
# Object Oriented Programming

# INTRODUCTION

OOP stands for **Object Oriented Programming**.

It is a programming methodology that uses Objects to build a system or web applications using programming languages like C#, Vb.net etc.

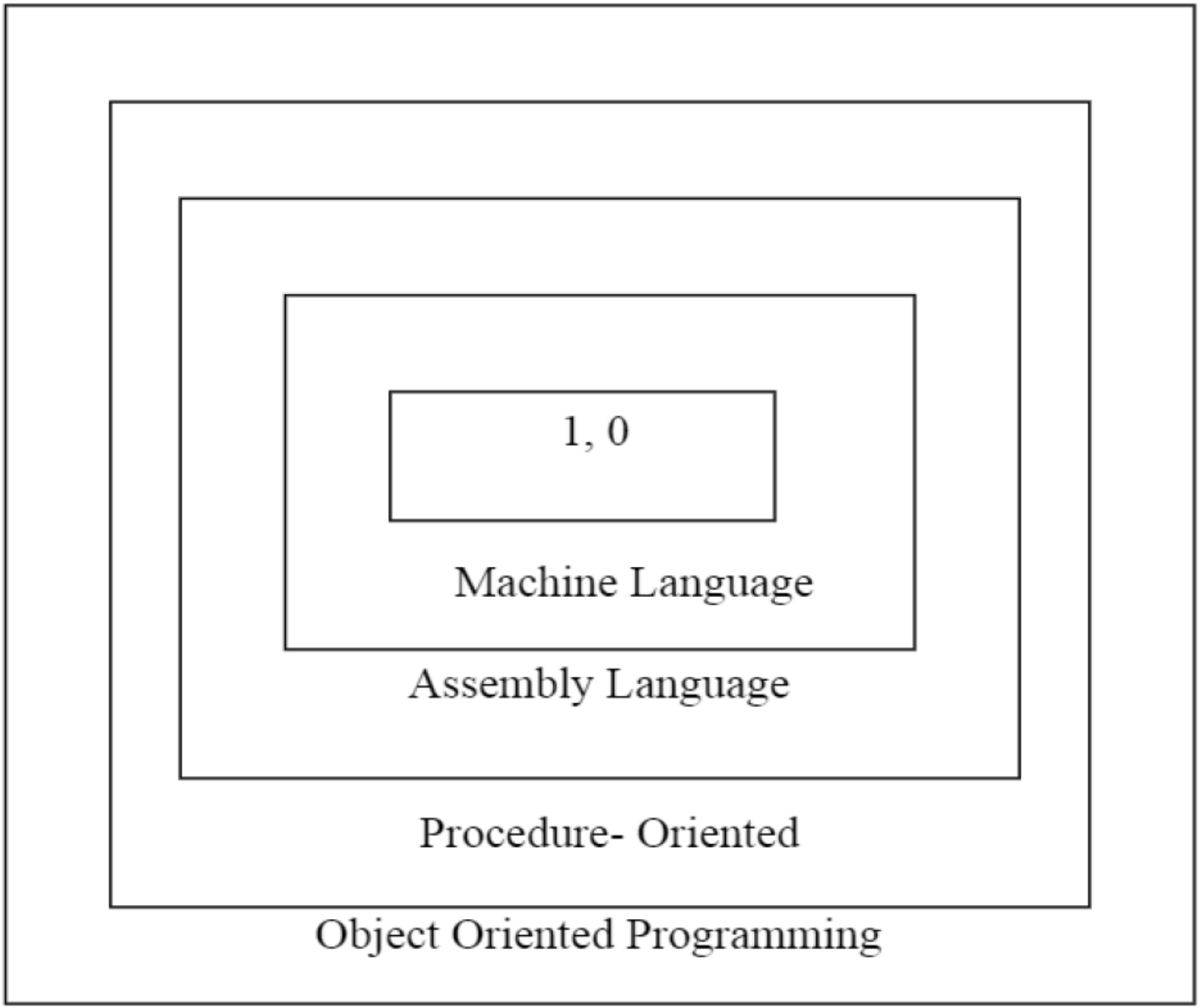
Here, **Objects** plays a very important role because it hides the implementation details and exposed only the needed functionalities and related stuff that is required to adopt it. We can access class properties and methods by creating class object.

# Programming Languages

- Programming languages allow programmers to code software.

- The three major families of languages are:

  - Machine languages
  - Assembly languages
  - High-Level languages


1, 0
Machine Language
Assembly Language
Procedure- Oriented
Object Oriented Programming

# Machine Language

- Comprised of 0 , 1

  **1110100010101 111010101110**

- The “native” language of a computer
- Difficult to program

  one misplaced 1 or 0 will cause the program to fail

# Assembly Language

- step towards easier programming

  **ADD  1001010, 1011010**

- set of commands which are tied to a specific processor.
- needs to be translated to machine language before the computer processes it.

