Difference between Setter and Constructor Injection in Spring framework  
  
1) Fundamental difference between setter and constructor injection, as there name implies is How dependency is injected.  Setter injection in Spring uses setter methods like setDependency() to inject dependency on any bean managed by Spring's IOC container. On the other hand constructor injection uses constructor to inject dependency on any Spring managed bean.

2) Because of using setter method, setter Injection in **more readable** than constructor injection in Spring configuration file usually applicationContext.xml.

Since setter method has name e.g. setReporotService() by reading Spring XML config file you know which dependency you are setting. While in constructor injection, since it uses index to inject dependency, it’s not as readable as setter injection and you need to refer either Java documentation or code to find which index corresponds to which property.

3) Another difference between setter vs constructor injection in Spring and one of the drawback of setter injection is that **it does not ensures**[**dependency Injection**](http://javarevisited.blogspot.sg/2012/03/10-object-oriented-design-principles.html)**. *You cannot guarantee that certain dependency is injected or not***, which means you may have an object with incomplete dependency. On other hand constructor Injection does not allow you to construct object, until your dependencies are ready.

4) One more drawback of setter Injection is Security. By using setter injection, you **can**[**override**](http://javarevisited.blogspot.in/2011/12/method-overloading-vs-method-overriding.html)**certain dependency** which is not possible with constructor injection because every time you call constructor, a new object is gets created.  
  
5) If Object A and B are dependent each other i.e A is depends on B and vice-versa. Spring throws ObjectCurrentlyInCreationException while creating objects of A and B bcz A object cannot be created until B is created and vice-versa. So spring can resolve circular dependencies through setter-injection. Objects constructed before setter methods invoked.  
  
**When to use Setter Injection over Constructor Injection in Spring**

Setter Injection has upper hand over Constructor Injection in terms of readability. Since for configuring Spring we use [XML files](http://javarevisited.blogspot.in/2011/12/parse-xml-file-in-java-example-tutorial.html), readability is much bigger concern.

Also drawback of setter Injection around ensuring mandatory dependency injected or not can be handled by configuring Spring to check dependency using "dependency-check" attribute of tag. Another worth noting point to remember while comparing Setter Injection vs Constructor Injection is that, once number of dependency crossed a threshold e.g. 5 or 6 its handy manageable to passing dependency via constructor. Setter Injection is preferred choice when number of dependency to be injected is lot more than normal, if some of those arguments is optional than using [Builder design pattern](http://javarevisited.blogspot.in/2012/06/builder-design-pattern-in-java-example.html) is also a good option.