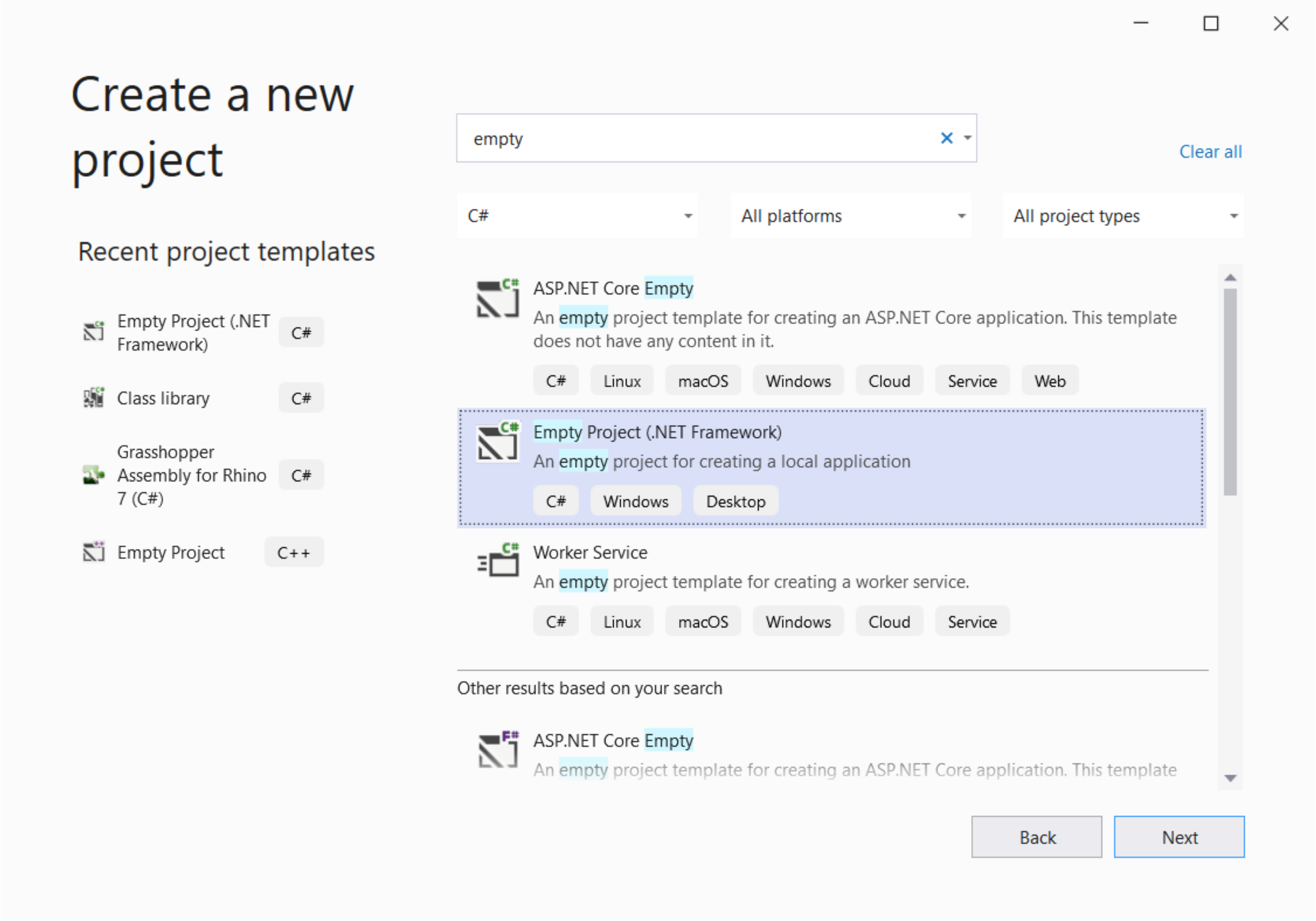
Steps to create Grasshopper Plugin

MoP Spring 2024, Abed Naran
based on slides by Haitham Fadila

Framewrok Requirements

- Install Visual Studio 2019.
- Install Rinho 8. You have three months of a free trial.

1. Create a Visual studio (19) C# empty project.



Create a new project

Recent project templates

- Empty Project (.NET Framework) C#
- Class library C#
- Grasshopper Assembly for Rhino 7 (C#) C#
- Empty Project C++

empty

C# | All platforms | All project types

ASP.NET Core Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
C# Linux macOS Windows Cloud Service Web