# High-Level Language

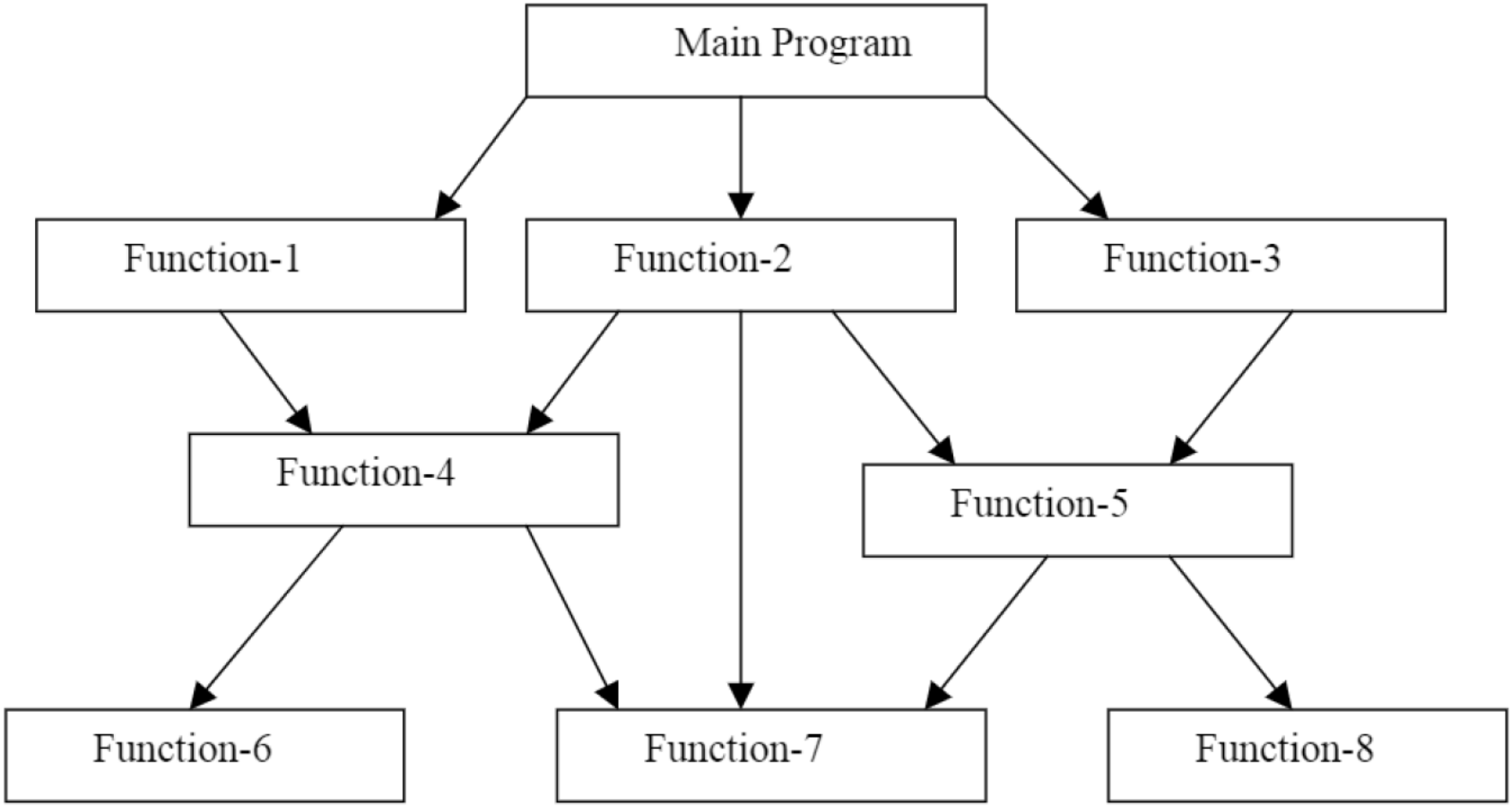
- represent easier programming.
- The syntax of HL languages is similar to English.
- we divide HL languages into two groups:
  - Procedural Oriented programming languages (POP)
  - Object-Oriented programming languages (OOP)

# Procedural Languages (POP)

- Early high-level languages are typically called procedural languages.
- Procedural languages are characterized by sequential sets of linear commands. The focus of such languages is on *structure*.
- Examples include C, COBOL, Fortran, LISP, Perl, HTML, VBScript

- **Examples of OOP languages include:**

**C++, Visual Basic.NET and Java.**

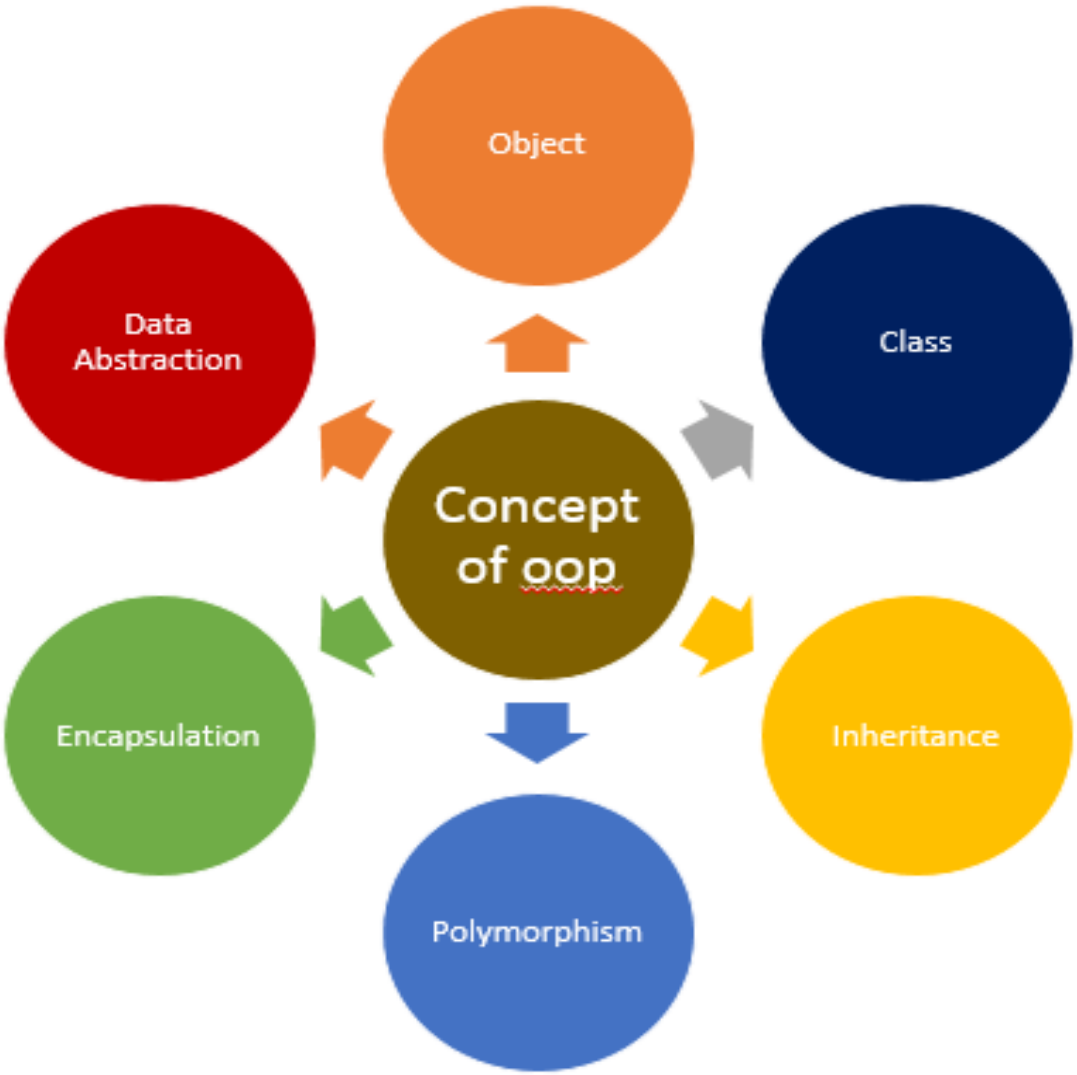



Typical structure of procedural oriented programs

Object-Oriented Programming (OOP) is an approach to program organization and development, which attempts to eliminate some of the pitfalls of conventional programming methods by incorporating the best of structured programming features with several new concepts.

