CSC340: Comparing C++ with Java

Main Topics:
1. **Why do you need to learn C++?**
2. **What are the main differences between Java and C++?**

Readings:
1. **Appendix K**

Hui Yang
Computer Science Department
San Francisco State University
http://www.cs.sfsu.edu/~huiyang/

A brief history of C++ and Java

- **C was developed by Dennis Ritchie at AT&T Bell Labs in the 1970s.**
  - To implement the Unix operating system
  - Efficiency
- **C++ was developed by Bjarne Stroustrup at AT&T Bell Labs in the 1980s.**
  - Incorporate object oriented programming
  - Stay compatible with C as much as possible (**C remains a subset of C++)**
  - Efficiency: run as fast as C.
- **Java was developed by Sun Microsystem in the early 1990s.**
  - Marked as a modern alternative to C++
  - Syntactically similar to C++ ("C++ lite")
  - Pure object-oriented language
  - Designed to be hard for programmers to write incorrect programs.

C++ vs. Java: Similarities

- **Syntactically and semantically similar**
  - One can mechanically translate Java code to C++ code without great difficulty.
  - Most Java constructs have at least one corresponding C++ construct.
  - "Java is C++ without the guns, knives, and clubs."
    --James Gosling, the creator of Java
- **Functionally equivalent**
  - Generally speaking, if one can solve a problem in Java, one can also do it in C++. The opposite is true as well.

Why the 35-year-old C++ still dominates 'real' dev?

- **A recent interview with Bjarne Stroustrup by InfoWorld:**
  - http://www.infoworld.com/t/application-development/stroustrup-why-the-35-year-old-c-still-dominates-real-dev-248457 (August 15, 2014)
- **Synopsis**
  - [C++] age remains vital and relevant ... because **of its ability to handle complexity (**"nothing that can handle complexity runs as fast as C++"), making it the **go-to solution** for telecom, financial, and embedded applications and online systems such as Amazon and Google.
  - he used C++ for projects that "required a real programming language and **real performance**" [as C++ is for] high performance, high reliability, small footprint, low energy consumption, all of these good things.
  - it's always been used together with some scripting language or other.
  - C++ is **NOT FOR** small apps or hobbyists

Why learning C++?

- **C++ is still widely used in both industry and academia**
  - Many job openings require the candidates to master C++.
- **C++ is in general much more efficient in both space and time.**
  - Most high-performance computing systems are written in C/C++.
- **C/C++ is supported by many parallel programming standards**
  - OpenMP (shared memory systems)
  - POSIX threads (p-threads) (shared memory systems)
  - MPI (distributed memory systems)
- **Pointers**
  - Allow programmers to optimize the performance of a program
- **Templates: a missing feature in Java**
  - Allow one to write generic, type-independent code.
- **Operator overloading: another missing feature in Java**
  - Allow one to define operators (e.g., + and =) for user-specified data types (or classes)

C++ vs. Java: Main Differences (1)

- **Compiled vs. interpreted code**
  - C++ code is compiled into native code.
  - Poor cross-platform compatibility
  - Runtime efficiency
- **Java code is compiled into Java bytecode that is interpreted by the Java Virtual Machine at runtime.**
  - Good compatibility
  - Less than ideal runtime efficiency
- **Java uses packages to group related topics, whereas C++ uses libraries**

C++ vs. Java: Main Differences (2)

	C++	Java
API:	small	huge
Compiler checks:	some	numerous
Garbage collection:	no	yes
Operator overloading:	yes	no
Check array indices:	no	yes
Standard GUI library:	no	yes
Exception handling:	some	yes
Pointer variables:	yes	no
Mutli-threading	platform specific	yes

C++ vs. Java: Main Differences (3)

- **Boolean variables**
  - Java
    - Data type: boolean
    - Values: true and false
  - C++:
    - Data type: bool
    - Values: true and false; BUT they are actually integers (one and zero)
    - In a predicate or test clause, any non-zero will be considered as true
- **String in Java vs. string in C++**
  - #include <string> //C++
- **Constants**
  - Java: declared within a class or method using the syntax
    - public static final int MAX_SIZE = 20;
  - C++: declared outside a class using the keyword const
    - const MAX_SIZE=20;

C++ vs. Java: Main Differences (3)

- **Arrays**
  - Java: arrays are objects; have a public field *length*; an exception is thrown if attempting to access a non-existing position
    - int[] myArray = new int [MAX_SIZE];
  - C++: arrays are sequences of numbers of objects.
    - int myArray [MAX_SIZE]; //fixed-size array
    - int myArray[] = new int [MAX_SIZE]; //dynamic array
    - delete [] myArray; //programmers need to deallocate the space
  - The C++ equivalence of Java arrays is vector.
    - #include <vector>
- **Any numeric data type can be declared as ''unsigned''**
  - int [-231 , +231]
  - unsigned int [0, 232]

C++ vs. Java: Main Differences (4)

- **Test expressions**
  - Two if-else statements

```
if (x == 7)
    y += 10;
else
    y += 20;
```

```
if (x = 7)
    y += 10;
else
    y += 20;
```

- **Functions**
  - C++ functions don't need to be inside a class.
  - Function prototype
    - double is_even(int x);
  - Types
    - Stand-alone functions
    - Member functions
    - Non-member functions in a class

C++ vs. Java: Main Differences (5)

- **Difference between syntax when defining a class**
  - E.g., need a semicolon at the end of a class declaration
  - Separate compilation: header file and implementation file (will be discussed in a later lecture)
- **Input and output**
  - Java: use System.in and System.out

```
Scanner keyboard = new Scanner(System.in);
System.out.print("How many apples are in a box? ");
int applesPerBox = keyboard.nextInt();
System.out.print("You entered " + applesPerBox + " apples per box. ");
```

  - C++: Does not have System.in and System.out
    - Use cin and cout

```
#include <iostream>
Using namespace std;
cin>> number; cout << number;
```

    - Many different ways

```
Scanf("%d %s", &number, string);  printf("%d %s\n", number, string);
```


C++ vs. Java: Main Differences (6)

- **Compilation**
  - Java compiler will automatically search a class in an imported package.
  - C++
    - all the classes, functions and variables must be declared before they are used.
    - Use a pre-processor mechanism so that classes can be specified in a separate header file, e.g., "myClass.h"
      - #include "myClass.h" //pre-processor directives

Java: Unique Features

- **The "synchronized" keyword.**
- **JavaDoc**
  - doxygen.org for C++
- **The "instanceof" operator**
- **C++ does not have interfaces*.**
- **C++ does not come with a large standardized class library as Java does.**

C++: Unique Features

- **Constant and Macros**
  - #define min(a,b) (((a) < (b)) ? (a) : (b))
- **Compilation directives**

  #ifdef HOST_SPARC
  #include "myheader.h"
  #endif
- **Templates**
- **Life outside of a class**
  - It is common to declare and define functions and variables outside a class
- **Pointers and memory management**
  - The "new" operator allows the programmer to request memory.
  - The "delete" operator allows the programmer to return memory.

Summary

- C++ and Java are syntactically similar in many ways.
- They are designed with different goals.
- Major syntactic differences between C++ and Java
  - Boolean
  - Constants
  - Class and functions
  - Arrays
  - Input and output
  - Compilation
- It is essential and practical to master both languages.

Quiz: True or False

1. In C++, you can declare a boolean variable as follows:

   ```
   boolean is_Correct;
   ```

2. In general, Java is designed to be more programmer friendly, whereas C++ is designed to solve a problem more efficiently.
3. Similar to Java, C++ uses the concept of "Package" to bundle related topics.
4. C++ always checks the boundary of an array.