



# Introduction to JAVA

CMSC 331

Fall 2000

UMBC CMSC 331 Java

## Why Java?

- It's the current "hot" language
- It's almost entirely object-oriented
- It has a vast library of predefined objects
- It's platform independent (except for J++)
  - this makes it great for Web programming
- It's designed to support Internet applications
- It's secure
- It isn't C++

UMBC CMSC 331 Java

3

## Introduction

- Present the syntax of Java
- Introduce the Java API
- Demonstrate how to build
  - stand-alone Java programs
  - Java applets, which run within browsers e.g. Netscape
  - Java servlets, which run with a web server
- Example programs tested using Java on Windows 98 and/or Unix

UMBC CMSC 331 Java

2

## Important Features of Java

- Java is a *simple* language (compared to C++).
- Java is a completely *object-oriented* language.
- Java programs can be *multi-threaded*.
- Java programs automatically *recycle memory*.
- Java is a *distributed* and *secure* language.
- Java is *robust* (potential errors are often caught).
- To make Java *portable*, so that they run on a variety of hardware, programs are translated into *byte code* which is *executing by a Java Virtual Machine*.

UMBC CMSC 331 Java

4

## Historical note

- In 1991, a group led by James Gosling and Patrick Naughton at Sun designed a language (code-named “Green”) for use in consumer devices such as intelligent TV “set-top” boxes and microwaves.
- The design choices made reflect the expectation that the language would be used to implement small, distributed, and necessarily robust programs on a variety of hardware.
- No customer was ever found for this technology.
- The language was renamed “Oak” (after a tree outside Gosling’s office) and was used to develop the *HotJava* browser, which had one unique property: it could dynamically download programs (“applets”) from the Web and run them.
- “Oak” was already taken as a name for a computer language, so Gosling thought of the name Java in a coffee shop.

UMBC CMSC 331 Java

5

## Applets, Servlets and applications

- An *applet* is a program designed to be embedded in a Web page and run in a web browser
  - Applets run in a *sandbox* with numerous restrictions; for example, they can’t read files
- A *Servlet* is a program which runs in a web server and typically generates a web page.
  - Dynamically generated web pages are important and Java servlets are an alternative to using Basic (ASP), Python, specialized languages (PHP), and vendor specific solutions (e.g., Oracle)
- An *application* is a conventional program
- Java isn’t a baby language anymore!

UMBC CMSC 331 Java

7

## What is OOP?

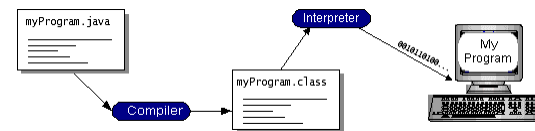
- Object-oriented programming technology can be summarized by three key concepts:
  - Objects that provide **encapsulation** of procedures and data
  - Messages that support **polymorphism** across objects
  - Classes that implement **inheritance** within class hierarchies

UMBC CMSC 331 Java

6

## Building Standalone JAVA Programs (on UNIX)

- Prepare the file `myProgram.java` using any editor
- Invoke the compiler: `javac myProgram.java`
- This creates `myProgram.class`
- Run the java interpreter: `java myProgram`

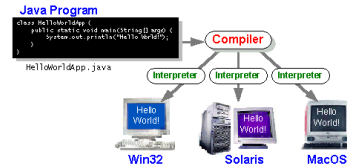


UMBC CMSC 331 Java

8

## Java Virtual Machine

- The .class files generated by the compiler are not executable binaries
  - so Java combines compilation and interpretation
- Instead, they contain “byte-codes” to be executed by the Java Virtual Machine
  - other languages have done this, e.g. UCSD Pascal, Prolog
- This approach provides platform independence, and greater security



## HelloWorld Application

```
=> cd java
=> ls
HelloWorld.java
=> more HelloWorld.java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
=> javac HelloWorld.java
=> java HelloWorld
Hello World!
=>
```

## HelloWorld Application

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

- Note that String is built in
- println is a member function for the System.out class
- Every standalone Java application must have a main method like

```
public static void main(String[] args) { }
```

## Java Applets

- The JAVA virtual machine may be executed under the auspices of some other program, e.g. a Web browser or server.
- Bytecodes can be loaded off the Web, and then executed locally.
- There are classes in Java to support this