It is a new way of organizing and developing programs and has nothing to do with any particular language. However, not all languages are suitable to implement OOP concepts easily.

# Basic Concepts of OOP



Object-oriented programming is a

**programming methodology**

characterized by the following concepts:

1. **Data Abstraction**: problem solving via the formulation of abstract data types (ADT's).
2. **Encapsulation**: the proximity of data definitions and operation definitions.
3. **Information hiding**: the ability to selectively hide implementation details of a given ADT.
4. **Polymorphism**: the ability to manipulate different kinds of objects, with only one operation.
5. **Inheritance**: the ability of objects of one data type, to inherit operations and data from another data type.
6. **Class - Object**

Procedural programming is about writing procedures or methods that perform operations on the data,

while object-oriented programming is about creating objects that contain both data and methods.

Object-oriented programming has several advantages over procedural programming:

- **OOP is faster and easier to execute**
- **OOP provides a clear structure for the programs**
- **OOP helps to keep the C# code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug**
- **OOP makes it possible to create full reusable applications with less code and shorter development time**

- **Some of the features of object oriented programming are:**

- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Data structures are designed such that they characterize the objects.
- Functions that operate on the data of an object are ties together in the data structure.
- Data is hidden and cannot be accessed by external function.
- Objects may communicate with each other through function.
- New data and functions can be easily added whenever necessary.
- Follows bottom up approach in program design.

# Comparison of POP and OOP

| POP | OOP |
| --- | --- |
| Focus is on the function | Focus is on the data |
| Data is not secure and can be corrupted | Data is secure |
| Use top down programming design | Use bottom up programming design |
| Does not model real world problem | Models real world problem |
| Programs are divided into function | Program are divided into object |

| Structured Approach | Object Oriented Approach |
|---|---|
| It works with Top-down approach. | It works with Bottom-up approach. |
| Program is divided into number of submodules or functions. | Program is organized by having number of classes and objects. |
| Function call is used. | Message passing is used. |
| Software reuse is not possible. | Reusability is possible. |
| Structured design programming usually left until end phases. | Object oriented design programming done concurrently with other phases. |
| Structured Design is more suitable for offshoring. | It is suitable for in-house development. |
| It shows clear transition from design to implementation. | Not so clear transition from design to implementation. |
| It is suitable for real time system, embedded system and projects where objects are not the most useful level of abstraction. | It is suitable for most business applications, game development projects, which are expected to customize or extended. |
| DFD & E-R diagram model the data. | Class diagram, sequence diagram, state chart diagram, and use cases all contribute. |
| In this, projects can be managed easily due to clearly identifiable phases. | In this approach, projects can be difficult to manage due to uncertain transitions between phase. |

# Benefits of OOP

OOP offers several benefits to both the program designer and the user. The new technology promises greater programmer productivity, the better quality of software, and lesser maintenance cost. The principal advantages are:

1.Through inheritance, we can eliminate redundant code and extend the use of existing classes.

2.We can build programs from the standard working modules that communicate with one another, rather than having to start writing the code from scratch. This leads to saving of development time and higher productivity.

3.The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program.

4.It is possible to have multiple objects to coexist without any interference.

5.It is possible to map objects in the problem domain to those objects in the program.

6.It is easy to partition the work in a project based on objects.

7.The data-centered design approach enables us to capture more details of a model in an implementable form.

8.Object-oriented systems can be easily upgraded from small to large systems.

9.Message passing techniques for communication between objects make the interface descriptions with external systems much simpler.

10.Software complexity can be easily managed.

