

DSG Support Multi Solution

.NET Framework & Architecture

.NET फ्रेमवर्क माइक्रोसॉफ्ट द्वारा विकसित एक सॉफ्टवेयर डेवलपमेंट फ्रेमवर्क है जो विंडोज ऑपरेटिंग सिस्टम पर एप्लिकेशन बनाने और चलाने के लिए रनटाइम वातावरण और लाइब्रेरी और टूल का एक सेट प्रदान करता है। .NET फ्रेमवर्क मुख्य रूप से विंडोज पर उपयोग किया जाता है, जबकि .NET कोर (जो संस्करण 5 से शुरू होकर केवल .NET में विकसित हुआ) क्रॉस-प्लेटफॉर्म है। यह फ्रेमवर्क कई प्रोग्रामिंग भाषाओं, जैसे C#, F#, और VB.NET का समर्थन करता है, और डेस्कटॉप, वेब, मोबाइल, क्लाउड और गेमिंग एप्लिकेशन सहित कई प्रकार के एप्लिकेशन का समर्थन करता है।

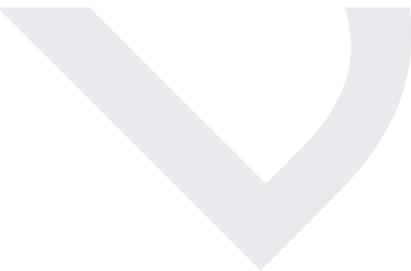
NET फ्रेमवर्क के मुख्य घटक

• सामान्य भाषा रनटाइम (सीएलआर) :

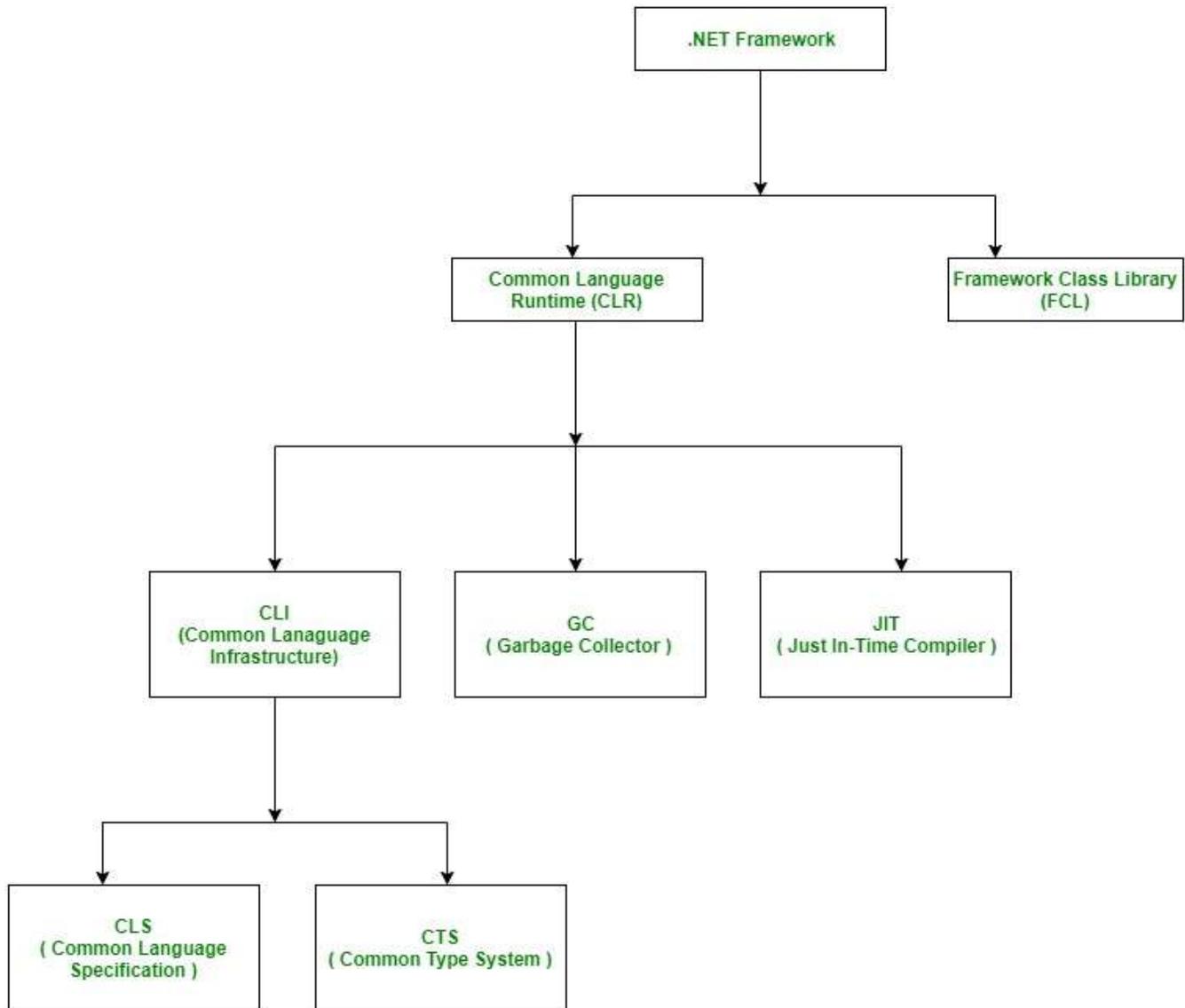
- CLR .NET फ्रेमवर्क का दिल है, जो एक वर्चुअल मशीन के रूप में कार्य करता है जो कोड चलाता है और मेमोरी प्रबंधन, सुरक्षा और थ्रेड प्रबंधन जैसी विभिन्न सेवाओं का प्रबंधन करता है। CLR के भीतर संकलित और निष्पादित किए जाने वाले कोड को "प्रबंधित कोड" कहा जाता है, जबकि CLR द्वारा प्रबंधित नहीं किए जाने वाले कोड को "अप्रबंधित कोड" के रूप में जाना जाता है।

