm Concepts

- Arrays
- Operators
- Loops
- Conditionals
- Classes
- Objects
- Instance variables
- Methods
- Access Control

Midterm Concepts

- Inheritance
- static
- final
- abstract
- Constructors
- Exceptions

Final Concepts

- Packages
 - Declaring
 - Using
 - Importing classes
- Interfaces
 - Methods
 - Variables
- Abstract Classes

Final Concepts

- Applets
 - Life cycle
 - init()
 - start()
 - paint()
 - stop()
 - destroy()
 - Requesting repaint()
 - Using with Threads
 - Accepting parameters from APPLET tag

Final Concepts

- Streams
 - byte streams
 - character streams
 - writing to a file
- Collections & Maps
 - Collection interface
 - List interface
 - Iterator, ListIterator
 - Concrete collections
 - Concrete maps

Final Concepts

- AWT
 - Events
 - Event Sources
 - Button
 - TextField
 - Frame
 - Event Listeners
 - Adapter classes
 - Containers
 - Graphics class

Final Format

- Most problems similar to class exercises and homeworks.
- Some problems asking for expected output from given code.
- Some problems asking to identify errors in given code.