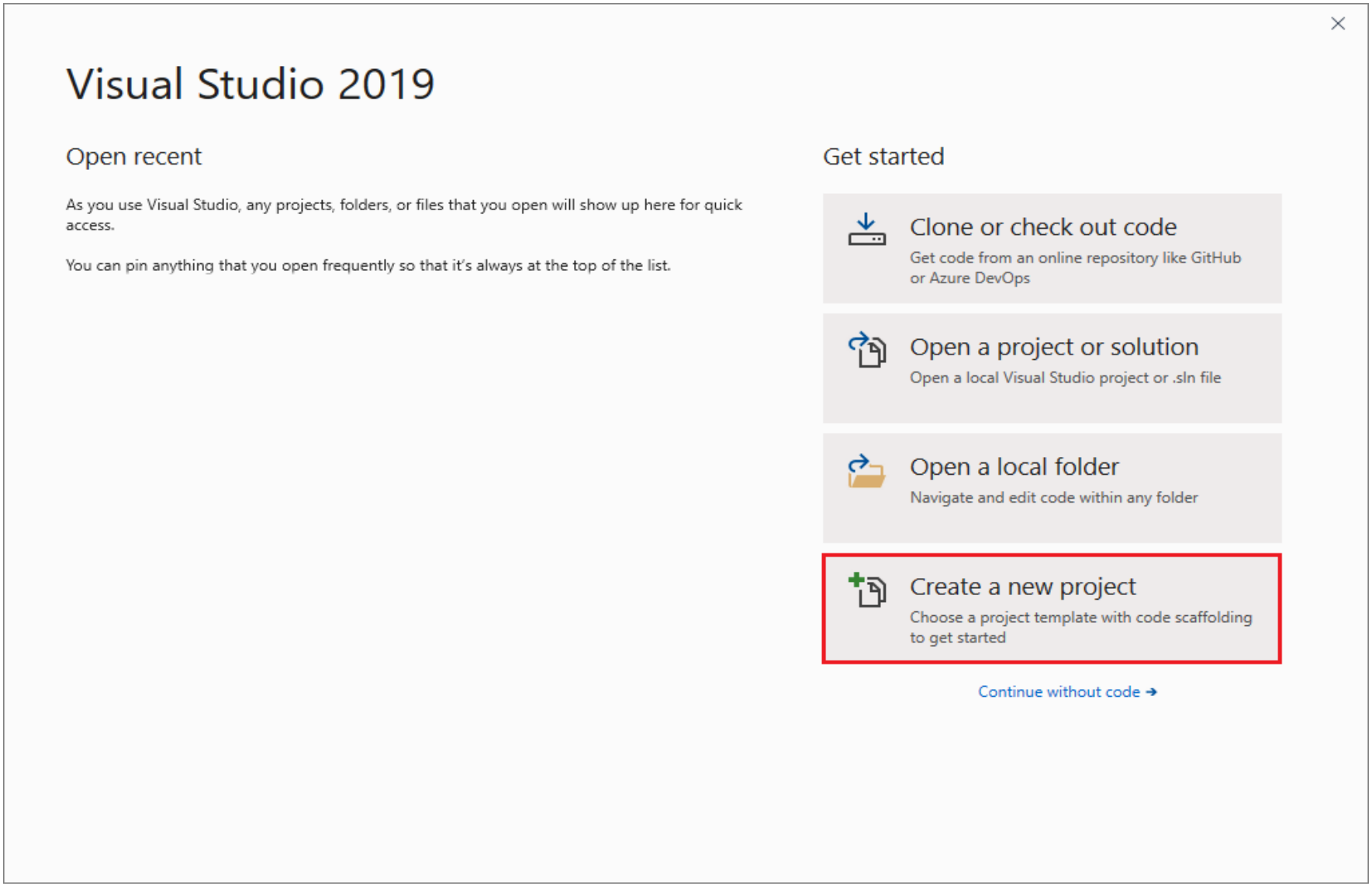
11.Reusability

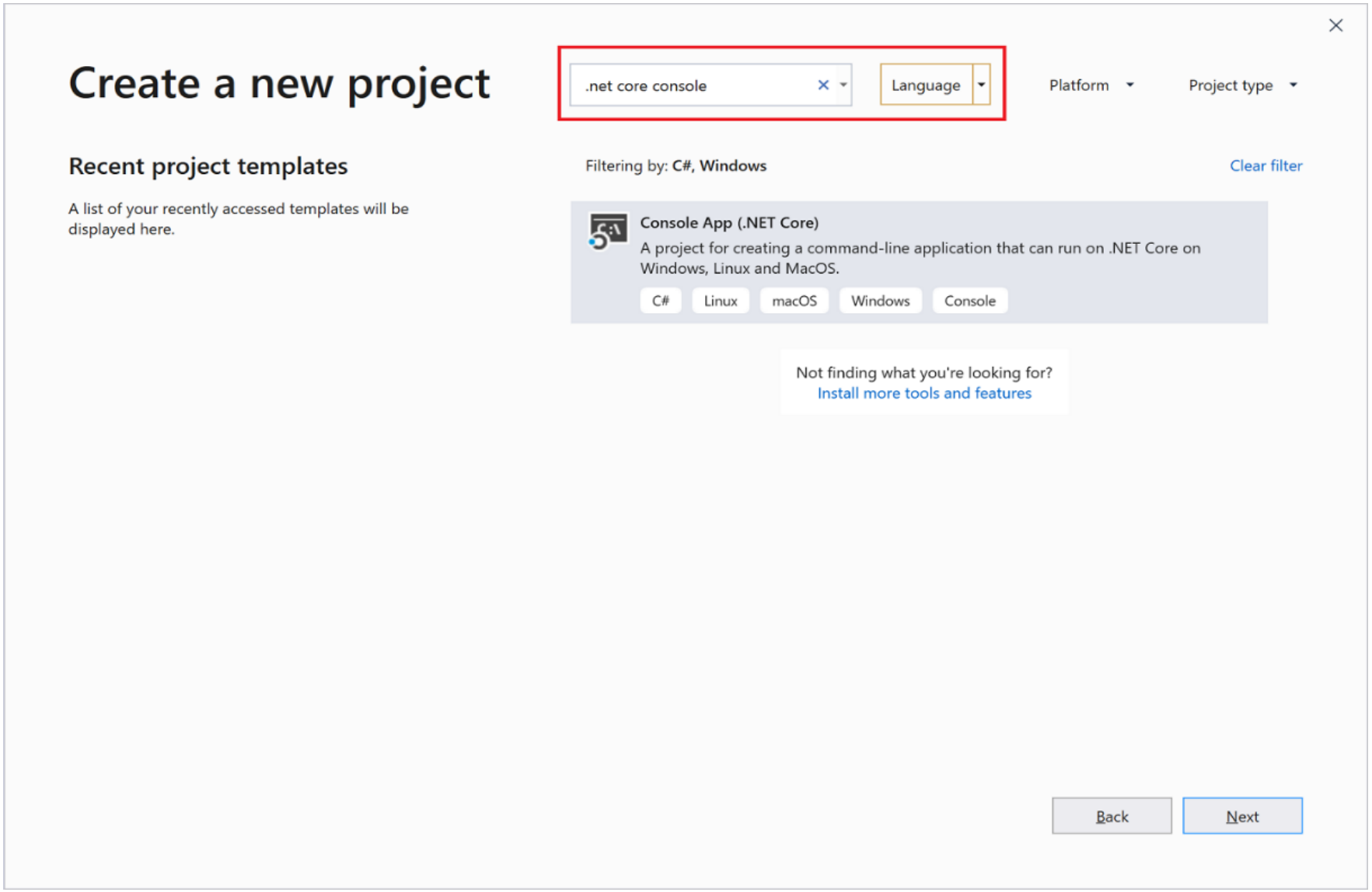
## Application of OOP

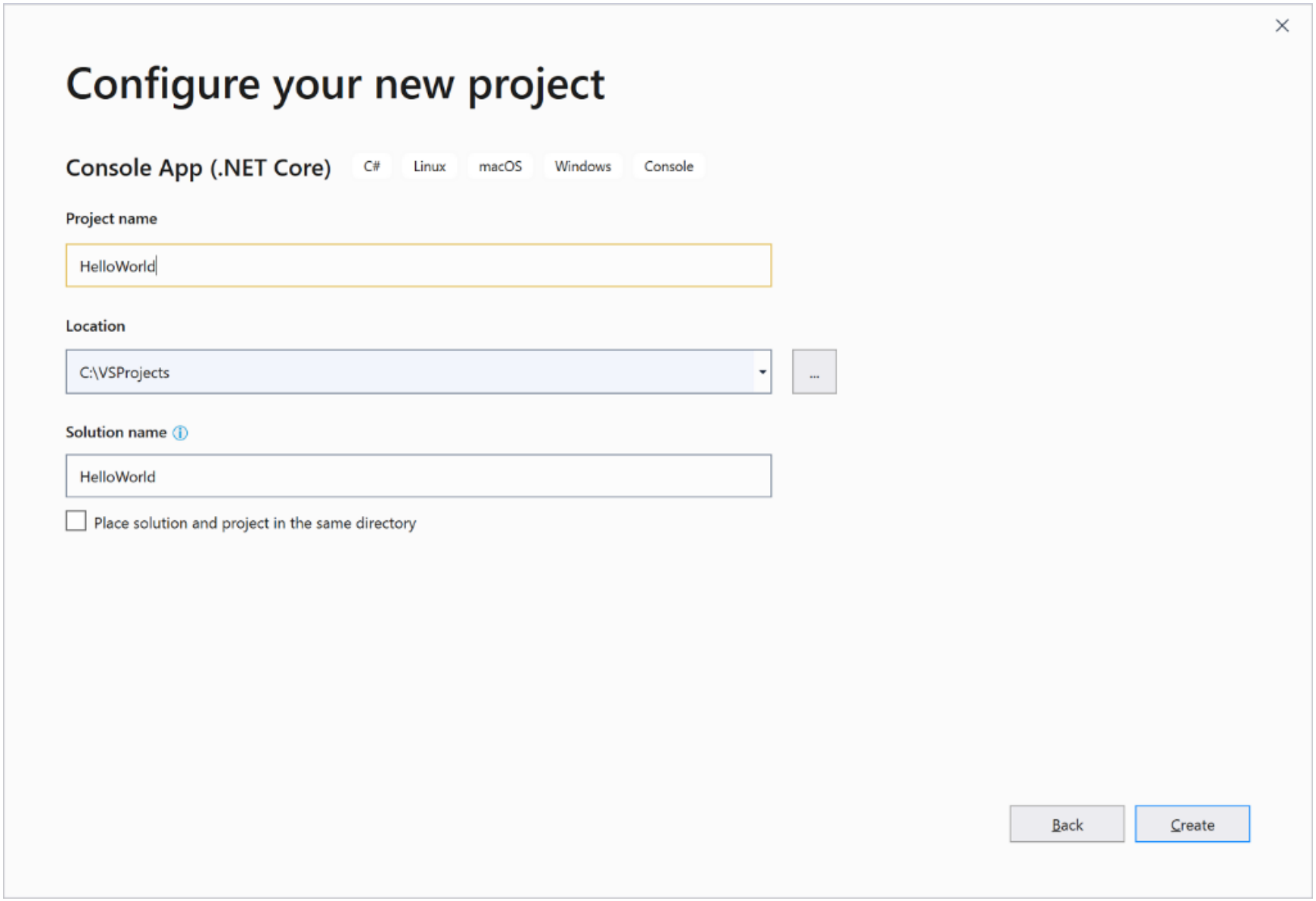
- Real-time system
- Simulation and modeling
- Object-oriented data bases
- AI and expert systems
- Neural networks and parallel programming
- Decision support and office automation systems