• .NET फ्रेमवर्क क्लास लाइब्रेरी (FCL):

- एफसीएल अनुप्रयोग विकास के लिए पुनः प्रयोज्य कक्षाओं और विधियों का एक बड़ा सेट प्रदान करता है।
- इसमें इनपुट/आउटपुट संचालन, नेटवर्किंग, डेटा एक्सेस, यूआई नियंत्रण आदि के लिए लाइब्रेरी शामिल हैं।



DSG Support Multi Solution



कुल मिलाकर, .NET फ्रेमवर्क एक शक्तिशाली और बहुमुखी विकास मंच है जो विंडोज ऑपरेटिंग सिस्टम पर अनुप्रयोगों के निर्माण और चलाने के लिए उपकरणों और लाइब्रेरीज़ की एक विस्तृत श्रृंखला प्रदान करता है।

- **.NET** एक सॉफ्टवेयर फ्रेमवर्क है जिसे Microsoft द्वारा डिज़ाइन और विकसित किया गया है। .Net फ्रेमवर्क का पहला संस्करण 1.0 था जो वर्ष 2002 में आया था। आसान शब्दों में, यह **C#**, VB.Net आदि जैसी विभिन्न भाषाओं में लिखे गए प्रोग्रामों को संकलित और निष्पादित करने के लिए एक वर्चुअल मशीन है।
- इसका उपयोग फॉर्म-आधारित एप्लिकेशन, वेब-आधारित एप्लिकेशन और वेब सेवाओं को विकसित करने के लिए किया जाता है। .Net प्लेटफॉर्म पर कई तरह की प्रोग्रामिंग भाषाएँ उपलब्ध हैं, जिनमें VB.Net और **C#** सबसे आम हैं। इसका उपयोग विंडोज़, फ़ोन, वेब आदि के लिए एप्लिकेशन बनाने के लिए किया जाता है। यह बहुत सारी कार्यक्षमताएँ प्रदान करता है और उद्योग मानकों का भी समर्थन करता है।

