Java vs C++

About the languages

- Java
  - Is an abstract, general-purpose, high-level programming language that is designed to minimize implementation dependencies.
  - mainly used for design web based application but also use for develop desktop application.
- C++
  - Low level, well optimizable language to implementation, which is extending the procedural programming C language (C was primary designed for efficient execution)
  - used for design only desktop application like OS, Compiler, embedded systems etc.

Base language characteristics, features, differences

	C++	**Java**
Main paradigms	Mix: Object oriented, Functional, Procedural, Generic, Template metaprogramming	Mainly Object Oriented, but can: Functional, Procedural, Generic
Standard library	• C++ is an extension for C • Standard lib has a limited scope* • But has multiple 3rdparty libs (eg:Boost)	Extensive library grown with releases
Run && Performance	Compiled, then natively runs (platform dependent) WOCA - Write Once Compile Anywhere	Compiles, then use interpereter: Java Virtual Machine (platform independent) WORA -Write Once Run Anywhere||Everywhere
Resource management	Supervised, explicit memory management, (constructor && destructors, smartpointers, scope end) lifetime-based resource management (RAII - Resource acquisition is initialization)	Garbage collector (non-deterministic) Trywith resources Can be tuned with JRockit, but still not the best
Miscellaneous	Header file include to share declarations between sourcefiles	Package system + compiler imports the classes

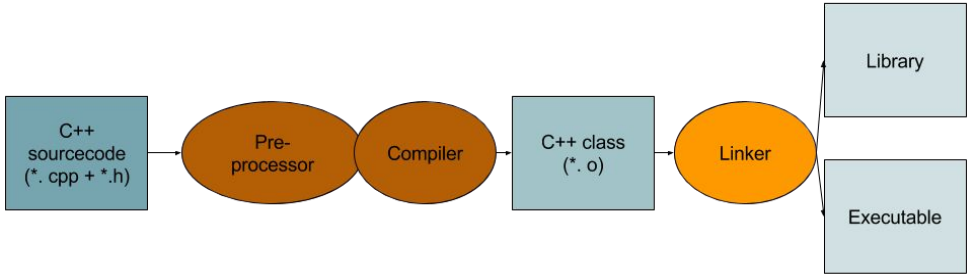
Data-storing structure differences

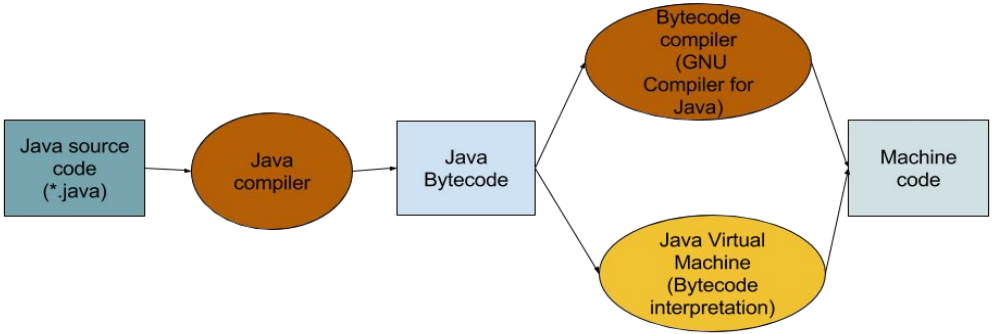
	C++	**Java**
Type && binding	Statically typed*	Statically typed*, but type-erase at runtime
Pointer arithmetic*	supports	sun.misc.Unsafe
Language tools for datasStructures	• Class • Primitive • Enum • Pointer • Structure • Union* concept + Tuples*	• Class • Primitive • Enum