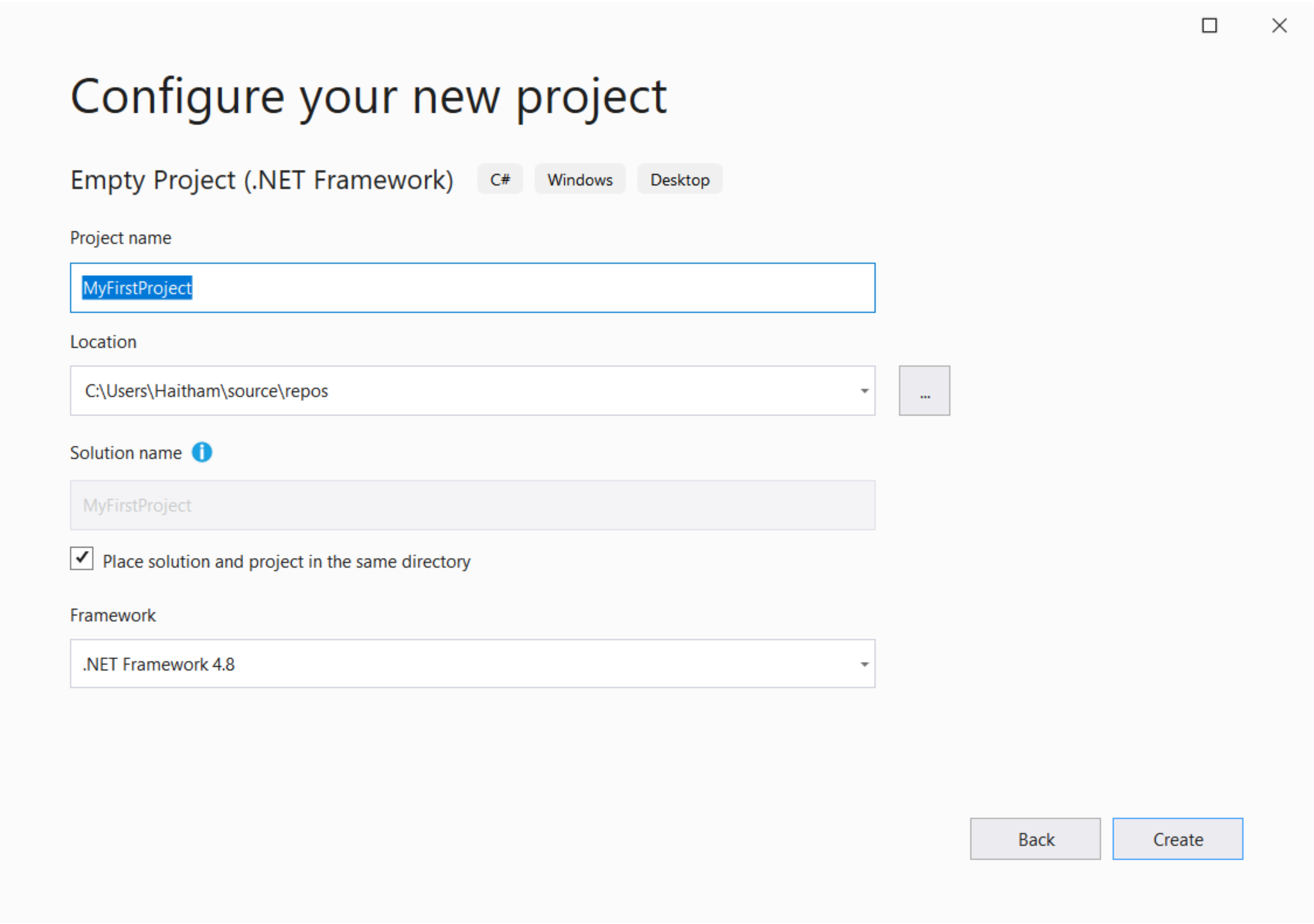
Empty Project (.NET Framework)
An empty project for creating a local application
C# Windows Desktop

Worker Service
An empty project template for creating a worker service.
C# Linux macOS Windows Cloud Service

Other results based on your search

ASP.NET Core Empty
An empty project template for creating an ASP.NET Core application. This template

2. Set the Framework to: “.NET Framework 4.8”



Configure your new project

Empty Project (.NET Framework)

