

0. Summary

1. New Project

1.1. Create New Project : Sample

2. Sample : Program.cs

0. Summary

What to learn :

OrderBy

OrderByDescending

ThenBy

ThenByDescending

Reverse

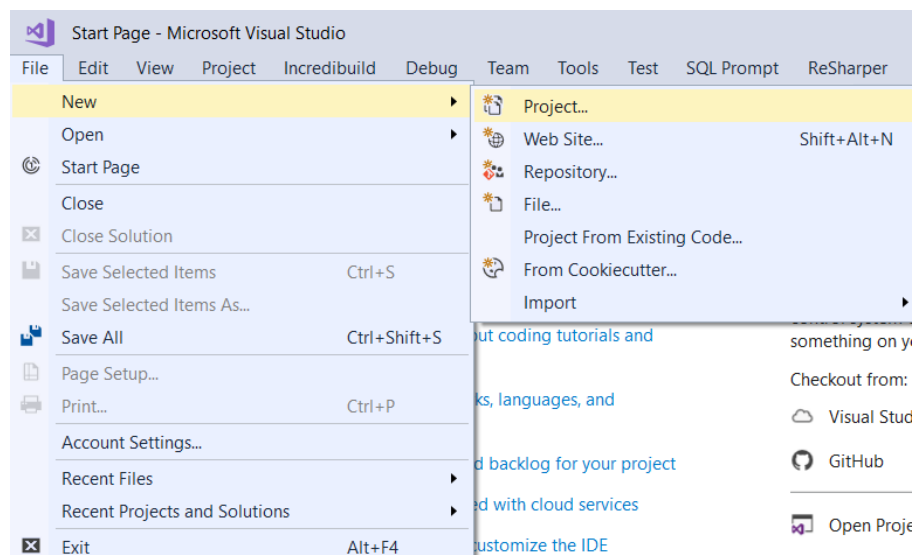
1. New Project

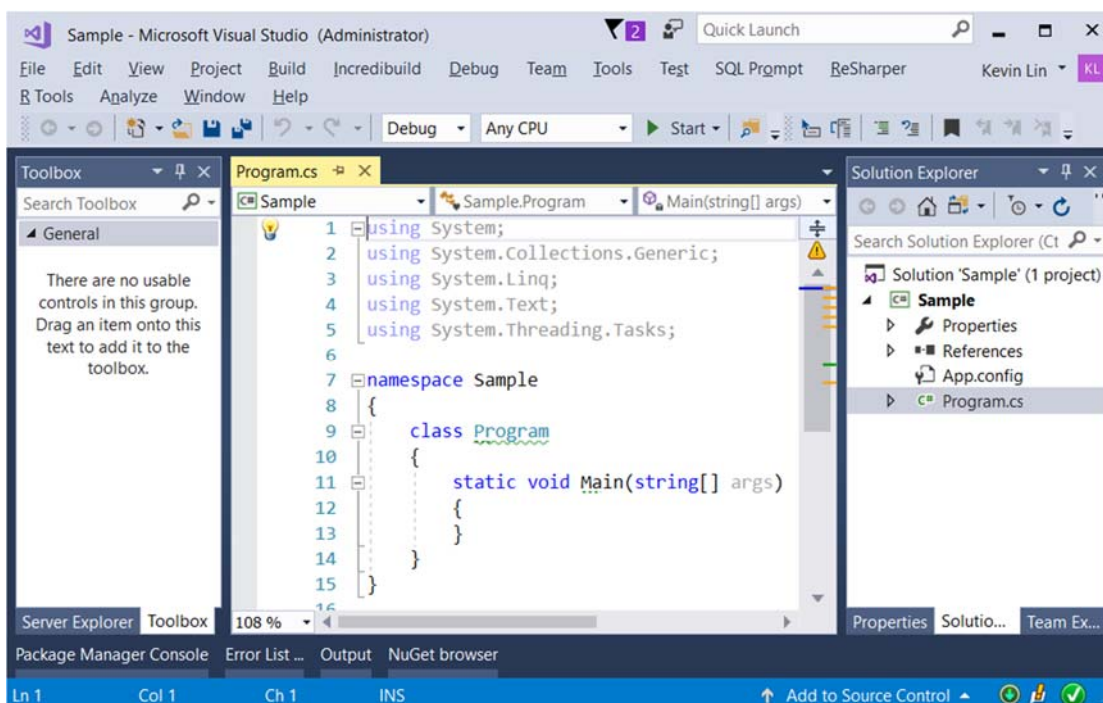
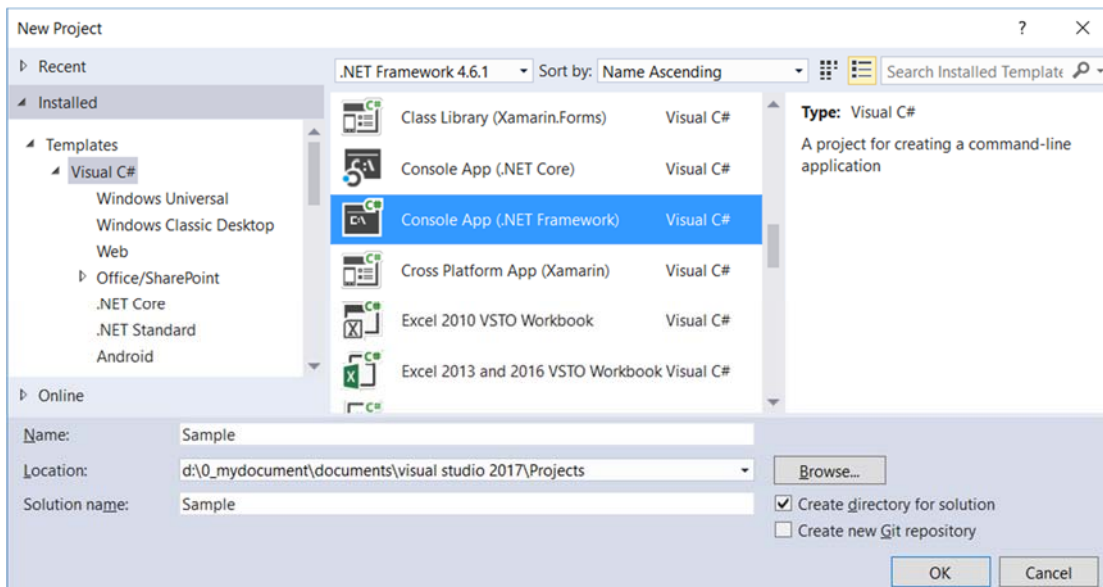
1.1. Create New Project : Sample

