

Programming

TALKING SO COMPUTERS UNDERSTAND YOU

What is Java?

- Java is probably one of the most controversial, confused, and misused terms in the entire field of Computer Science. Sun Microsystems, the developers of Java, hoping to alleviate this confusion, have provided a simple definition; Java is :
- **"A Simple, Object-oriented, Distributed, Interpreted, Robust, Secure, Architecture-neutral, Portable, High-performance, Multithreaded, Dynamic Programming Language"**

- A well-known Indian legend tells the story of seven blind men describing an elephant. The one who felt the trunk thought the elephant was like a snake, the one who felt the legs thought the elephant was like a tree, and so on.
- Java is a little bit like that; you'll often hear people say things like:
 - "Java is slow" or, less often, "Java is as fast as C++"
 - Java programs look "strange"
 - Java is better for building Web applications than Microsoft .NET
 - Microsoft .NET supports Java along with other programming languages
 - Java is cross-platform while VB-script only works in Internet Explorer

- Each of these statements, except the last, contains a kernel of truth; each also suffers from confusion between the three parts of the Java platform. In reality, Java is:
 - an object-oriented **programming language**.
 - an interpreter-based runtime system called the **Java Virtual Machine** or **JVM**. There are several different kinds of JVM's. Some use an interpreter, some use a Just In Time compiler [JIT].
 - a set of universally distributed classes called the **Java Class Libraries**. The class libraries provide support for GUI programming, network and file I/O, etc.
- The combination of all three of these are called a Java platform. The most commonly used Java Platform is supplied by Sun [called the JRE], but there are also versions supplied by Apple, Microsoft, IBM, HP, and so on.

Types of Java Programs

- There are three major types of Java programs:

1. Applets:
 - programs that are "hosted" inside a Web page.
 - The compiled machine code for your program is stored on a Web server.
 - The code is automatically downloaded when a user visits your Web page.
 - The user's Web browser loads a Java Virtual Machine when it encounters a Java applet, and then runs the applet inside the browser's JVM. This is similar to the way Flash/Shockwave animation, or other plugins are handled.
 - For safety sake, Java applets are prohibited from performing actions that might damage a user's machine if used maliciously. Applets cannot read and write files on the user's local machine, for instance.

2. Applications

- programs that are installed and run on the user's local machine. There are two kinds of Java applications :
 - **Console-mode applications** - These are traditional "teletype-style" programs similar to those you'd write in a Pascal or C++ class.
 - **GUI applications** - These graphical programs use windows, buttons, and mice to create the kind of interactive programs you're used to.
- Java actually contains two different GUI libraries, the original AWT [Abstract Window Toolkit] which uses your computer's native look-and-feel, and the JFC [Java Foundation Classes], which provides a set of advanced components that can be made to look and feel the same on all platforms.

- Servlets:
 - Also called a server-side application.
 - Server-side applications are not downloaded or installed on the user's computer. Instead, the Java program runs on an Application Server, and only the input or output is sent to the user's machine.
 - Server-side applications are most often accessed through a Web browser, and are used for ecommerce and other interactive Web sites. We won't do any server-side Java in this course.

Why Java is important

- Java is a cross-platform binary portable language.
- A portable language is one that is designed to run on different computer operating systems or different computer hardware. The C language is portable, for instance, because you can run the same program on both your Windows computer and your Macintosh, by taking your C-language source code and recompiling it with a different C compiler on each machine.
- This is called source code portability, and it only works if:
 - there is a compiler or interpreter available on each platform.
 - the program uses no platform-dependent facilities.
- C and C++ were designed to be portable. Languages like Visual Basic, or AppleScript, or Intel assembly language, were not.
- Java programs are designed to offer more than source-code portability; Sun's goal for the Java Platform is summed up in the "Pure Java" refrain called WORA, or "Write Once: Run Anywhere"

- With WORA, you should be able to compile your Java program on a Mac and then run it, with no changes at all, on a Windows or Unix computer. This is possible because the JVM installed on each platform understands the same byte-code, and because every Java platform has the same classes available.
- The major problem with Java is that "WORA" really isn't perfect; each implementation of the Java platform has subtle differences, and so developers ended up with: "Write Once: Test Everywhere"
- There are several reasons for this: AWT GUI performance depends on native peers. In other words, Button objects on the Mac act differently from Button objects on Windows or Unix. Sun changed the classes in the platform but the different browsers didn't keep up.
- Sun and Microsoft exhibited a failure to "play well together."
- In addition to the WORA failings, Java speed is still definitely an issue. Java GUI apps use more memory and perform more slowly than C++ programs.