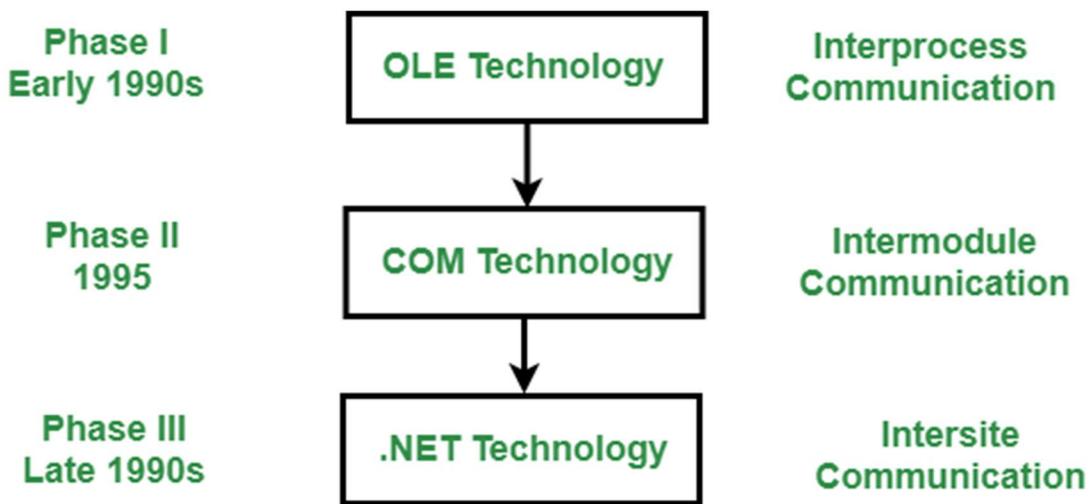
DSG Support Multi Solution

- .NET फ्रेमवर्क 60 से ज़्यादा प्रोग्रामिंग भाषाओं को सपोर्ट करता है, जिनमें से 11 प्रोग्रामिंग भाषाओं को Microsoft द्वारा डिज़ाइन और विकसित किया गया है। बाकी **गैर-Microsoft भाषाएँ** .NET फ्रेमवर्क द्वारा समर्थित हैं, लेकिन Microsoft द्वारा डिज़ाइन और विकसित नहीं की गई हैं।

.NET प्रौद्योगिकी का विकास

.NET प्रौद्योगिकी के विकास के तीन महत्वपूर्ण चरण हैं।

1. ओएलई प्रौद्योगिकी
2. COM प्रौद्योगिकी
3. .NET प्रौद्योगिकी



- OLE तकनीक:** OLE (ऑब्जेक्ट लिंकिंग और एम्बेडिंग) माइक्रोसॉफ्ट के घटक दस्तावेज़ की तकनीकों में से एक है। मूल रूप से, इसका मुख्य उद्देश्य विभिन्न अनुप्रयोगों के तत्वों को एक दूसरे से जोड़ना है।
- COM तकनीक:** ऑपरेटिंग सिस्टम के Microsoft Windows परिवार की तकनीक, Microsoft COM (कॉमन ऑब्जेक्ट मॉडल) विभिन्न सॉफ्टवेयर घटकों को संचार करने में सक्षम बनाती है। COM का उपयोग अधिकांशतः डेवलपर्स द्वारा विभिन्न उद्देश्यों के लिए किया जाता है जैसे कि पुनः प्रयोज्य सॉफ्टवेयर घटक बनाना, अनुप्रयोगों के निर्माण के लिए घटकों को एक साथ जोड़ना, और Windows सेवाओं का लाभ उठाना। COM के ऑब्जेक्ट्स को प्रोग्रामिंग भाषाओं की एक विस्तृत श्रृंखला के साथ बनाया जा सकता है।
- .NET तकनीक:** विंडोज़ और वेब एप्लिकेशन विकसित करने के लिए तकनीकों का संग्रह या सेट। .Net की तकनीक Microsoft द्वारा विकसित की गई है और इसे फरवरी 2002 में लॉन्च किया गया था, मूल परिभाषा के अनुसार, Microsoft की नई इंटरनेट रणनीति। इसे मूल रूप से NGWS (नेक्स्ट जेनरेशन वेब सर्विसेज) कहा जाता था। इसे आज उपलब्ध सबसे शक्तिशाली, लोकप्रिय और बहुत उपयोगी इंटरनेट तकनीक में से एक माना जाता है।

