Java Programming

Outline

- C, C++, and Java
- How Java is related to C and C++
- Advantages of Java
- Writing good codes in Java is easier than in C or C++
- Java libraries (packages)
- Where the power of Java come from

C, C++, and Java

- C designed in 1970’s for operating system programming, i.e. UNIX
  - High-level language compared with assembly language
  - Low-level compared with C++ or ja programming, i.e. UNIX
  -
  - Low level = Easy manipulation of hardware
  - High level = Relying one lower level software to implement task with minimum amount of code.

Strengths

- Not restrictive, especially in relation to type conversion.
- More efficient than C++ and Java.
- Weaknesses:

Not object-oriented: does not support abstraction and encapsulation

Difficult to manage in large projects.

Little built-in functionalities. Programmer need to start from scratch.

Easy to make mistakes. (pointers, a=b vs a==b)

C, C++, and Java

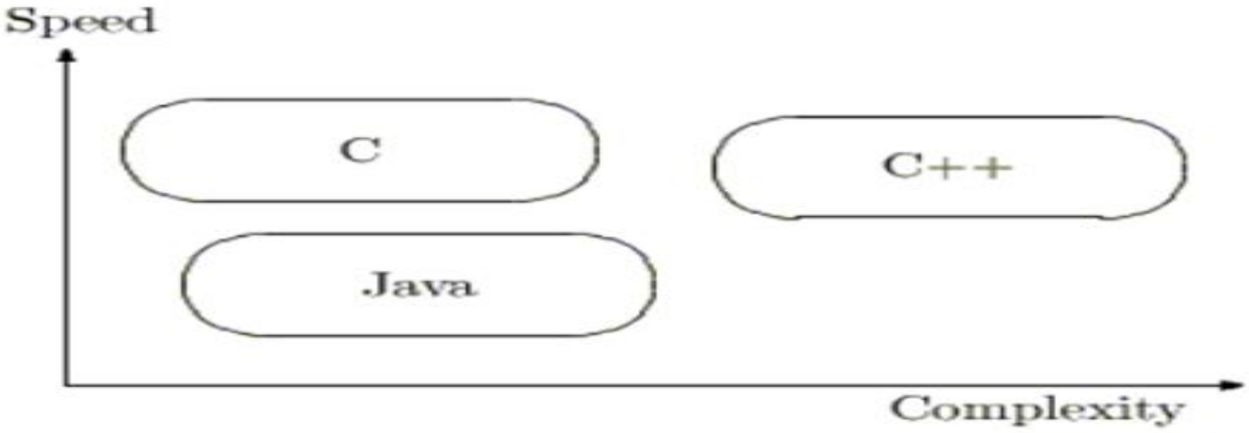
- C++ designed in 1980’s, complete superset of C
- Changes
  - Support for Object-Oriented programming
  - …
- Strengths
  - Improved data abstraction and encapsulation

    Makes it easier to manage large projects
  - More extensive built-ins (standard libraries)
- Weakness
  - Considered by many to be over-complicated
  - Contains all of C’s problems

C, C++ and Java

- Java: a language of 1990's
- The design of Java starts with C syntax and semantics
- Adds a few features from C++: Objects, exceptions
- Leaves out parts unneeded, unsafe, complex (not backward compatible)
  - Gosling: "Java omits many rarely used, poorly understood, confusing features of C++ that in our experience bring more grief than benefits."
- Adds a few facilities not present in C or C++
  - Garbage collection, concurrency, runtime error checking, object serialization, interface, inner classes, threads
- Strengthens portability and security
-

C, C++ and Java



Outline

- C, C++, and Java
  - How Java is related to C and C++
- Advantages of Java
  - Writing good codes in Java is easier than in C or C++
- Java libraries (packages)

Where the power of Java come from

- Java and the internet
  - What Java is good for
- Course content

Advantages of Java

- According to Sun’s Java White paper:

  “Java is a simple, objected-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multi-threaded, and dynamic language”.

- Most of the claims are justified, while others are controversial.