- CAM/CAD systems

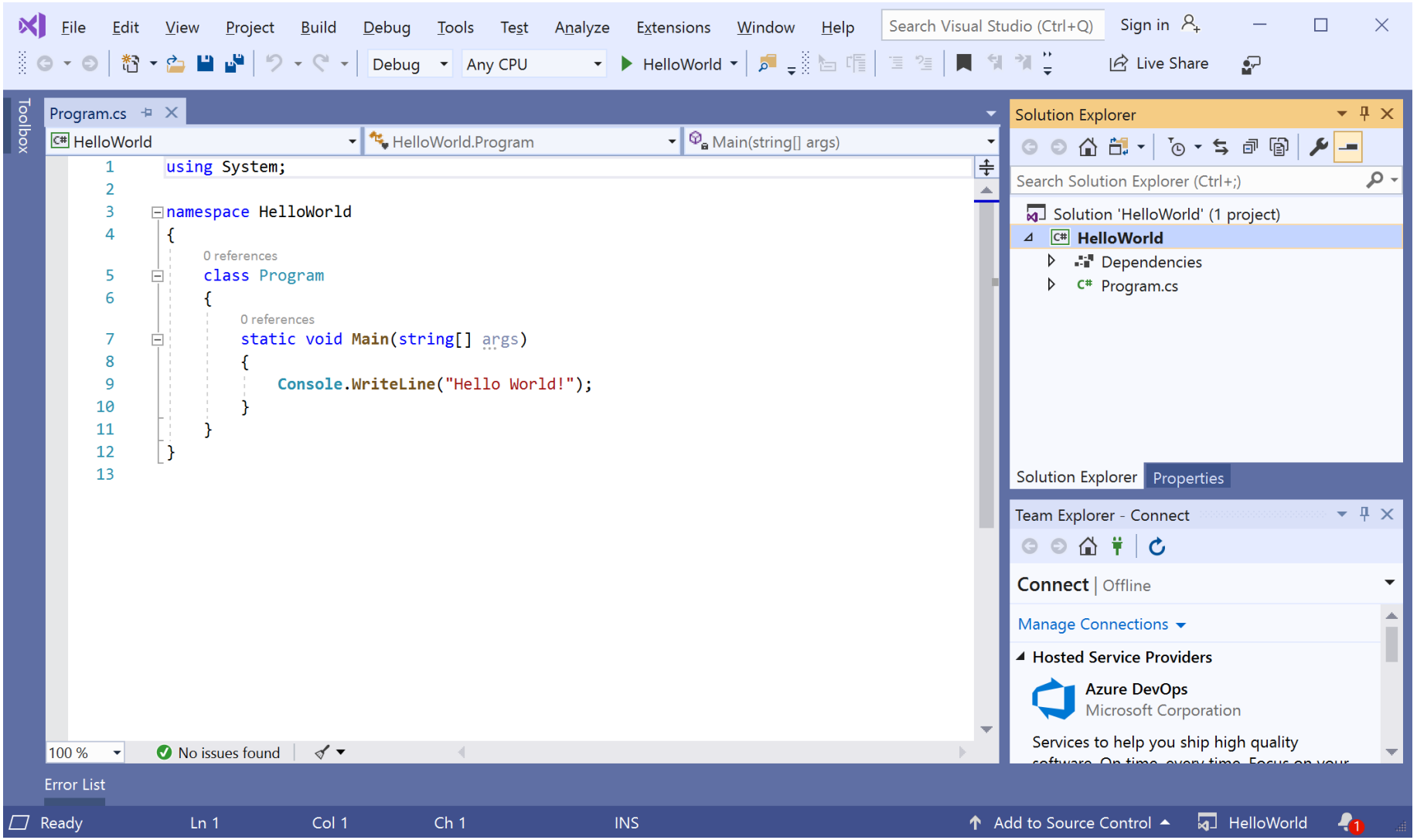
OOP environment will enable the software industry to improve not only the quality of software system but also its productivity. Object-oriented technology is certainly going to change the way the software engineers think, analyze, design and implement future system.