Project name

MyFirstProject

Location

C:\Users\Haitham\source\repos

Solution name

MyFirstProject

☑ Place solution and project in the same directory

Framework

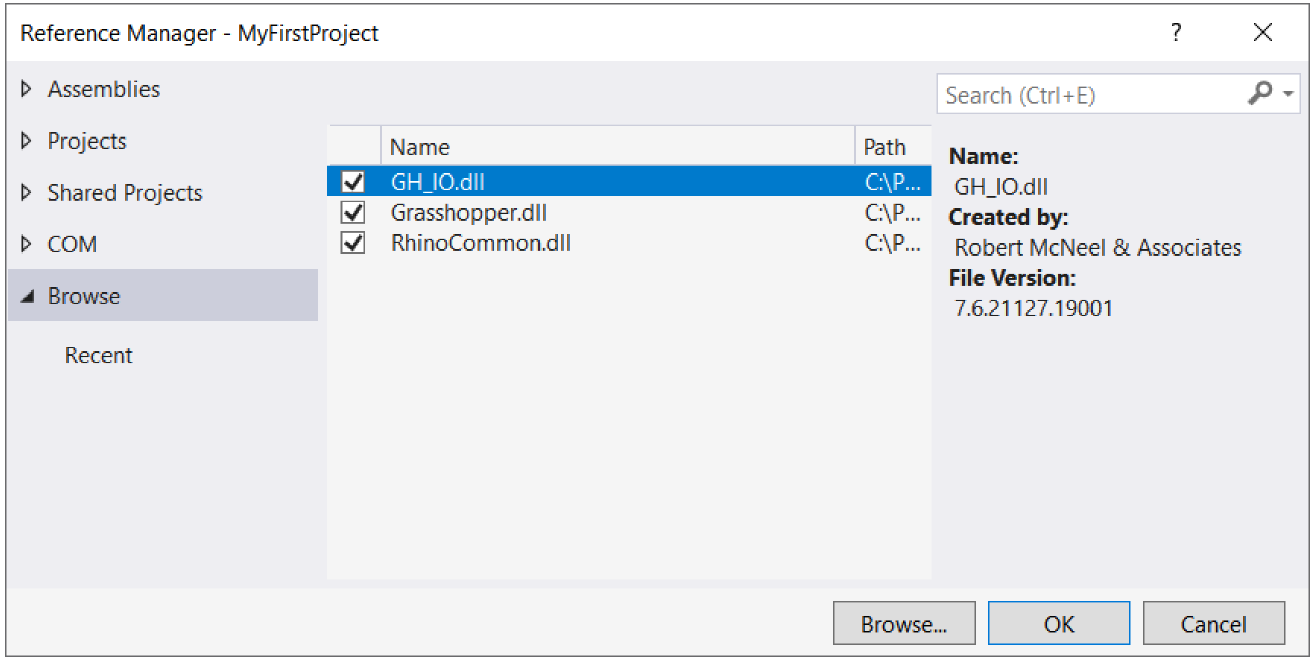
.NET Framework 4.8

3. Add to your solution all these Assembly References:

3.1) Navigate: Solution references -> add Reference -> Browse.

3.2) Add these references:

Assembly	Probable location
GH_IO.dll	*<Program Files>\Rhino 8\Plug-ins\Grasshopper\*
Grasshopper.dll	*<Program Files>\Rhino 8\Plug-ins\Grasshopper\*
RhinoCommon.dll	*<Program Files>\Rhino 8\System\*

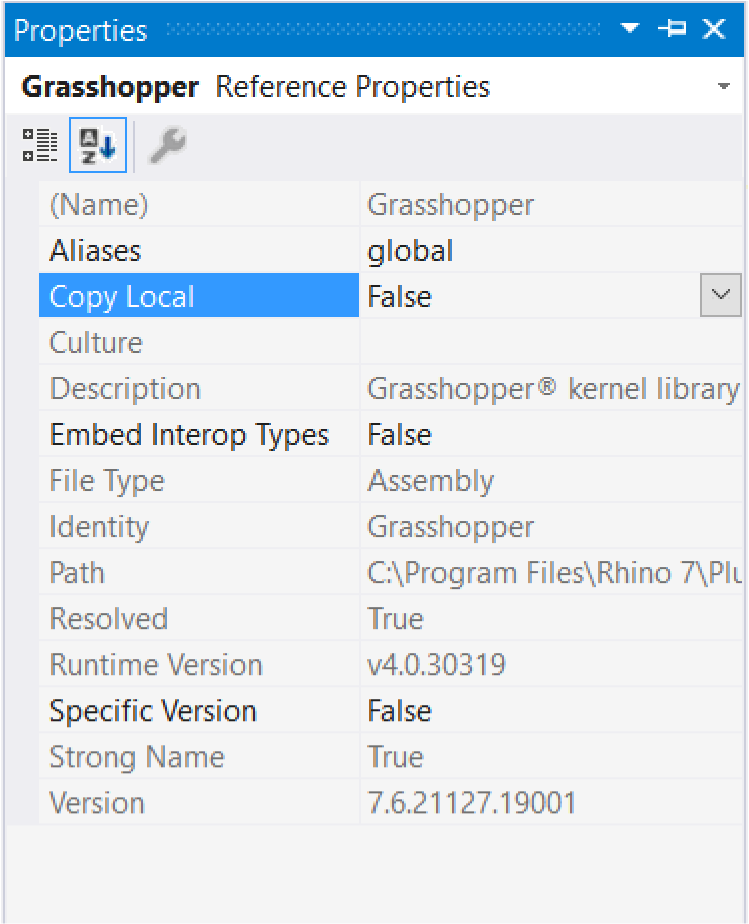


Reference Manager - MyFirstProject

	Name	Path
☑	GH_IO.dll	C:\P...
☑	Grasshopper.dll	C:\P...
☑	RhinoCommon.dll	C:\P...

Name:
GH_IO.dll
Created by:
Robert McNeel & Associates
File Version:
7.6.21127.19001

3.3) Set the **Copy Local** tag of GH_IO, Grasshopper, and RhinoCommon to **False** before you press the Compile button for the first time.

