C Programming, Chapter 1: C vs. Java

T. Karvi

March 2011
Although the syntax of Java and C are very similar, they are very different languages. The following table shows some of the major differences:

<table>
<thead>
<tr>
<th>C</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedural</td>
<td>Object-oriented</td>
</tr>
<tr>
<td>Compiled</td>
<td>Interpreted</td>
</tr>
<tr>
<td>No Memory Management</td>
<td>Memory Management</td>
</tr>
<tr>
<td>Pointers</td>
<td>References</td>
</tr>
<tr>
<td>Error Codes</td>
<td>Exceptions</td>
</tr>
</tbody>
</table>

Object-Oriented vs. Procedural:

One of the largest differences between Java and C is the use of a different programming paradigm. Java is an Object-Oriented language. A Java program consists of a collection of objects. These objects contain the data used in the program, and have methods to perform operations on this data.
The C language is procedural. A C program consists of a collection of procedures (or functions). The data used by the program can be put into local variables (inside of a function) or global variables (outside of functions). There is no notion of objects in C. Just like in Java, there is a special main function.

Interpreted vs. Compiled: Java is an interpreted language. Java source code is transformed to bytecode, which is then loaded by a program called an interpreter. This program then ‘executes’ each of the bytecode instructions one by one, translating them into something the machine understands.

C programs are compiled. Instead being translated to some intermediate format (like bytecode) it is translated directly into machine code. This machine code is directly executed by the processor, which is used to start the program.
Java is an interpreted language. Java source code is transformed to bytecode, which is then loaded by a program called an interpreter. This program then ‘executes’ each of the bytecode instructions one by one, translating them into something the machine understands.

C programs are compiled. Instead being translated to some intermediate format (like bytecode) it is translated directly into machinecode. This machinecode is directly executed by the processor.

Memory Management vs. NoMemoryManagement:

- In Java, the memory management is done automatically by the system. New objects can be created using the new keyword. When objects are no longer used (i.e., no longer have any references pointing to them) they are removed by the garbage collector.

- In C, the programmer has to do his own memory management. Using the keyword sizeof and the library calls malloc and free, blocks of memory can be allocated and freed.
References vs. Pointers:

- A reference in Java is a special variable which references (points-to) an object. Only objects can be referenced. For example, it is not possible to have a reference to an int.

- Pointers in C are in some ways similar to references in Java (they point to things), but in many ways they are very different.

Exceptions vs. Error Codes:

- Whenever an error occurs in Java, an exception is thrown.

- C has no exceptions. A function either returns some error code (when an error is expected), or your program crashes (usually long after the error has occurred).