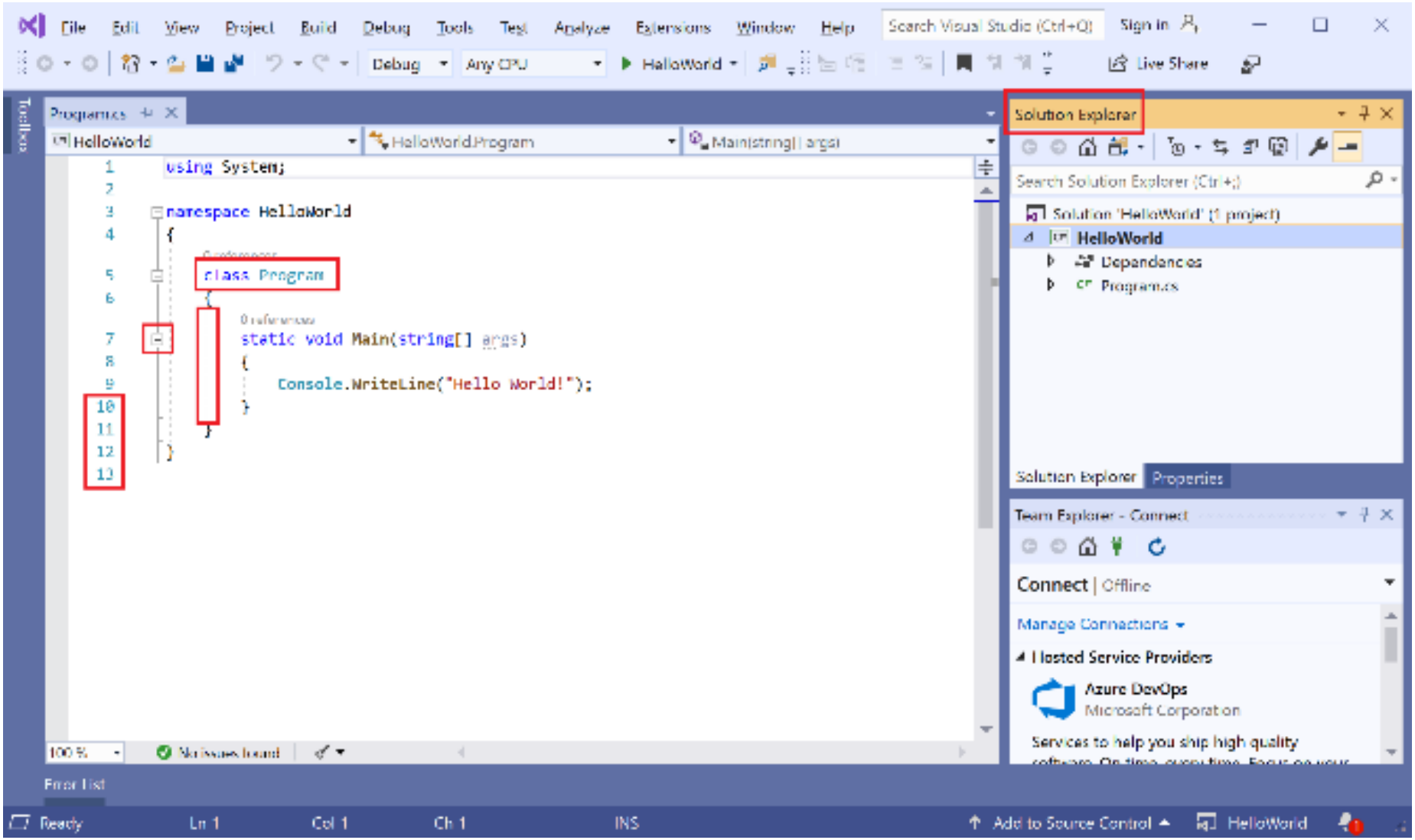
# Visual C# IDE

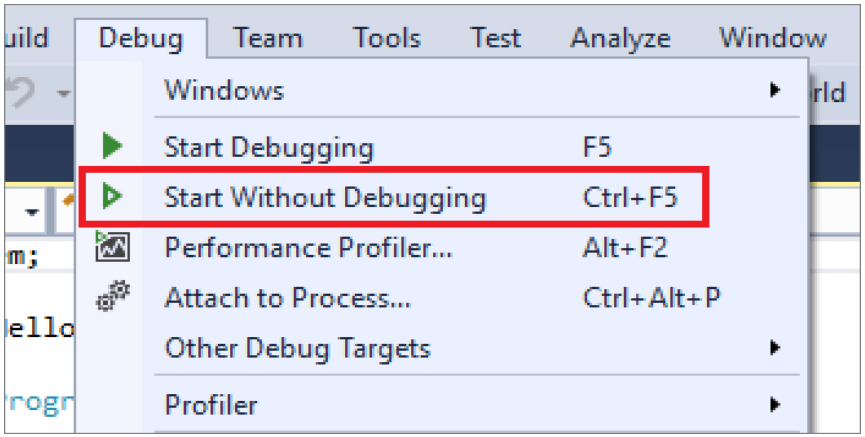

Visual Studio 2019
Open recent
As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.
You can pin anything that you open frequently so that it's always at the top of the list.
Get started
Clone or check out code
Get code from an online repository like GitHub or Azure DevOps
Open a project or solution
Open a local Visual Studio project or .sln file
Open a local folder
Navigate and edit code within any folder
Create a new project
Choose a project template with code scaffolding to get started
Continue without code →