DSG Support Multi Solution

.NET प्रोग्रामिंग भाषाएँ

.NET पारिस्थितिकी तंत्र Microsoft और तृतीय-पक्ष योगदानकर्ताओं द्वारा विकसित विभिन्न प्रोग्रामिंग भाषाओं का समर्थन करता है। सबसे व्यापक रूप से उपयोग की जाने वाली कुछ भाषाएँ हैं:

- **C# .NET:** एक आधुनिक, ऑब्जेक्ट-ओरिएंटेड भाषा जिसका उपयोग विभिन्न प्रकार के अनुप्रयोगों के लिए किया जाता है।
- **VB.NET:** उपयोग में आसानी के लिए डिज़ाइन की गई एक भाषा, जिसका वाक्यविन्यास पारंपरिक बेसिक के समान है।
- **F# .NET :** एक कार्यात्मक-प्रथम प्रोग्रामिंग भाषा, जो जटिल एल्गोरिदम और डेटा प्रोसेसिंग के लिए उपयुक्त है।
- **C++ .NET:** प्रबंधित कोड अनुप्रयोगों के लिए डिज़ाइन किया गया C++ का एक विस्तार।
- **J# .NET :** एक भाषा जो जावा डेवलपर्स के लिए .NET संगतता प्रदान करती है।
- **आयरनरूबी, आयरनपायथन :** रूबी और पायथन भाषाओं के .NET कार्यान्वयन।
- **अन्य भाषाएँ :** JScript.NET, C Omega, ASML, और अधिक।

क्या .NET अनुप्रयोग प्लेटफॉर्म पर निर्भर है या प्लेटफॉर्म से स्वतंत्र है?

डिफ़ॉल्ट रूप से, .NET एप्लिकेशन को प्लेटफॉर्म-निर्भर होने के लिए डिज़ाइन किया गया था, जो मुख्य रूप से विंडोज़ पर चलता था। लेकिन मोनो (एक क्रॉस-प्लेटफॉर्म फ्रेमवर्क) और .NET कोर की वजह से, डेवलपर्स अब लिनक्स, मैकओएस और यहां तक कि मोबाइल प्लेटफॉर्म पर भी .NET एप्लिकेशन चला सकते हैं।

मोनो फ्रेमवर्क :

- **नोवेल (अब माइक्रो फोकस का हिस्सा)** द्वारा विकसित मोनो फ्रेमवर्क, .NET अनुप्रयोगों को विभिन्न ऑपरेटिंग सिस्टम पर चलाने की अनुमति देता है।
- यद्यपि यह क्रॉस-प्लेटफॉर्म है, मोनो एक सशुल्क फ्रेमवर्क है।

.NET संस्करण

.NET फ्रेमवर्क का रिलीज इतिहास और विभिन्न विंडोज संस्करणों के साथ इसकी संगतता नीचे दी गई है:

.NET संस्करण	सीएलआर संस्करण	विकास उपकरण	विंडोज़ समर्थन
1.0	1.0	विजुअल स्टूडियो .NET	एक्सपी एसपी1
1.1	1.1	विजुअल स्टूडियो .NET 2003	एक्सपी एसपी2, एसपी3