File --> New --> Project... -->

Visual C# --> **Console App (.Net Framework)** -->

Name: **Sample**





=====

2. Sample : Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using OnlineGame;
namespace Sample
{
    class Program
    {
        static void Main(string[] args)
        {
```

```

// 1. =====
//GamerOrderByName
Console.WriteLine("1. GamerOrderByName() ===== ");
GamerOrderByName();
// 2. =====
//GamerOrderByNameSqlLikeQuery
Console.WriteLine("2. GamerOrderByNameSqlLikeQuery() ===== ");
GamerOrderByNameSqlLikeQuery();
// 3. =====
//GamerOrderByNameDescending
Console.WriteLine("3. GamerOrderByNameDescending() ===== ");
GamerOrderByNameDescending();
// 4. =====
//GamerOrderByNameDescendingSqlLikeQuery
Console.WriteLine("4. GamerOrderByNameDescendingSqlLikeQuery() ===== ");
GamerOrderByNameDescendingSqlLikeQuery();
// 5. =====
//OrderByScoreByNameById
Console.WriteLine("5. OrderByScoreByNameById() ===== ");
OrderByScoreByNameById();
// 6. =====
//OrderByScoreByNameByIdSqlLikeQuery
Console.WriteLine("6. OrderByScoreByNameByIdSqlLikeQuery() ===== ");
OrderByScoreByNameByIdSqlLikeQuery();
// 7. =====
//OrderByScoreByNameByIdDescending
Console.WriteLine("7. OrderByScoreByNameByIdDescending() ===== ");
OrderByScoreByNameByIdDescending();
// 8. =====
//OrderByScoreByNameByIdDescendingSqlLikeQuery
Console.WriteLine("8. OrderByScoreByNameByIdDescendingSqlLikeQuery() ===== ");
OrderByScoreByNameByIdDescendingSqlLikeQuery();
// 9. =====
//ReverseSample
Console.WriteLine("9. ReverseSample() ===== ");
ReverseSample();
Console.ReadLine();
}

```

```

// 1. =====
static void GamerOrderByName()
{
    IEnumerable<Gamer> gamers =
        GamerHelper.GetSampleGamers()
            .OrderBy(g => g.Name);
    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
}
//Order by Name
//Id==1,Name==NameA,Score==2000
//Id==2,Name==NameA,Score==2000
//Id==3,Name==NameB,Score==2000
//Id==5,Name==NameC,Score==2500

```

