# Terms and Definitions

| Term | Definition |
| --- | --- |
| Atomic | An Atomic operation is an operation that completes which cannot be interrupted. For example, setting a one word integer is an atomic operation while setting the characters in a string can be interrupted because multiple memory operations are needed. |
| Call Stack | A set of data pushed onto the Stack for communications between the calling code and the method called. This data includes a place to store the return value and the parameters passed to the method. Other information may be stored in the call stack based on the operating system and the compiling and linking options used. |
| Compiler | A program that reads source code of another program and generates machine code that can run on the computer. In some cases, as with Java, the program is translated in p-code which runs on a virtual machine like the Java JVM. |
| Declaration | To define a data type such as a class and its properties such as data types and methods. Note, no memory or storage is used by declaring a data type. This information is used by the compiler not the program. [See Definition] |
| Definition | To create a variable or object. This is when memory is assigned to the object. The object is used by the program. [See Declaration] |
| Heap | Dynamic memory used when creating dynamic variables, using **new** in Java. In C++ the Heap is used when one of the allocate methods are called or when using **new** to create an object. When a dynamically created (**new**) object is no longer referenced in Java or deleted in C++. The memory used it marked as reusable and may be assigned to the next dynamically created variable. |
| Interpreter | A program that reads source code of a scripted language program and executes the script. This involves translating source code into machine code that can actually run on the computer. |
| Interrupt | An event that pauses the current code that is executing so that other code can be executed. There are two major sources of interrupts. The first is an external event occurs that causes the program to pause so that an Interrupt Handler can run. The other is if the program is running on multiple threads, one thread can be paused so that another thread can run. |
| Interrupt Handler | Special code that is used to process external events. This code will interrupt the currently running code. Examples of external events: a button is pushed, a range sensor detects something inside a specified distance. |
| Linker | A program that takes the binary or object file produced by the compiler and assigns addresses to libraries and methods within the program. The output of the Linker is an executable program. |
| Lock | There are several locking mechanisms used in multi-threaded programs. Basically a lock allows a section of code to run to completion in one thread without another thread being able execute the same code or change the same objects. Locks are used to make code thread safe. |
| Multi-Threaded | A program that has more than one thread of execution. Programs that run on multiple threads have to be careful when an object is used on multiple threads at the same time. |
| Stack | Dynamic memory used when a function is called. When a function is called, the arguments passed to it are pushed onto the Stack and any variables or objects are also pushed onto the Stack. When the function returns, all the variables and arguments are popped off the Stack so that memory is available for reuse. |
| Thread Safe | Code is said to be thread safe if it can be run on multiple threads at the same time without changes to data in one thread being lost to another thread. |
| UML | Unified Modeling Language – A graphical representation of programming, especially suited of object oriented programming. |