# Assembly practice - Tools

Create a solution that contains 2 projects:

1. Tools class library project that contains Extensions static class the extends string class with HasSpace , HasLowerCase , HasUpperCase , HasNumber static methods that returns true|false after checking their string parameter.
2. ToolsTester console application project that uses Tools library to validate user input and display the result per input .
3. User interface example:

Text

Description automatically generated

## **Create and use dll assembly from another assembly using Visual studio code :**

1. Create a solution. "dotnet new sln"
2. Create a class library project "dotnet new classlib -o X"
3. Add code and build the class library in the project folder "dotnet build"
4. Add project to solution "dotnet sln add X/X.csproj" --
5. Create a console app "dotnet new console -o Y"
6. Add project to solution "dotnet sln add Y/Y.csproj"
7. Add reference in project Y to project X "dotnet add Y/Y.csproj reference X/X.csproj"
8. Add code that uses X logic and build the console app in the project folder "dotnet build" --
9. Build the solution in the solution folder "dotnet build"

<https://docs.microsoft.com/en-us/dotnet/core/tutorials/library-with-visual-studio-code?pivots=dotnet-6-0>