C and Java
Similarities, Differences

Similarities btwn C, Java

- Both are compiled, procedural languages , with main() being the entry point to the program

- Their procedures (C functions, Java methods) look & act essentially the same. There are minor differences, e.g. what 'static' means.

- Same primitive data types: byte, char, short, int, long, float, double

  NOTE: C doesn't have boolean. Java doesn't have unsigned.

- Sizes of data types are standardized in Java . E.g. all Java ints have 4 bytes. In C, the size of an int is typically the processor's word size (e.g. 4 bytes on a 32 bit machine, 8 bytes on a 64 bit one)

- Same operators: *, /, %, +, -, =, ==, <, <=, >, >=, !=, !, ++, --, (cast), &&, ||, [], (), ?:, &, |, ~, shortcut assignment operators such as +=

  NOTE: C doesn't have objects, so it doesn't have the instanceof operator. Java doesn't have pointers (which directly access memory) so it doesn't have the pointer operators &,*, ->. Both have a dot operator but they mean different things.

Similarities, cont.

- Same flow control statements: if, while, for, do-while, return, break, continue, switch

```
// Minor difference: a variable may not be declared in a
// for-loop header's first expression.
// This is not allowed:
   for (int = i; i < 10; i++) { ... }

// Instead do this:
   int i = 0;
   for (i = 0; i < 10; i++){ ... }

// Or do this:
   int i = 0;
   for (; i < 10; i++){ ... }
```

- Arrays are accessed the same (data[i]) (but the syntax for creating them is different).

- Same code blocks, delimited by { and }, within which local variables may be declared. White space is ignored.

- Same comments

```
// this is a single line comment
/* this is a
 * multi-line comment
 */
```

Java Program vs. C Program

```java
// A Java program


public class X {
  public static void main(String[] args){
     System.out.print("hello world");
     foo(5);
  }


  static void foo(int num){
     num += 2;
     System.out.println("num is " + num);
   }

}
```

```c
// A C program that does the same thing
#include <stdio.h>
void doit(int);

int main() {
   printf("hello, world");
   foo(5);
   return 0;
}

void foo(int num) {
   num += 2;
   printf("num is %d\n", num);
}
```

```c
// Alternative main() headers:
int main(int argc, char *argv[]){
int main(int argc, char **argv ){
```

Java and C Differences

Java is type-safe and memory-safe

C is not!!!

Important Differences

- Java is more robust; it's type-safe and memory-safe.
  C is faster, but more prone to errors (in particular, memory errors).

- C has no objects! Hence, it has no classes, fields (instance variables), constructors, or public/private/protected keywords

- Legend has it that if you can get a Java program to compile, in many cases it will run almost flawlessly. Not so with C - the compiler is (too) lax, so more problems show up at runtime.

- C is not "type safe". A C compiler may allow things such as
  - if (x = 1)
  - incorrect return type

- C has no boolean type; instead integers are used.
  0 is considered to be false, non-zero to be true
  A common mistake that may compile:

```
if ( x = 0 )  // always false!
   statement;
```

Important Differences cont

- C is not memory safe. THESE FACTORS ARE MAJOR:
  - C allows array out of bounds access
  - C does not have garbage collection to reclaim unused memory
  - C allows writing to an illegal address (with inappropriate type)
- C compiles to (processor-dependent) machine code. Java compiles to portable bytecode; when the program is run the JVM translates it in to machine code.
- Java's language specification is relatively precise. In most cases, a Java program behaves the same no matter what machine it's run on. C is ambiguous regarding several issues such as sizes of data types and behavior under special circumstances. Standards (ANSI, ISO) have helped. Still, be aware that there are portability issues.

Important Differences cont.

- Java is inherently more organized/modular. All the code is in classes. It (like all OOP languages) has "encapsulation": the data and the procedures that act on it are bundled together in an object. The data can be protected from corruption by "hiding it" (making it private) and only letting the object's methods read/write it. C does not have this level of organization.

- A C program may have one or more source files, each or which may have multiple functions. One and only file has a main() method. C programs may have global variables (declared outside a function), but they should be avoided.

- A C programmer has to be more diligent about keeping their code organized. An example of a disorganized C program file is one with a lot of global variables that contains a lot of functions that are not logically related. This makes the code difficult to read, debug, and maintain.

Other Differences

- C has a "preprocessor"
- Java has import statements; C has #include statements
- C does not allow function overloading (functions with same name but different parameter list at same area of scope)
- In Java, an array is an object, so you create it with the new operator. C doesn't have objects.
- In C, a struct can be used to group related data (e.g. name, addr, phone number). However, unlike an object, a struct doesn't have procedures inherently associated with it