**Before we discuss multithreading, first let's understand the following terms**  
**1. Process**- Process is what the operating system uses to facilitate the execution of a program by providing the resources required. Each process has a unique process Id associated with it. You can view the process within which a program is being executed using windows task manager.  
**2. Thread**- Thread is a light weight process. A process has at least one thread which is commonly called as main thread which actually executes the application code. A single process can have multiple threads.  
  
**Please Note:**All the threading related classes are present in System.Threading namespace.

**Advantages of multithreading:**  
**1.** To maintain a responsive user interface  
**2.** To make efficient use of processor time while waiting for I/O operations to complete.  
**3.** To split large, CPU-bound tasks to be processed simultaneously on a machine that has multiple CPUs/cores.

**Disadvantages of multithreading:**  
**1.** On a single-core/processor machine threading can affect performance negatively as there is overhead involved with context-switching.  
**2.** Have to write more lines of code to accomplish the same task.  
**3.** Multithreaded applications are difficult to write, understand, debug and maintain.  
  
**Please Note:** Only use multithreading when the advantages of doing so outweigh the disadvantages.