Advantages of Java/Simple

- ❖ Streamlined C language:
  - ○No typedef, union, enum, goto, comma operator
  - ○No header files
    - ■C: list.h, list.c
    - ■Java: List.java
- ❖ No makefile
  - ■Java compiler can figure out dependencies among classes

Advantages of Java/Simple

❖ Fixed problematic areas:

  ○No pointers, no function pointers

  ○Add garbage collection. Memory leaks no more

  ○No code like this

```
       *Head == NULL ){
      *Head=(NODESET *) memAllocate(sizeof(NODESET));
      *(Head+5) = tmpNode;
      (*Head)->next=(*Head)->prev=NULL;
    }
memDeallocate( Head );
```

❖ “if (a=b)” does not compile

Advantages of Java/Simple

- Everything is class except several primitive types
  - Array is a class. Cannot go over bound.
    - No this
    - int a[5]; a[5]=0;
- However, Java libraries are quite complicated

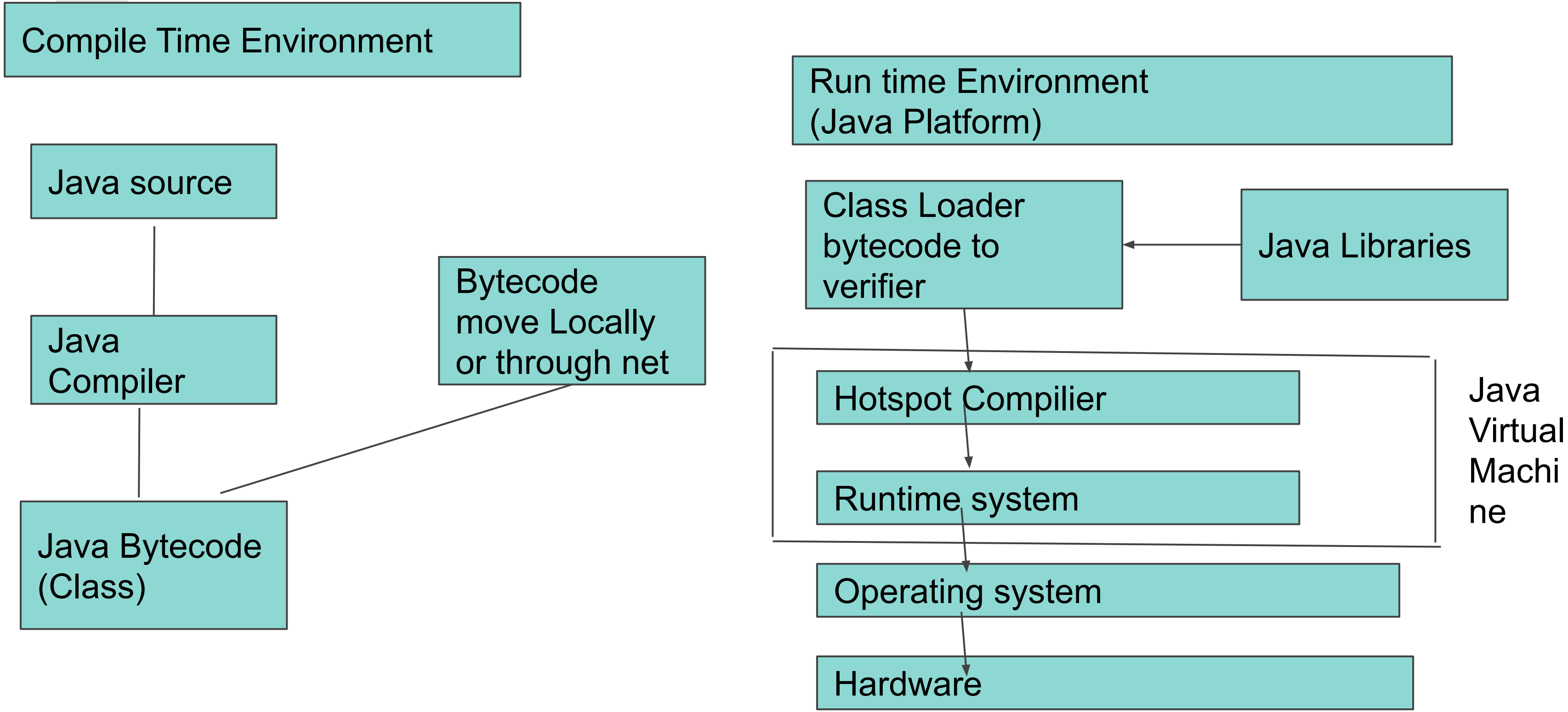
Advantages of Java/Object-Oriented

- ❖ More so than C++.
    - ○Combine data and behavior into one unit, the object
    - ○Programs are collections of interacting, cooperating objects
- ❖ Advantages of OOP
    - ○Provide strong data abstraction and encapsulation
    - ○Gives framework for design
    - ○Allows independent development and testing
    - ○Facilitates system extensions and maintenance
    - ○More opportunity for code re-use

Advantages of Java/Platform-Independen

- C and C++ programs are compiled into object code,
  - ○Object code is directly processed by hardware processor.
  - ○Require a separate compiler for each computer platform, i.e. for each computer operating system and the hardware set of instructions that it is built on.
- Java programs are compiled into bytecode
  - ○Bytecode is processed by a program called java virtual machine (JVM), rather than by the "real" computer machine, the hardware processor.
  - ○Platform differences dealt with by JVM
  - ○Consequently, Java programs can run on any platform. “Write once, run anywhere”.

Java Virtual Machine



Compile Time Environment

Java source

Java Compiler

Java Bytecode (Class)

Bytecode move Locally or through net

Run time Environment (Java Platform)

Class Loader bytecode to verifier

Java Libraries

Hotspot Compilier

Runtime system

Java Virtual Machi ne

