What are extension methods?

Extension methods enable you to add methods to existing types without creating a new derived type, recompiling, or otherwise modifying the original type.   
  
An extension method is a special kind of static method, but they are called as if they were instance methods on the extended type.

How to use extension methods?

An extension method is a static method of a static class, where the "this" modifier is applied to the first parameter. The type of the first parameter will be the type that is extended.  
  
Extension methods are only in scope when you explicitly import the namespace into your source code with a using directive.

The ref keyword **is** used to pass values by **reference**. ... **Output parameters** specified with the out keyword are **for** returning values from a method. One key **difference in the** code **is** that you must set the value **of** an **output parameter** within the method. This **is** not the case **for** ref **parameters**

**When the yield return statement is reached** the value is **returned**, and the current **location** in code is retained. Execution is restarted from that **location** the next time that the iterator function is called. The end result is that you get the numbers 1 to 5 printed in the console.

**Yield**\* with **Return**. **Yield**\* with a **return** behaves a bit differently than the normal**yield**\*. When **yield**\* is used with a **return** statement it evaluates **to** that value,**meaning** the entire **yield**\* function() becomes equal **to** the value **returned** from the associated generator function.

A variable declared inside the **anonymous method can**'**t** be **accessed outside** the**anonymous method**. An **anonymous method**, declared without parenthesis **can**be assigned **to** a **delegate** with any signature. An **anonymous method can**'**t access** the **ref or out parameters** of an outer scope.

3 Answers. In general you add all prerequisite steps to **setUp** and all clean-**up** steps to **tearDown**. When a **setUp**() **method** is defined, the test runner will run that**method** prior to each test. Likewise, if a **tearDown**() **method** is defined, the test runner will invoke that **method** after each test.

**Referential integrity** (RI) is a relational database concept, which states that table relationships must always be consistent.

**MSDB** is an important system database in Microsoft SQL server. The **msdb** database is used mainly by the SQL Server Agent to store system activities like sql server jobs, mail, service broker, maintenance plans, user and system database backup history, etc..It is also used by database engine and management studio.

**Data Definition Language** (**DDL**) **DDL** is used for specifying the database schema. It is used for creating tables, schema, indexes, constraints etc. in database.

Temporary stored procedure is a little known feature of SQL Server.  This article is an attempt to throw some light on this interesting feature.

Temporary stored procedures are like normal stored procedures but, as their name suggests, have fleeting existence. There are two kinds of temporary stored procedures local and global. Temporary stored procedures are created just like any other SPs; however the name must begin with a hash (#) for a local temporary SP and two hashes (##) for a global temporary stored procedure.

A local temporary stored procedure is available only in the current session and is dropped when the session is closed. A global temporary stored procedure is visible to all sessions and is dropped when the session of the user that created it is closed. If there are any executing versions of the global stored procedure when the creator session is closed, those are allowed to complete, but once they are done and the creator session is closed, no further execution of the global temporary stored procedure is allowed.

Aside from local and global temporary stored procedures, you can also create regular stored procs in tempdb. These only exist until sql server is restarted because tempdb is recreated each time server is restarted.

Global **Temporary Stored Procedures** are available to all SQL Server connections as long as the process that **created** it exists. When last connection using the**procedure is** closed, then the **procedure is**dropped. **Temporary Stored Procedures** like**temporary** tables are **created in TempDB**database.

**Q.** Temporary Stored Procedure  
  
  (i) These Stored procedures are created in tempdb and  dropped when clients terminate  
       the connection to the SQL Server.  
  (ii) All the Temporary stored procedure are prefixed with either # or ##