Language tool differences

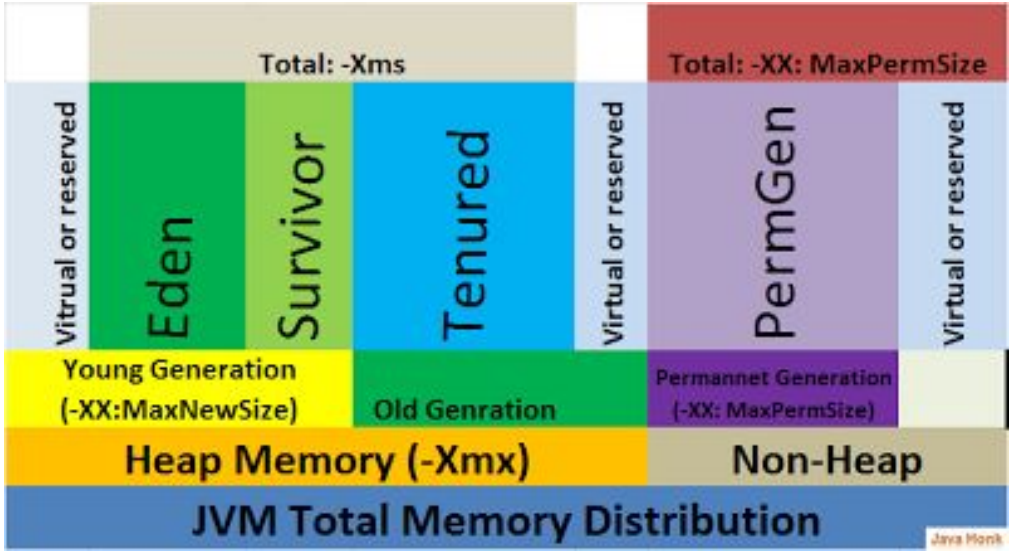
	C++	Java
Exception-handling	supports	supports
Inheritance	Multiple	• Interface* • Just single inheritance (since Java 8 : Default methods in interfaces)
“Generics”	• Generics (type is given) • Templates (type is just a “parameter”)	• Generics (type is given) • Wildcards (type is just a “parameter”)
Reflection	RTTI (but just compile time)	Runtime Reflection
Delegates*	has	Inner classes or interfaces can use the way like delegating, AND lambda (since java 8)
Overloading	Method (if marked virtual) + operator	Method(virtual by default, but can marked final)*

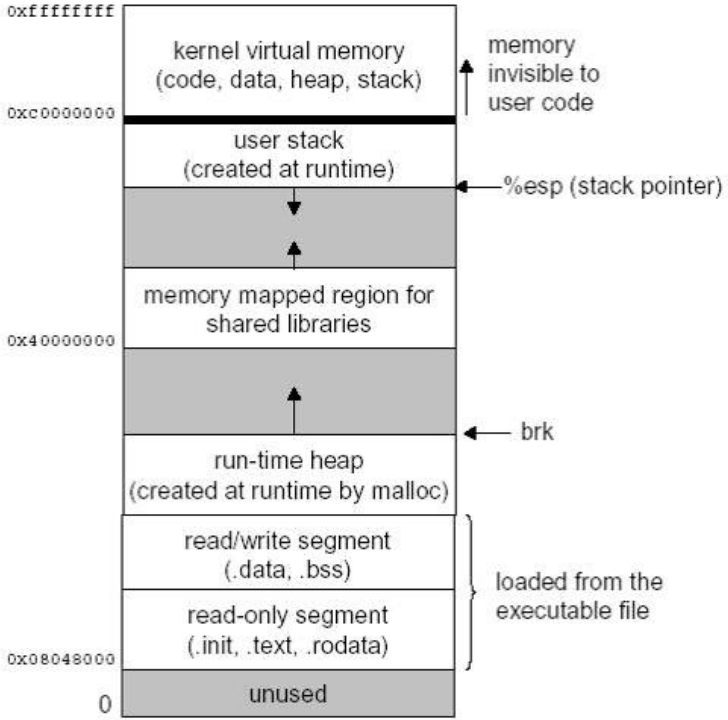
Run the code


C++ sourcecode (*. cpp + *.h)
Pre-processor
Compiler
C++ class (*. o)
Linker
Library
Executable


Java source code (*.java)
Java compiler
Java Bytecode
Bytecode compiler (GNU Compiler for Java)
Java Virtual Machine (Bytecode interpretation)
Machine code

Memory management





Pointers

Pointers in C:

```
void main()
{
 int* x; // Allocate the pointers x and y
 int* y; // (but not the pointees)
 x = malloc(sizeof(int)); // Allocate an int pointee, // and set x to point to it
 *x = 42; // Dereference x to store 42 in its pointee
 *y = 13; // CRASH -- y does not have a pointee yet
 y = x; // Pointer assignment sets y to point to x's pointee
 *y = 13; // Dereference y to store 13 in its (shared) pointee
 }
```

Pointers in Java:

```
class IntObj
{
 public int value;
}
public class Binky()
{
 public static void main(String[] args) {
       IntObj x; // Allocate the pointers x and y
       IntObj y; // (but not the IntObj pointees)
       x = new IntObj(); // Allocate an IntObj pointe // and set x to point to it
       x.value = 42; // Dereference x to store 42 in its pointee
       y.value = 13; // CRASH -- y does not have a pointee yet
       y = x; // Pointer assignment sets y to point to x's pointee
       y.value = 13; // Deference y to store 13 in its (shared) pointee
 }
}
```

References && further interesting articles, topics

- Escape analisis java: http://docs.oracle.com/javase/7/docs/technotes/guides/vm/performance-enhancements-7.html
- Tuples in java: https://blog.jooq.org/2016/02/16/an-ingenious-workaround-to-emulate-sum-types-in-java/
- Javabytecode to native: https://gcc.gnu.org/
- Java security manager: http://mishadoff.com/blog/java-magic-part-5-securitymanager/