Operating system

Hardware

Advantages of Java/Robust and Secure

- Fewer language loopholes
  - No pointers, typecasts limited. (Easier to produce error free source code)
- Compilers are strict
  - Require initialization of variables, enforces type consistency, requires prototypes (Less chance of error in compiled code code)
- Runtime error checking
  - Array bounds, null reference access (Less chance of runtime error)
- Security manager
  - System to control permissions for high-level actions
- Runtime verifier checks untrusted bytecode
  - Avoids havoc from hand-constructed bytecode

Advantages of Java

- In Summary, writing good codes in Java is easier than in C or C++

- javadoc generates documentation automatically
  - Very useful
  - Example: HLCM
- Now, I write all my programs in java

Outline

- C, C++, and Java

  How Java is related to C and C++
- Advantages of Java

  Writing good codes in Java is easier than in C or C++
- Java libraries (packages)

  Where the power of Java come from
- Java and the internet

  What Java is good for
- Course content

Java Libraries

- Java has far expanded traditional scope of a language's libraries
  - Java 2 SDK 1.6:
    - More than 100 packages
    - More than 3000 classes
- Much less effort required to accomplish common tasks
- Platform-specific details are handled
- All programs improved when common core updated

Java Libraries

- Collection of classes grouped into packages
  - Java equivalent of C libraries
- java.lang
  - String, Math, Exception, Thread, Runtime, etc
- java.util
  - Vector, Stack, hashtable, Date, Tokenizer
- java.io
  - Varieties of input/output processing
- java.net
  - Networking, client/server sockets, URLs
- java.awt, javax.swing
  - Windows, buttons, drawing, images, events

Java Libraries

❖ Java.Security

Encryption ,digital signature,message digest

❖ Java.text

Formating and parising

❖ Java.sql

Database connectivity

❖ Java.rmi

REmote method invocation ,distributed objects

❖ The list goes on...

And it is continuing to expand.Java is still Young

Outline

- C, C++, and Java

  How Java is related to C and C++
- Advantages of Java
  - Writing good codes in Java is easier than in C or C++
- Java libraries (packages)