```

//Id==4,Name==NameD,Score==2000
// 2. =====
private static void GamerOrderByNameSqlLikeQuery()
{
    IOrderedEnumerable<Gamer> gamers =
        from gamer in GamerHelper.GetSampleGamers()
        orderby gamer.Name
        select gamer;
    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
}
//Order by Name
//Id==1,Name==NameA,Score==2000
//Id==2,Name==NameA,Score==2000
//Id==3,Name==NameB,Score==2000
//Id==5,Name==NameC,Score==2500
//Id==4,Name==NameD,Score==2000
// 3. =====
private static void GamerOrderByNameDescending()
{
    IOrderedEnumerable<Gamer> gamers =
        GamerHelper.GetSampleGamers()
        .OrderByDescending(g => g.Name);
    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
}
//Descending Order by Name
//Id==4,Name==NameD,Score==2000
//Id==5,Name==NameC,Score==2500
//Id==3,Name==NameB,Score==2000
//Id==1,Name==NameA,Score==2000
//Id==2,Name==NameA,Score==2000
// 4. =====
static void GamerOrderByNameDescendingSqlLikeQuery()
{
    IOrderedEnumerable<Gamer> gamers =
        from gamer in GamerHelper.GetSampleGamers()
        orderby gamer.Name descending
        select gamer;
    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
}
//Descending Order by Name
//Id==4,Name==NameD,Score==2000
//Id==5,Name==NameC,Score==2500
//Id==3,Name==NameB,Score==2000
//Id==1,Name==NameA,Score==2000
//Id==2,Name==NameA,Score==2000

```

```
// 5. =====
```

```
static void OrderByScoreByNameById()
```

```
{
    IOrderedEnumerable<Gamer> gamers =
        GamerHelper.GetSampleGamers()
            .OrderBy(g => g.Score)
            .ThenBy(g => g.Name)
            .ThenBy(g => g.Id);
    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
}
```

```
//Order by Score, Name, Id
```

```
//Id==1,Name==NameA,Score==2000
```

```
//Id==2,Name==NameA,Score==2000
```

```
//Id==3,Name==NameB,Score==2000
```

```
//Id==4,Name==NameD,Score==2000
```

```
//Id==5,Name==NameC,Score==2500
```

```
// 6. =====
```

```
static void OrderByScoreByNameByIdSqlLikeQuery()
```

```
{
    IOrderedEnumerable<Gamer> gamers = from gamer in GamerHelper.GetSampleGamers()
                                        orderby gamer.Score, gamer.Name, gamer.Id
                                        select gamer;

    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
}
```

```
//Order by Score, Name, Id
```

```
//Id==1,Name==NameA,Score==2000
```

```
//Id==2,Name==NameA,Score==2000
```

```
//Id==3,Name==NameB,Score==2000
```

```
//Id==4,Name==NameD,Score==2000
```

```
//Id==5,Name==NameC,Score==2500
```

```
// 7. =====
```

```
private static void OrderByScoreByNameByIdDescending()
```

```
{
    IOrderedEnumerable<Gamer> gamers =
        GamerHelper.GetSampleGamers()
            .OrderByDescending(g => g.Score)
            .ThenBy(g => g.Name)
            .ThenBy(g => g.Id);
    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
}
```

```
//Descending Order By Score,
```

```
//then Order by Name and Id
```

```
//Id==5,Name==NameC,Score==2500
```

```
//Id==1,Name==NameA,Score==2000
```

```
//Id==2,Name==NameA,Score==2000
```

```
//Id==3,Name==NameB,Score==2000
```

```

//Id==4,Name==NameD,Score==2000
// 8. =====
static void OrderByScoreByNameByIdDescendingSqlLikeQuery()
{
    IOrderedEnumerable<Gamer> gamers =
        from gamer in GamerHelper.GetSampleGamers()
        orderby gamer.Score descending, gamer.Name, gamer.Id
        select gamer;
    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
}
//Descending Order By Score,
//then Order by Name and Id
//Id==5,Name==NameC,Score==2500
//Id==1,Name==NameA,Score==2000
//Id==2,Name==NameA,Score==2000
//Id==3,Name==NameB,Score==2000
//Id==4,Name==NameD,Score==2000
// 9. =====
static void ReverseSample()
{
    List<Gamer> gamersList = GamerHelper.GetSampleGamers();
    Console.WriteLine("9.1. print each item ----- ");
    foreach (Gamer gamersItem in gamersList)
    {
        Console.WriteLine(gamersItem);
    }
    //Id == 1,Name == NameA,Score == 2000
    //Id == 2,Name == NameA,Score == 2000
    //Id == 3,Name == NameB,Score == 2000
    //Id == 4,Name == NameD,Score == 2000
    //Id == 5,Name == NameC,Score == 2500
    Console.WriteLine("9.2. Reverse and then print each item ----- ");
    gamersList.Reverse();
    foreach (Gamer gamersItem in gamersList)
    {
        Console.WriteLine(gamersItem);
    }
    //Id == 5,Name == NameC,Score == 2500
    //Id == 4,Name == NameD,Score == 2000
    //Id == 3,Name == NameB,Score == 2000
    //Id == 2,Name == NameA,Score == 2000
    //Id == 1,Name == NameA,Score == 2000
    Console.WriteLine("9.3. order by Score, Name, Id ----- ");
    IOrderedEnumerable<Gamer> gamers = gamersList
        .OrderBy(g => g.Score)
        .ThenBy(g => g.Name)
        .ThenBy(g => g.Id);
    foreach (Gamer gamersItem in gamers)
    {
        Console.WriteLine(gamersItem);
    }
    //Id == 1,Name == NameA,Score == 2000
    //Id == 2,Name == NameA,Score == 2000

```