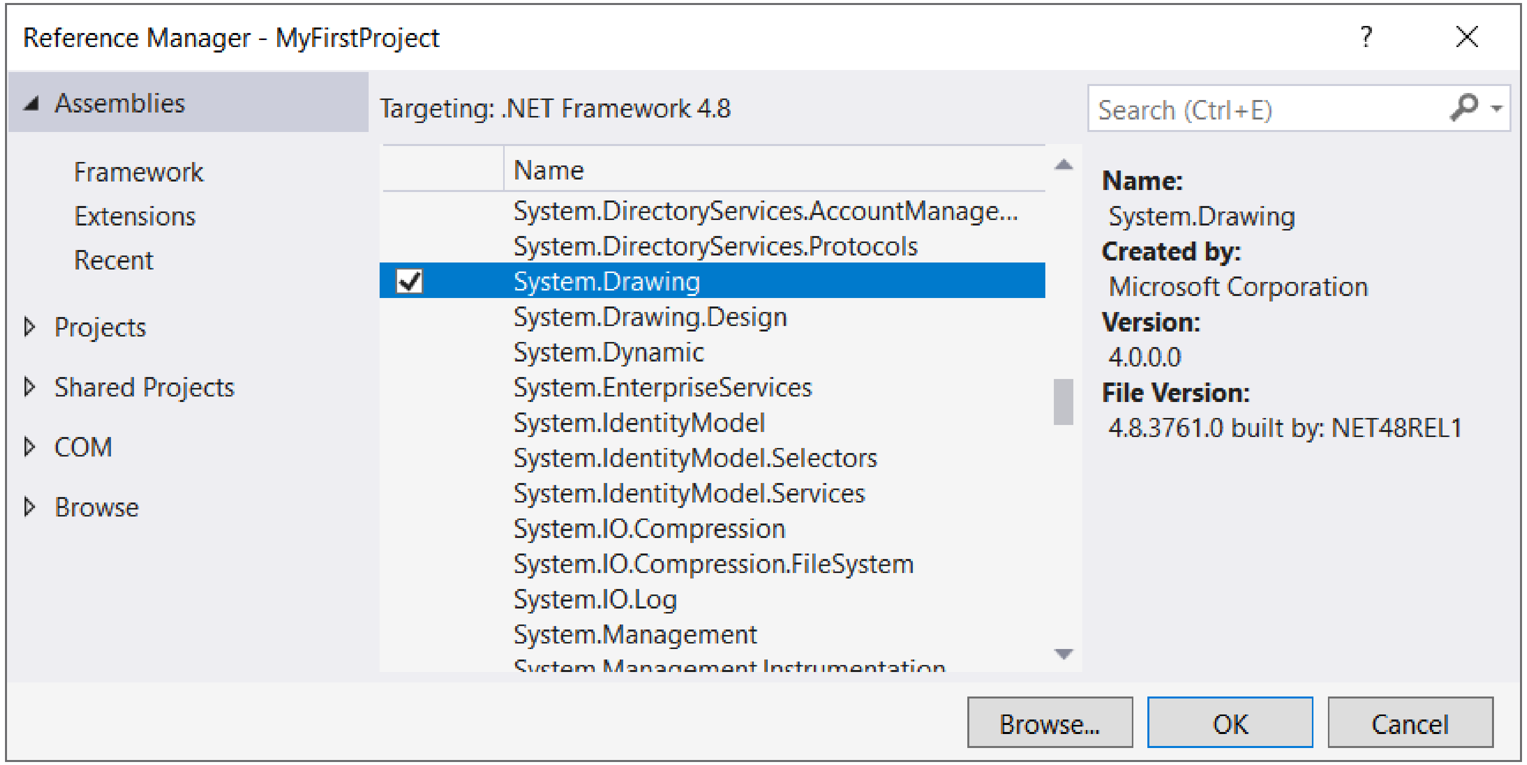


Grasshopper Reference Properties

(Name)	Grasshopper
Aliases	global
Copy Local	False
Culture	
Description	Grasshopper® kernel library
Embed Interop Types	False
File Type	Assembly
Identity	Grasshopper
Path	C:\Program Files\Rhino 7\Plu
Resolved	True
Runtime Version	v4.0.30319
Specific Version	False
Strong Name	True
Version	7.6.21127.19001

3.4) Navigate: Solution references -> add Reference -> Assembles.

3.5) Enable “System Drawing”



Targeting: .NET Framework 4.8

	Name
	System.DirectoryServices.AccountManage...
	System.DirectoryServices.Protocols
☑	System.Drawing
	System.Drawing.Design
	System.Dynamic
	System.EnterpriseServices
	System.IdentityModel
	System.IdentityModel.Selectors
	System.IdentityModel.Services
	System.IO.Compression
	System.IO.Compression.FileSystem
	System.IO.Log
	System.Management
	System.Management.Instrumentation

Name:
System.Drawing
Created by:
Microsoft Corporation
Version:
4.0.0.0
File Version:
4.8.3761.0 built by: NET48REL1

4) Open project properties

4.1) under *Application* -> *Output type*: Set it to **Class Library**.

4.2) under *Build* -> *Output* -> *Output path*: Set it to Grasshopper "**Components Folder**" path.

You can get it through open Rhino -> Grasshopper -> file -> Special Folders -> Components Folder.

under *Build* -> *General* enable Allow Unsafe Code

4.3) under *Build Events* -> *Post-build event command line* enter the following:

```
Copy "$(TargetPath)" "$(TargetDir)\$(ProjectName).gha"
Erase "$(TargetPath)"
```

4.4) under *Debug* -> *Enable "start external program:"* set it to "<Program Files>\Rhino 8\System\Rhino.exe"