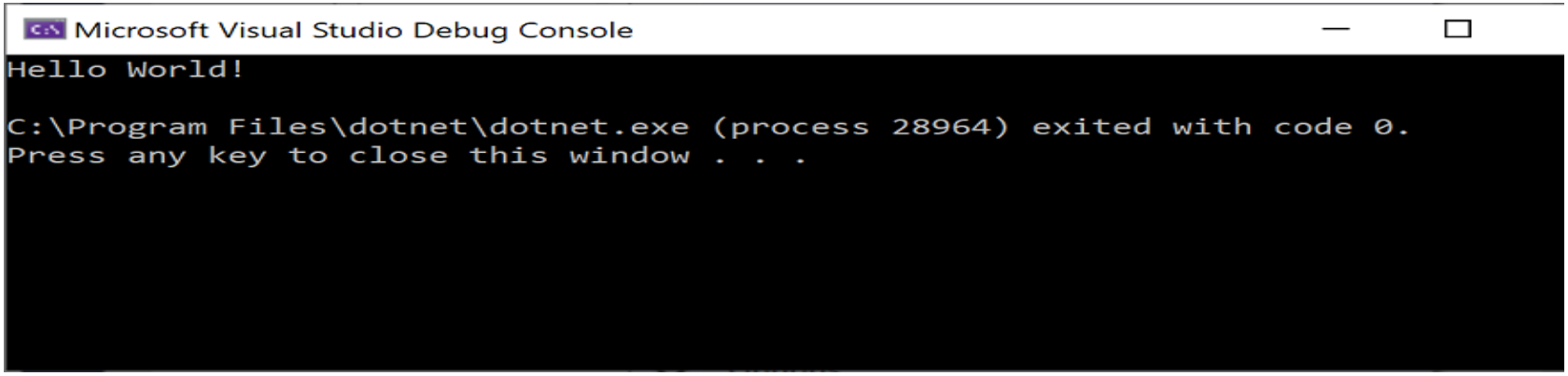

Create a new project
.net core console
Language
Platform
Project type
Recent project templates
A list of your recently accessed templates will be displayed here.
Filtering by: C#, Windows
Clear filter
Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C#
Linux
macOS
Windows
Console
Not finding what you're looking for?
Install more tools and features
Back
Next

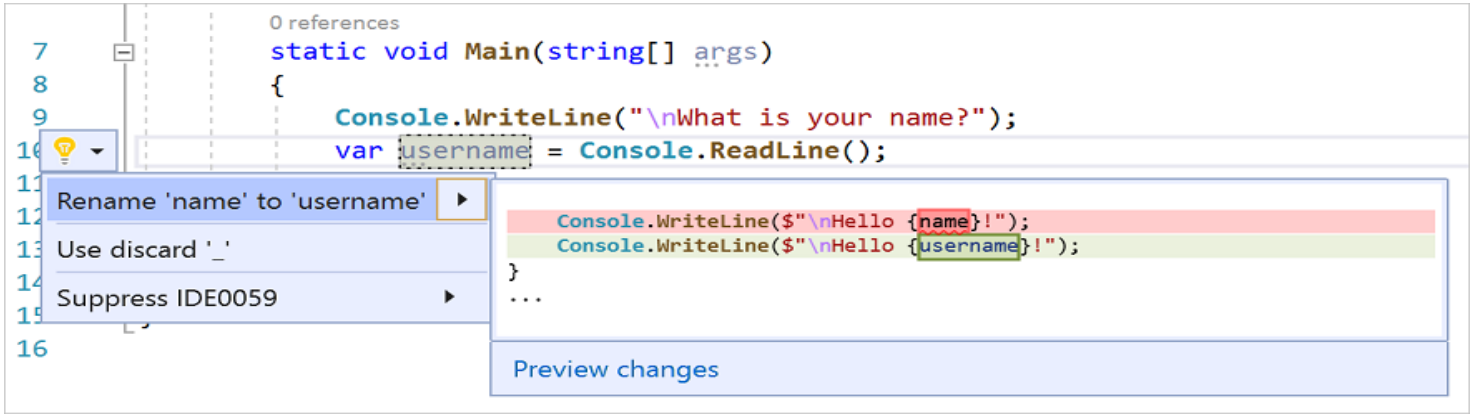

Configure your new project
Console App (.NET Core)
C#
Linux
macOS
Windows
Console
Project name
HelloWorld
Location
C:\VSProjects
...
Solution name
HelloWorld
Place solution and project in the same directory
Back
Create