```

        //Id == 3,Name == NameB,Score == 2000
        //Id == 4,Name == NameD,Score == 2000
        //Id == 5,Name == NameC,Score == 2500
        Console.WriteLine("9.4. Reverse and then print each item ----- ");
        foreach (Gamer gamersItem in gamers.Reverse())
        {
            Console.WriteLine(gamersItem);
        }
        //Id == 5,Name == NameC,Score == 2500
        //Id == 4,Name == NameD,Score == 2000
        //Id == 3,Name == NameB,Score == 2000
        //Id == 2,Name == NameA,Score == 2000
        //Id == 1,Name == NameA,Score == 2000
    }
}

namespace OnlineGame
{
    public class Gamer
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int Score { get; set; }
        public override string ToString()
        {
            return $"Id=={Id},Name=={Name},Score=={Score}";
        }
    }
    public class GamerHelper
    {
        public static List<Gamer> GetSampleGamers()
        {
            return new List<Gamer>
            {
                new Gamer{Id=1,Name="NameA",Score=2000},
                new Gamer{Id=2,Name="NameA",Score=2000},
                new Gamer{Id=3,Name="NameB",Score=2000},
                new Gamer{Id=4,Name="NameD",Score=2000},
                new Gamer{Id=5,Name="NameC",Score=2500}
            };
        }
    }
}

```



```
1. GamerOrderByName() =====
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
Id==3,Name==NameB,Score==2000
Id==5,Name==NameC,Score==2500
Id==4,Name==NameD,Score==2000
2. GamerOrderByNameSqlLikeQuery() =====
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
Id==3,Name==NameB,Score==2000
Id==5,Name==NameC,Score==2500
Id==4,Name==NameD,Score==2000
3. GamerOrderByNameDescending() =====
Id==4,Name==NameD,Score==2000
Id==5,Name==NameC,Score==2500
Id==3,Name==NameB,Score==2000
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
4. GamerOrderByNameDescendingSqlLikeQuery() =====
Id==4,Name==NameD,Score==2000
Id==5,Name==NameC,Score==2500
Id==3,Name==NameB,Score==2000
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
5. OrderByScoreByNameById() =====
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
Id==3,Name==NameB,Score==2000
Id==4,Name==NameD,Score==2000
Id==5,Name==NameC,Score==2500
6. OrderByScoreByNameByIdSqlLikeQuery() =====
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
Id==3,Name==NameB,Score==2000
Id==4,Name==NameD,Score==2000
Id==5,Name==NameC,Score==2500
```

```
7. OrderByScoreByNameByIdDescending() =====
Id==5,Name==NameC,Score==2500
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
Id==3,Name==NameB,Score==2000
Id==4,Name==NameD,Score==2000
8. OrderByScoreByNameByIdDescendingSqlLikeQuery() =====
Id==5,Name==NameC,Score==2500
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
Id==3,Name==NameB,Score==2000
Id==4,Name==NameD,Score==2000
```



```
9. ReverseSample() =====
9.1. print each item -----
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
Id==3,Name==NameB,Score==2000
Id==4,Name==NameD,Score==2000
Id==5,Name==NameC,Score==2500
9.2. Reverse and then print each item -----
Id==5,Name==NameC,Score==2500
Id==4,Name==NameD,Score==2000
Id==3,Name==NameB,Score==2000
Id==2,Name==NameA,Score==2000
Id==1,Name==NameA,Score==2000
9.3. order by Score, Name, Id -----
Id==1,Name==NameA,Score==2000
Id==2,Name==NameA,Score==2000
Id==3,Name==NameB,Score==2000
Id==4,Name==NameD,Score==2000
Id==5,Name==NameC,Score==2500
9.4. Reverse and then print each item -----
Id==5,Name==NameC,Score==2500
Id==4,Name==NameD,Score==2000
Id==3,Name==NameB,Score==2000
Id==2,Name==NameA,Score==2000
Id==1,Name==NameA,Score==2000
```