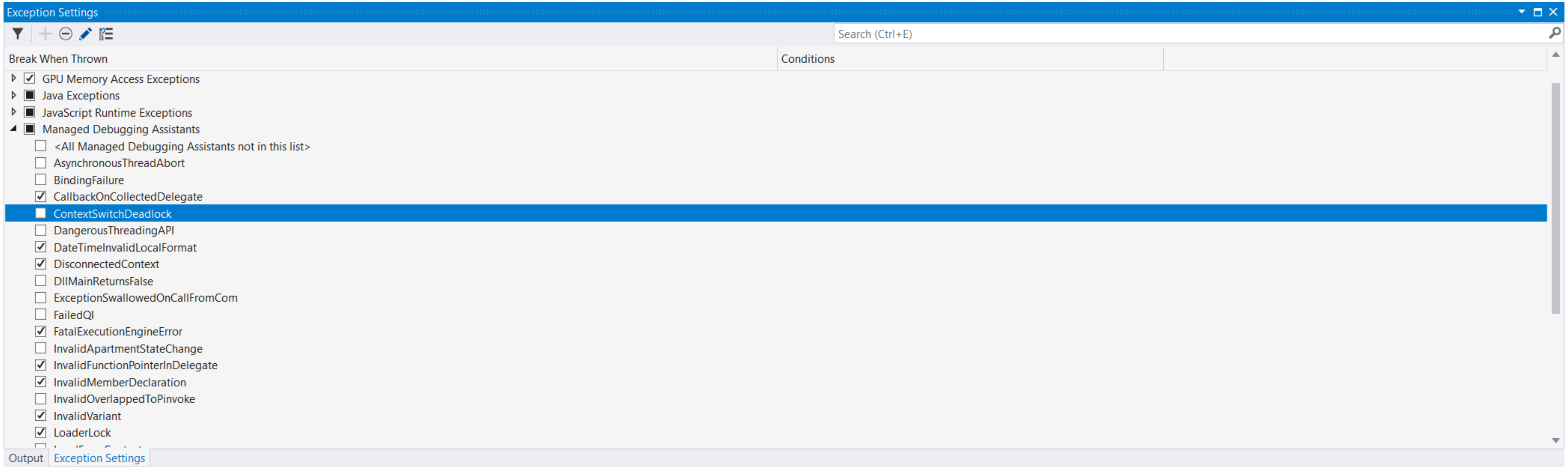
under *Debug* -> *Command Line Arguments* add

/netfx /nosplash

4.5) Save changes.

5) Open Exception Settings

5.1) under Managed Debugging Assistants -> Disable "ContextSwitchDeadlock".



Exception Settings

Search (Ctrl+E)

Break When Thrown | Conditions

- ☑ GPU Memory Access Exceptions
- ■ Java Exceptions
- ■ JavaScript Runtime Exceptions
- ■ Managed Debugging Assistants
  - ☐ <All Managed Debugging Assistants not in this list>
  - ☐ AsynchronousThreadAbort
  - ☐ BindingFailure
  - ☑ CallbackOnCollectedDelegate
  - ☐ ContextSwitchDeadlock
  - ☐ DangerousThreadingAPI
  - ☑ DateTimeInvalidLocalFormat
  - ☑ DisconnectedContext
  - ☐ DllMainReturnsFalse
  - ☐ ExceptionSwallowedOnCallFromCom
  - ☐ FailedQI
  - ☑ FatalExecutionEngineError
  - ☐ InvalidApartmentStateChange
  - ☑ InvalidFunctionPointerInDelegate
  - ☑ InvalidMemberDeclaration
  - ☐ InvalidOverlappedToPinvoke
  - ☑ InvalidVariant
  - ☑ LoaderLock

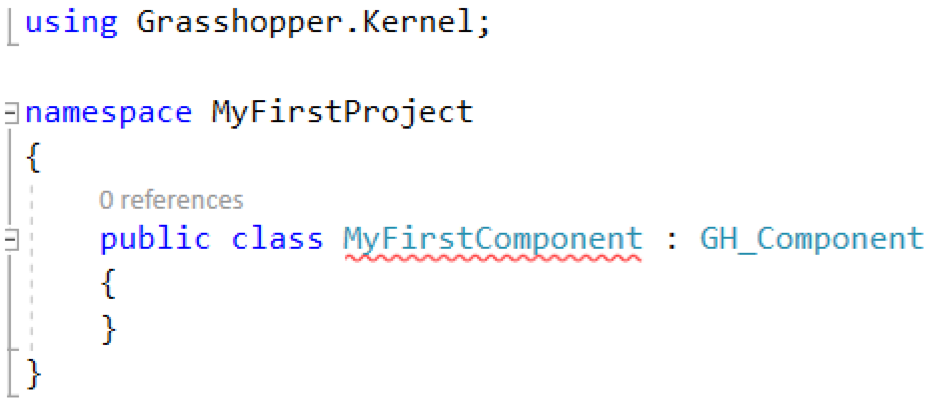
Create Component Class

1) Create a blank class (e.g. MyFirstComponent).

   1.1) Import “Grasshopper.Kernel” namespace.

   1.2) Make sure that the class is visible (public).