What is a programming language?

- programming language(Definition)
 - A programming language is a language designed to help you structure your instructions to a computer in a format that you can easily read and understand (relatively speaking).
 - Programming is really just a way of organizing a task so that it is replicable by something else -- a computer. If you have ever given someone directions, or written down a recipe, you have some experience with programming.

What is an algorithm

- algorithm(Definition):
 - An algorithm is a recipe that's intended to be used over and over again by lots of people, to solve a general problem.
 - In cooking terms, this is sort of like instructions for making good vegetable soup stock. There may be many different implementations of these instructions (even variations on them), but the instructions lay out a basic, and important concept that many people will want to use.
 - Similarly, an algorithm might be implemented in many different programming languages, and slight variations on how it is implemented will always exist, but at heart it always describes the same concept.

- There are *many, many* different programming languages out there. Java is a powerful and popular language that you are likely to encounter more than once in the years ahead, and well suited to learning about programming.
- One important characteristic of Java is that it organizes programs conceptually around *objects*. Although not all programming languages use objects, you will see during this class that they are central to how we write programs in Java.

What is an Object?

- An object is an abstraction for a grouping of information and skills that have a coherent relationship to each other. Because it's a program, each object typically has a very limited and specific set of skills (things it can do). An object may also contain information (things it knows about).
- In computer science terms, an object's skills are called *methods* and an object's information is stored in *fields*.

- You will be using objects over and over again. As an exercise right now, let's think of something in your kitchen that has a very specific function.
- For example, your kitchen probably has a refrigerator. A "Fridge" object might store information about what groceries it contains. It could have methods such as *isSpoiled()* which tells you whether a specific grocery item is spoiled.

Learn by Doing

- Write a short paragraph describing an object in your kitchen, including what information (if any) is stored in it, and what methods you think it might have.

Our Answer

- There are an endless number of correct answers to this question.
- Here is one example:
 - A sink is useful for cleaning, soaking, and (in some homes) storing dirty dishes. It might have data such as
 - numDirtyDishes , cupsWater , isSoapy (if the water in it is soapy and isClogged).
 - Possible methods include
 - closeDrain(), openDrain(), hotWaterStart(), hotWaterEnd(), coldWaterStart(), coldWaterEnd() and so on.

- As this class goes on, you may find yourself wondering why we can't use something more amenable to people, a language like English, perhaps. There are two problems with this:
 - First, for a set of instructions to be truly reproducible, they must be well specified. This is why even cooking recipes are not written in pure English (open a cookbook, and you'll see a very recognizable format, with ingredients, measurements, and so on).
 - Second, computers are not smart enough to understand English yet -- you have to help bridge the gap between what the computer really wants to see (described next) and what you really want to say ("go implement my homework please").

Learn by Doing

- Write down instructions for how to make your favorite breakfast. Tomorrow morning, try to follow the instructions provided for making your favorite breakfast!

Our Answer:

- Your explanation was probably in English, which is fine. Here we demonstrate what an explanation might look like written in pseudo code (something similar to Java but that hasn't really been implemented). Note that the use of intuitive names makes this code very legible!

```
oven.setTemp(350)
freezer.open()
while (oven.getTemp() < 350)
...wait...
oven.open()
oven.setContents(bagel)
oven.close()
Time t = Time.now()
while(Time.now() - t < 10min)
...wait...
```

```
Oven.open()
Bagel = oven.getBagel()
Knife breadKnife = draw.getKnife()
Bagel.slice(breadKnife)
Cheese creamcheese =
  fridge.getCreamCheese()
creamcheese.open()
Bagel.spread(knife, creamcheese)
Eat up!
```