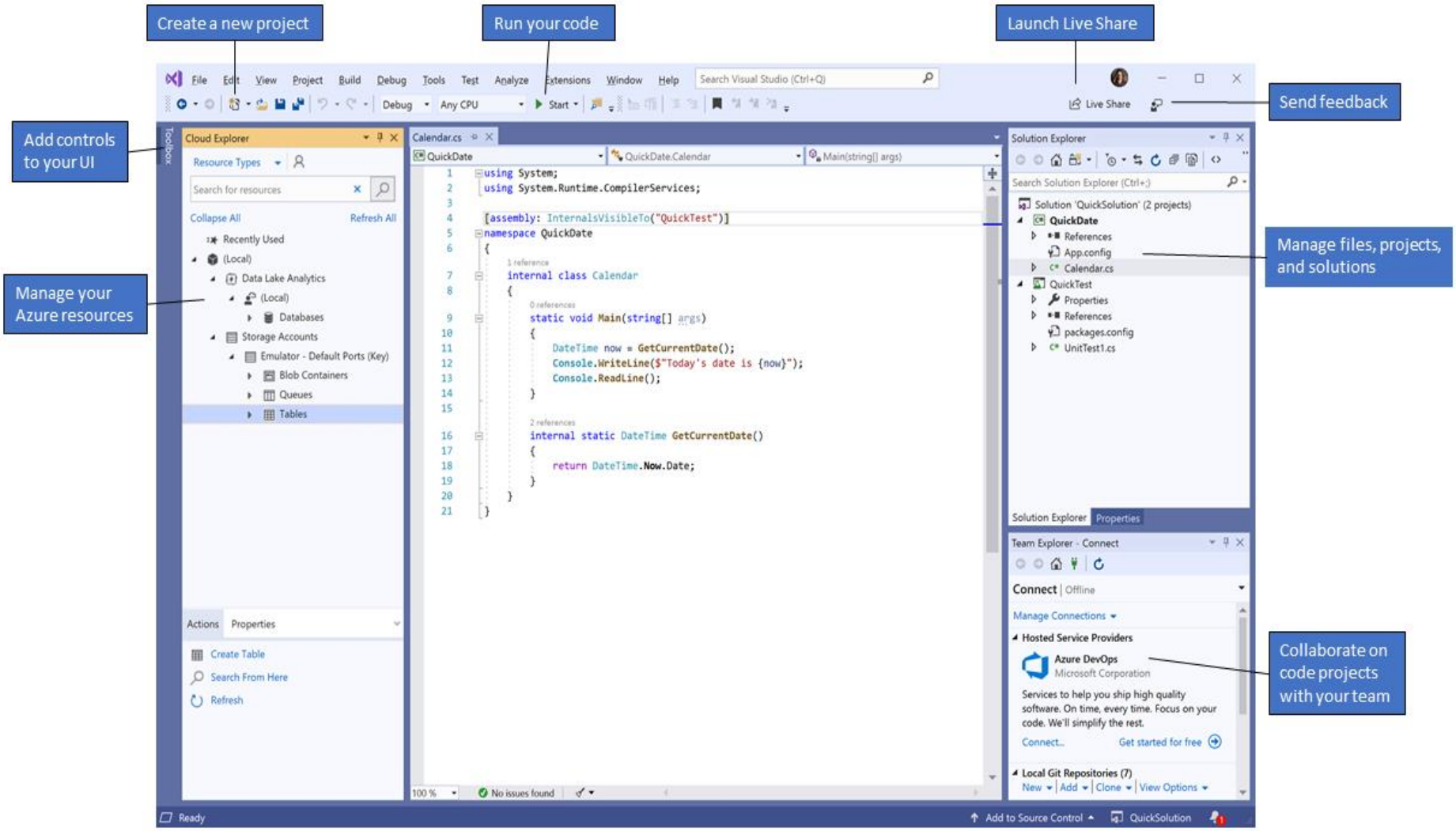

File Edit View Project Build Debug Tools Test Analyze Extensions Window Help
Search Visual Studio (Ctrl+Q)
Sign in
Debug
Any CPU
HelloWorld
Live Share
Toolbox
Program.cs
HelloWorld
HelloWorld.Program
Main(string[] args)
using System;
namespace HelloWorld
{
0 references
class Program
{
0 references
static void Main(string[] args)
{
Console.WriteLine("Hello World!");
}
}
}
100 %
No issues found
Error List
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'HelloWorld' (1 project)
HelloWorld
Dependencies
Program.cs
Solution Explorer
Properties
Team Explorer - Connect
Connect | Offline
Manage Connections
Hosted Service Providers
Azure DevOps
Microsoft Corporation
Services to help you ship high quality
Ready
Ln 1
Col 1
Ch 1
INS
Add to Source Control
HelloWorld


File Edit View Project Build Debug Tools Test Analyze Extensions Window Help
Search Visual Studio (Ctrl+Q)
Sign in
Debug
Any CPU
HelloWorld
Live Share
Program.cs
HelloWorld
HelloWorld.Program
Main(string[] args)
using System;
namespace HelloWorld
{
class Program
{
static void Main(string[] args)
{
Console.WriteLine("Hello World!");
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'HelloWorld' (1 project)
HelloWorld
Dependencies
Program.cs
Solution Explorer
Properties
Team Explorer - Connect
Connect | Offline
Manage Connections
Hosted Service Providers
Azure DevOps
Microsoft Corporation
Services to help you ship high quality
100 %
No issues found
Error List
Ready
Ln 1
Col 1
Ch 1
INS
Add to Source Control
HelloWorld

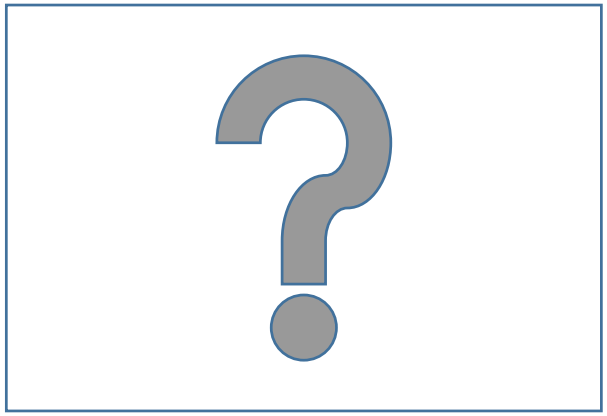

Debug
Team
Tools
Test
Analyze
Window
Windows
Start Debugging F5
Start Without Debugging Ctrl+F5
Performance Profiler... Alt+F2
Attach to Process... Ctrl+Alt+P
Other Debug Targets
Profiler


Microsoft Visual Studio Debug Console
Hello World!
C:\Program Files\dotnet\dotnet.exe (process 28964) exited with code 0.
Press any key to close this window . . .


0 references
static void Main(string[] args)
{
Console.WriteLine("\nWhat is your name?");
var username = Console.ReadLine();
Rename 'name' to 'username'
Use discard '_'
Suppress IDE0059
Console.WriteLine($"\nHello {name}!");
Console.WriteLine($"\nHello {username}!");
}
...
Preview changes


Create a new project
Run your code
Launch Live Share
Send feedback
Add controls to your UI
Manage your Azure resources
Manage files, projects, and solutions
Collaborate on code projects with your team
File Edit View Project Build Debug Tools Test Analyze Extensions Window Help
Search Visual Studio (Ctrl+Q)
Debug
Any CPU
Start
Live Share
Cloud Explorer
Resource Types
Search for resources
Collapse All
Refresh All
Recently Used
(Local)
Data Lake Analytics
(Local)
Databases
Storage Accounts
Emulator - Default Ports (Key)
Blob Containers
Queues
Tables
Actions
Properties
Create Table
Search From Here
Refresh
Calendar.cs
QuickDate
QuickDate.Calendar
Main(string[] args)
using System;
using System.Runtime.CompilerServices;
[assembly: InternalsVisibleTo("QuickTest")]
namespace QuickDate
{
1 reference
internal class Calendar
{
0 references
static void Main(string[] args)
{
DateTime now = GetCurrentDate();
Console.WriteLine($"Today's date is {now}");
Console.ReadLine();
}
2 references
internal static DateTime GetCurrentDate()
{
return DateTime.Now.Date;
}
}
}
100 %
No issues found
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'QuickSolution' (2 projects)
QuickDate
References
App.config
Calendar.cs
QuickTest
Properties
References
packages.config
UnitTest1.cs
Solution Explorer
Properties
Team Explorer - Connect
Connect | Offline
Manage Connections
Hosted Service Providers
Azure DevOps
Microsoft Corporation
Services to help you ship high quality software. On time, every time. Focus on your code. We'll simplify the rest.
Connect...
Get started for free
Local Git Repositories (7)
New | Add | Clone | View Options
Ready
Add to Source Control
QuickSolution

# QUESTION



**Google Classroom :**