## Building Applets

- Prepare the file `myProgram.java`, and compile it to create `myProgram.class`
- Invoke an Applet Viewer, or a Java-aware browser such as Netscape, and open an HTML file such as `myProgram.html`
- Browser invokes the Java Virtual Machine

## hello.html

```
<title>Hello, World</title>
<h1>Hello, World</h1>
<applet code="HelloWorld.class"
        width=100 height=140>
</applet>
```

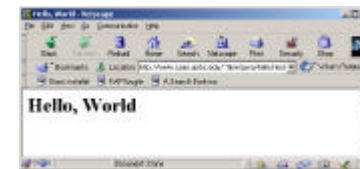
## HelloWorld.java

```
import java.applet.*;

public class HelloWorld extends Applet {
    public void init() {
        System.out.println("Hello, world!");
    }
}
```

## Running the Applet

```
[3:43pm] linuxserver1 => pwd
/home/faculty4/finin/www/java
[3:43pm] linuxserver1 => ls
HelloWorld.java hello.html
[3:43pm] linuxserver1 => javac HelloWorld.java
[3:43pm] linuxserver1 => ls
HelloWorld.class HelloWorld.java hello.html
```



## Java Servlets

- Most interesting web applications provide services, which requires invoking **programs**.
- More and more of the web consists of pages that are not statically created by human editors, but dynamically generated when needed by **programs**.
- How do we invoke these programs and what programming languages should we use?
  - CGI: Common Gateway Interface
  - Web servers with built in support for servlets written in Python, Lisp, Tcl, Prolog, Java, Visual Basic, Perl, etc.
  - ASP (Active Server Pages) is a scripting environment for Microsoft Internet Information Server in which you can combine HTML, scripts and reusable ActiveX server components to create dynamic web pages.
  - ASP begat PHP, JSP, ...
- Java turns out to be an excellent language for servlets

## Why Build Web Pages Dynamically?

- **The Web page is based on data submitted by the user**
  - Eg., results page from search engines and order-confirmation pages at on-line stores
- **The Web page is derived from data that changes frequently**
  - Eg., a weather report or news headlines page
- **The Web page uses information from databases or other server-side sources**
  - Eg., an e-commerce site could use a servlet to build a Web page that lists the current price and availability of each item that is for sale.

## A Servlet's Job

- **Read any data sent by the user**
  - From HTML form, applet, or custom HTTP client
- **Look up HTTP request information**
  - Browser capabilities, cookies, requesting host, etc.
- **Generate the results**
  - JDBC, RMI, direct computation, legacy app, etc.
- **Format the results inside a document**
  - HTML, Excel, etc.
- **Set HTTP response parameters**
  - MIME type, cookies, compression, etc.
- **Send the document to the client**

## The Advantages of Servlets Over “Traditional” CGI

- **Efficient**
  - Threads instead of OS processes, one servlet copy, persistence
- **Convenient**
  - Lots of high-level utilities
- **Powerful**
  - Sharing data, pooling, persistence
- **Portable**
  - Run on virtually all operating systems and servers
- **Secure**
  - No shell escapes, no buffer overflows
- **Inexpensive**

## Simple Servlet Template

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

public class ServletTemplate extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        // Use "request" to read incoming HTTP headers
        // (e.g. cookies) and HTML form data (query data)

        // Use "response" to specify the HTTP response status
        // code and headers (e.g. the content type, cookies).

        PrintWriter out = response.getWriter();
        // Use "out" to send content to browser
    }
}
```

UMBC CMSC 331 Java

21

## Summary

- Java is an object-oriented programming language.
- Java features make it ideally suited for writing network-oriented programs.
- Java class definitions and the programs associated with classes are compiled into byte code to facilitate program portability.
- Java class definitions and the programs associated with them can be loaded dynamically via a network.
- Java programs can be multithreaded, thereby enabling them to perform many tasks simultaneously.
- Java does automatic memory management, relieving you of tedious programming and frustrating debugging, thereby increasing your productivity.
- Java has syntactical similarities with the C and C++ languages.

UMBC CMSC 331 Java

23

## HelloWorld Servlet

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

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        out.println("Hello World");
    }
}
```



UMBC CMSC 331 Java

22