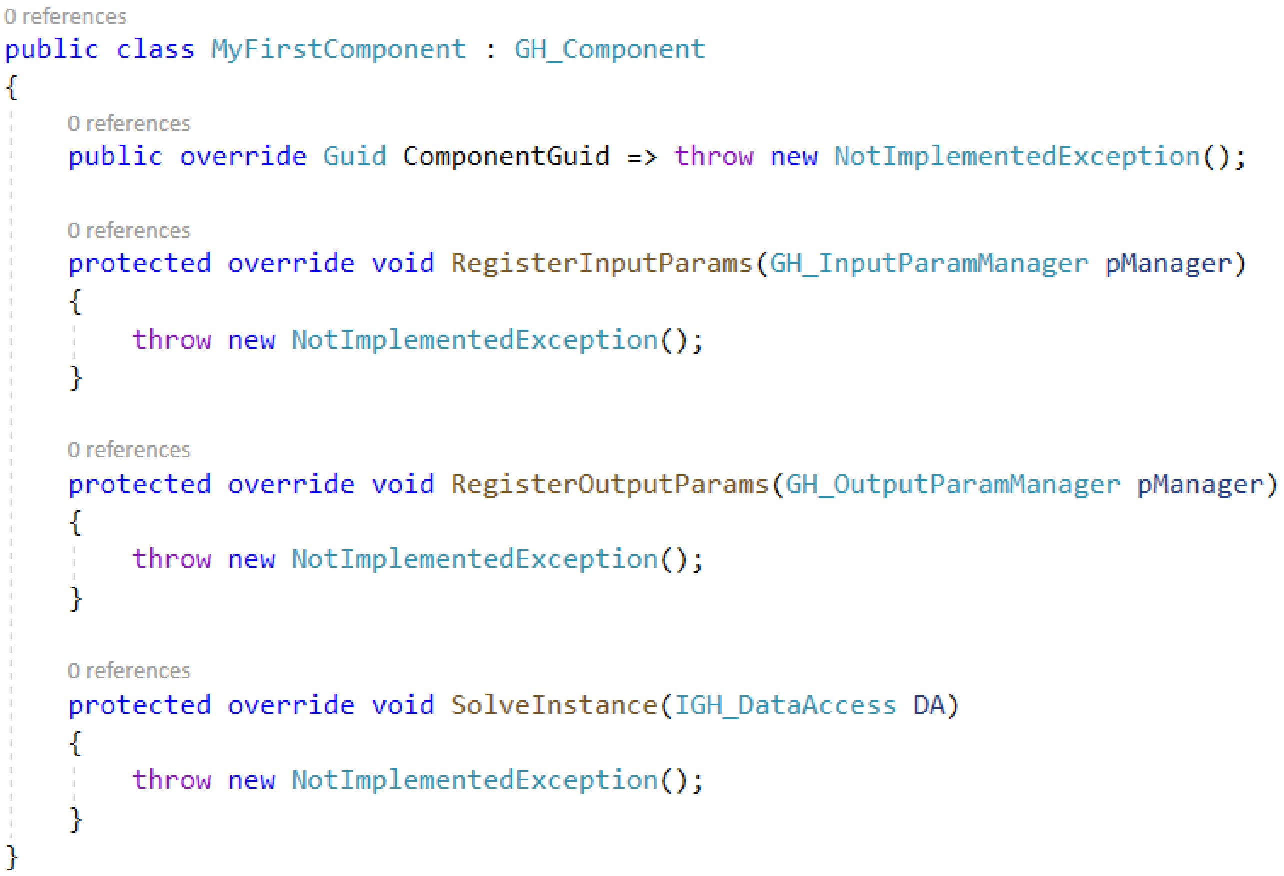
   1.3) Derive “MyFirstComponent” class from the “**GH_Component**” base class defined inside Grasshopper.



```csharp
using Grasshopper.Kernel;

namespace MyFirstProject
{
    public class MyFirstComponent : GH_Component
    {
    }
}
```

Cont.

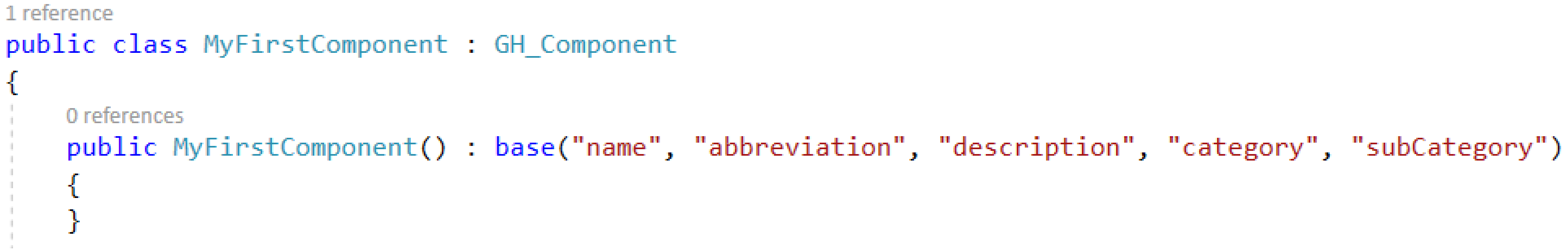
1.4) Deriving (inheriting) from “GH_Component” requires you to implement a number of methods. Visual Studio can insert default implementations for all of these via the “Implement Abstract Class” menu option:



```
0 references
public class MyFirstComponent : GH_Component
{
    0 references
    public override Guid ComponentGuid => throw new NotImplementedException();

    0 references
    protected override void RegisterInputParams(GH_InputParamManager pManager)
    {
        throw new NotImplementedException();
    }

    0 references
    protected override void RegisterOutputParams(GH_OutputParamManager pManager)
    {
        throw new NotImplementedException();
    }

    0 references
    protected override void SolveInstance(IGH_DataAccess DA)
    {
        throw new NotImplementedException();
    }
}
```

Cont.