Where the power of Java come from

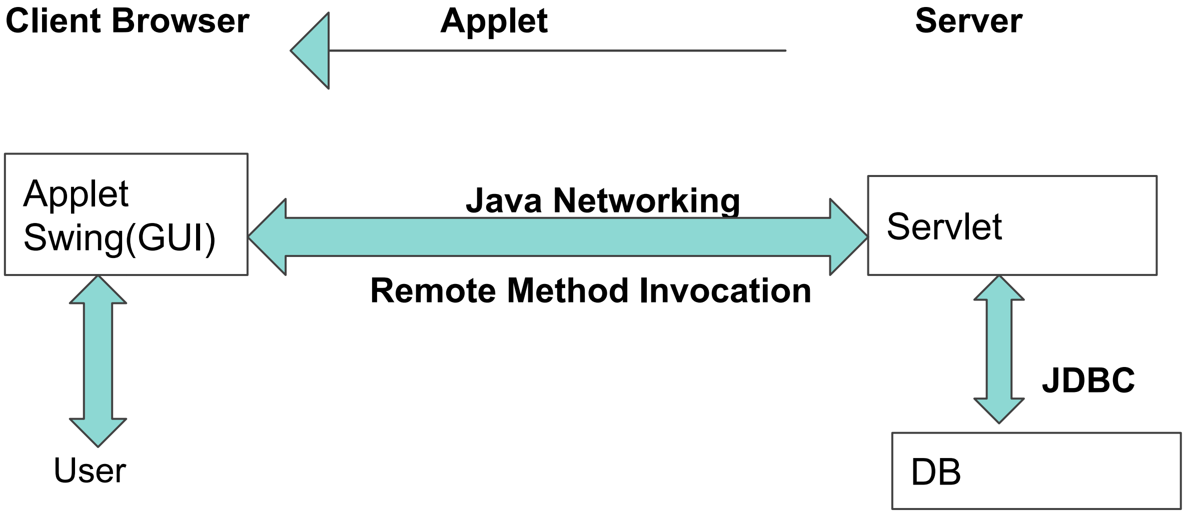
- Java and the internet
  - What Java is good for

- Course content

Java and the Internet

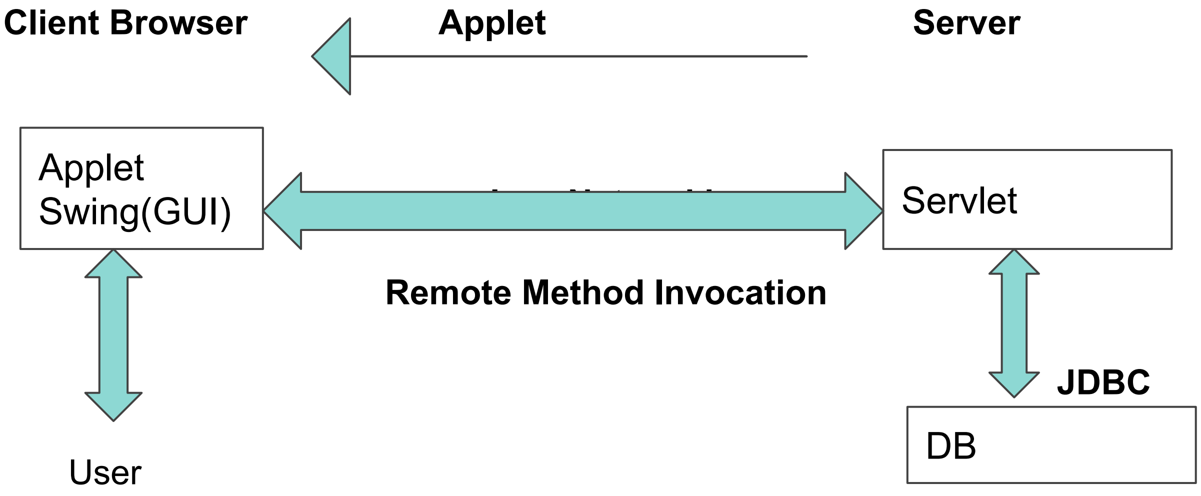
- Java is intended to be used in networked/distributed environments.
- Increasingly used for “middleware” to communicate between clients and serves, acting as a universal glue that connect user with information from various sources. Example: JMOL

Made possible by portability and multithreading and networking capabilities



Course Contents

- Language Basics:
    - Classes, objects, inheritance, interface, inner classes, exception, I/O
- Components in the following diagram
- Other issues
    - Multithreading, security



Client Browser

Applet

Server

Applet
Swing(GUI)

Servlet

Remote Method Invocation

JDBC

DB

User