

SE 450 Object-Oriented Software Development

Instructor: Dr. Xiaoping Jia

Office: CST 729

Tel: (312) 362-6251

Fax: (312) 362-6116

E-mail: jia@cs.depaul.edu

URL: <http://se.cs.depaul.edu/se450/se450.html>

Requirements

- Prerequisite: **CSC 416**
- Knowledge of object-oriented programming and C++
- Knowledge of HTML and web-browsers.

Topics

- Applets, WWW, HTML
- Classes and interfaces
- Data types and control flows
- AWT, Swing, and animation
- Input and output, collections
- Multi-threaded programming
- Networking
- Object-oriented design
- Introduction to *design patterns* and frameworks

Textbooks

- Primary Text
 - *Object-Oriented Software Development in Java -- Principles, Patterns, and Frameworks*, Xiaoping Jia, 1/ed. Addison-Wesley, 2000
- References
 - *The Java Programming Language*, Ken Arnold, and James Gosling. 2/ed. Addison-Wesley 1997.
 - *The Java Class Libraries --- An Annotated Reference*, Patrick Chan, Rosanna Lee. Vol. 1 & 2. 2/ed. Addison-Wesley 1997

On-Line Reference

- Java API Documentation
- Java Tutorial

Available at <http://java.sun.com>

Software Required

- Operating System
 - Unix (Linux, Solaris, ...)
 - Windows 95/NT/2000
 - MacOS
- Development environment:
 - Java 2 Platform SE (JDK 1.2.x)
- Java-enabled web browser
 - Netscape 4.7 or higher

A Brief History

- January 1996: first official release JDK 1.0
 - Web: applets, security, URL, networking
 - GUI: Abstract Windows Toolkit (AWT)
- February 1997: JDK 1.1
 - Authentication: digital signatures, certificates
 - Distributed computing: RMI, object serialization, Java IDL/CORBA
 - Database connectivity: JDBC
 - Component architecture: JavaBean

A Brief History (cont'd)

- December 1998: Java 2 Platform (JDK 1.2):
 - Standard, Enterprise, and Micro Editions
 - JFC: Swing, Java2D, Java3D
 - Java Cryptography Extension (JCE)
 - Enterprise computing: enterprise JavaBean (EJB), servlets, Java server page (JSP), Jini, XML
 - Java Multimedia Framework (JMF)
 - Embedded systems: KVM, JavaCard
 - Performance enhancement: JIT, HotSpot VM

What's Cool About Java

Simple	Object-Oriented
Statically Typed	Compiled
Architecture Neutral	Multi-Threaded
Garbage Collected	Robust
Secure	Extensible

First Java Application

Java source in Hello.java

```
// Filename: Hello.java

public class Hello {
    /** A Java app prints "Hello from Venus!" */
    public static void main (String args[]) {
        System.out.println("Hello from Venus!");
        /* System.out refers to the standard output */
    }
}
```

Elements of Java Programs

- Class: Hello
- Method: main(...)
- Statement: System.out.println(...)
- Comments:
 - // a comment
 - /* another comment */
 - /** document comment */
- Source file: Hello.java

Compile and Run an App

- To compile:

```
javac Hello.java
```

Generates Hello.class

- To run:

```
java Hello
```

First Java Applet

Java source in HelloFromVenus.java

```
import java.awt.*;
import java.applet.Applet;
public class HelloFromVenus extends Applet {
    public void paint(Graphics g) {
        Dimension d = getSize();
        g.setColor(Color.black);
        g.fillRect(0,0,d.width,d.height);
        g.setFont(new Font("Sans-serif",Font.BOLD,24));
        g.setColor(new Color(255, 215, 0));
        g.drawString("Hello From Venus!", 40, 25);
        g.drawImage(getImage(getCodeBase(),
            "Venus.gif"), 20, 60, this);
    }
}
```

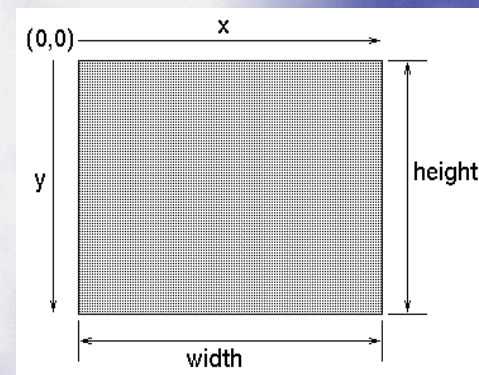
HTML Source

```
<!--HelloFromVenusDemo.html-->
<html><head>
<title> Hello From Venus Applet </title>
</head>
<body bgcolor=black text=white>
<h2>Here is the <em>Hello From Venus</em>
    Applet</h2>
<center>
<applet code="HelloFromVenus.class"
    width=300 height=350>
</applet>
</center>
Venus photo courtesy of NASA.
<hr>
<a href="HelloFromVenus.java">The source.</a>
</body></html>
```

Elements of Java Applets

- Superclass: `java.applet.Applet`
- No `main()` method
- `paint()` method paints the picture
- Applet tag:
 `code width height`

Graphics Coordinate



Compile and Run an Applet

- To compile:

```
javac HelloFromVenus.java
```

Generates HelloFromVenus.class

- To run:

- a) Use the appletviewer from JDK.

```
appletviewer HelloFromVenusDemo.html
```

- b) Open page from browser:

```
HelloFromVenusDemo.html
```