1.5) Create the Component Constructor

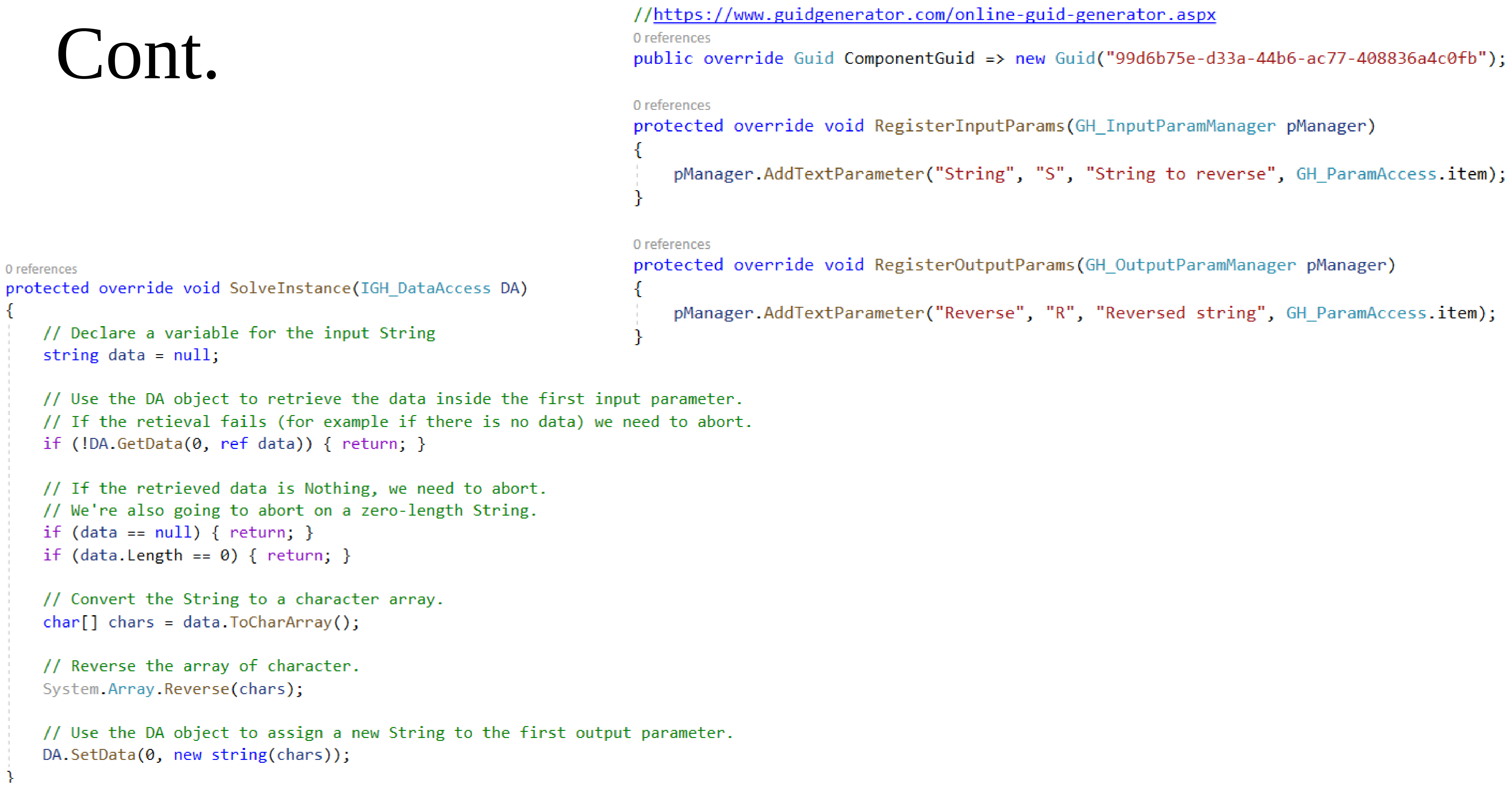


```
1 reference
public class MyFirstComponent : GH_Component
{
    0 references
    public MyFirstComponent() : base("name", "abbreviation", "description", "category", "subCategory")
    {
    }
```

1.6) Implement the abstarct methods.

1.7) For the Guid method, you should generate a Gui Id from this generator.

Cont.



```
//https://www.guidgenerator.com/online-guid-generator.aspx
0 references
public override Guid ComponentGuid => new Guid("99d6b75e-d33a-44b6-ac77-408836a4c0fb");

0 references
protected override void RegisterInputParams(GH_InputParamManager pManager)
{
    pManager.AddTextParameter("String", "S", "String to reverse", GH_ParamAccess.item);
}

0 references
protected override void RegisterOutputParams(GH_OutputParamManager pManager)
{
    pManager.AddTextParameter("Reverse", "R", "Reversed string", GH_ParamAccess.item);
}
```

```
0 references
protected override void SolveInstance(IGH_DataAccess DA)
{
    // Declare a variable for the input String
    string data = null;

    // Use the DA object to retrieve the data inside the first input parameter.
    // If the retieval fails (for example if there is no data) we need to abort.
    if (!DA.GetData(0, ref data)) { return; }

    // If the retrieved data is Nothing, we need to abort.
    // We're also going to abort on a zero-length String.
    if (data == null) { return; }
    if (data.Length == 0) { return; }

    // Convert the String to a character array.
    char[] chars = data.ToCharArray();

    // Reverse the array of character.
    System.Array.Reverse(chars);

    // Use the DA object to assign a new String to the first output parameter.
    DA.SetData(0, new string(chars));
}
```

Create Info Class

1) Create a blank class (e.g. MyFirstInfo).

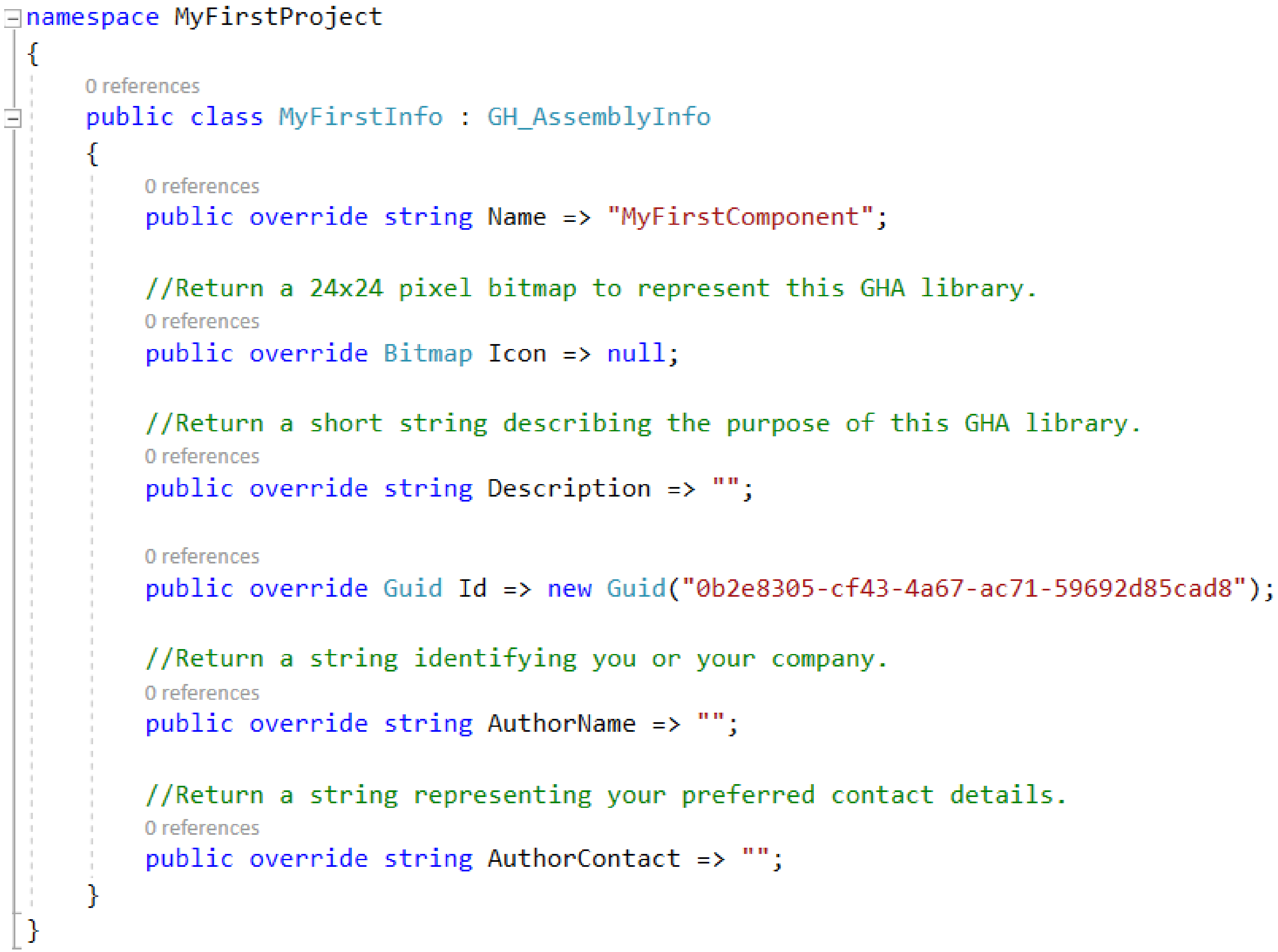
   1.1) Import “Grasshopper.Kernel’’ and “System.Drawing’’ namespaces.

   1.2) Make sure that the class is visible (public).

   1.3) Derive “MyFirstInfo’’ class from the “GH_AssemblyInfo’’ base class defined inside Grasshopper.

   1.4) Override this methods: Name, Icon, Description, Id, AuthorName and AuthorContact.

Cont.



```
namespace MyFirstProject
{
    0 references
    public class MyFirstInfo : GH_AssemblyInfo
    {
        0 references
        public override string Name => "MyFirstComponent";

        //Return a 24x24 pixel bitmap to represent this GHA library.
        0 references
        public override Bitmap Icon => null;

        //Return a short string describing the purpose of this GHA library.
        0 references
        public override string Description => "";

        0 references
        public override Guid Id => new Guid("0b2e8305-cf43-4a67-ac71-59692d85cad8");

        //Return a string identifying you or your company.
        0 references
        public override string AuthorName => "";

        //Return a string representing your preferred contact details.
        0 references
        public override string AuthorContact => "";
    }
}
```

Resources / Useful Links

- Project Setup
- Simple Component
- Rhino API
- Grasshopper API
- GUID Generator