DSG Support Multi Solution

.NET संस्करण	सीएलआर संस्करण	विकास उपकरण	विंडोज समर्थन
2.0	2.0	विजुअल स्टूडियो 2005	एन/ए
3.0	2.0	अभिव्यक्ति मिश्रण	विस्टा
3.5	2.0	विजुअल स्टूडियो 2008	7, 8, 8.1, 10
4.0	4	विजुअल स्टूडियो 2010	एन/ए
4.5	4	विजुअल स्टूडियो 2012	एन/ए
4.5.1	4	विजुअल स्टूडियो 2013	एन/ए
4.5.2	4	एन/ए	एन/ए
4.6	4	विजुअल स्टूडियो 2015	एन/ए
4.6.1	4	विजुअल स्टूडियो 2015 अद्यतन 1	एन/ए
4.6.2	4	एन/ए	10 वी1607
4.7	4	विजुअल स्टूडियो 2017	10 वी1703
4.7.1	4	विजुअल स्टूडियो 2017	10 वी1709
4.7.2	4	विजुअल स्टूडियो 2017	10वी 1803
4.8	4	विजुअल स्टूडियो 2019	11

DSG Support Multi Solution

.NET संस्करण	सीएलआर संस्करण	विकास उपकरण	विंडोज समर्थन
4.8.1	4	विजुअल स्टूडियो 2019	10वीं 20एच2
6	-	विजुअल स्टूडियो 2022	11
8.0 (2024)	-	विजुअल स्टूडियो 2024	10
9.0 (2025)	-	विजुअल स्टूडियो 2025	11

.NET फ्रेमवर्क के लाभ

- बहु-भाषा समर्थन:** .NET फ्रेमवर्क कई प्रोग्रामिंग भाषाओं का समर्थन करता है, जिनमें C#, F# और विजुअल बेसिक शामिल हैं, जिससे डेवलपर्स को अपनी आवश्यकताओं के अनुरूप सर्वोत्तम भाषा चुनने की सुविधा मिलती है, जबकि वे लाइब्रेरी और टूल के समान सेट से लाभ उठा सकते हैं।
- क्रॉस-प्लेटफॉर्म संगतता (.NET कोर और मोनो के माध्यम से):** हालांकि शुरुआत में विंडोज के लिए डिज़ाइन किया गया था, .NET फ्रेमवर्क अब .NET कोर और मोनो के माध्यम से विभिन्न ऑपरेटिंग सिस्टम पर चल सकता है, जिससे यह क्रॉस-प्लेटफॉर्म विकास के लिए एक बहुमुखी समाधान बन जाता है।
- बड़ा और सक्रिय डेवलपर समुदाय:** .NET फ्रेमवर्क को व्यापक डेवलपर आधार से लाभ मिलता है, जो लाइब्रेरी, टूल और संसाधनों का एक समृद्ध पारिस्थितिकी तंत्र बनाता है। इससे डेवलपर्स को समस्याओं को अधिक आसानी से हल करने और समुदाय में समाधान साझा करने में मदद मिलती है।
- सुरक्षा विशेषताएं:** इस फ्रेमवर्क में विभिन्न सुरक्षा विशेषताएं शामिल हैं, जैसे कोड एक्सेस सुरक्षा और डिजिटल हस्ताक्षर, जो अनुप्रयोगों को दुर्भावनापूर्ण खतरों और कमजोरियों से बचाने में मदद करते हैं।
- उत्पादकता:** .NET द्वारा प्रस्तुत पूर्व-निर्मित लाइब्रेरीज़ और टूल का सेट विकास समय को तीव्र कर देता है, जिससे डेवलपर्स को सामान्य कार्यात्मकताओं को पुनः आविष्कृत करने के बजाय व्यावसायिक तर्क पर ध्यान केंद्रित करने में सहायता मिलती है।

.NET फ्रेमवर्क के नुकसान

- विंडोज निर्भरता:** यद्यपि यह क्रॉस-प्लेटफॉर्म पर समर्थन करता है, .NET फ्रेमवर्क मूल रूप से विंडोज के लिए बनाया गया था, और कुछ विशेषताएं अभी भी मुख्य रूप से विंडोज ऑपरेटिंग सिस्टम के लिए अनुकूलित हो सकती हैं।

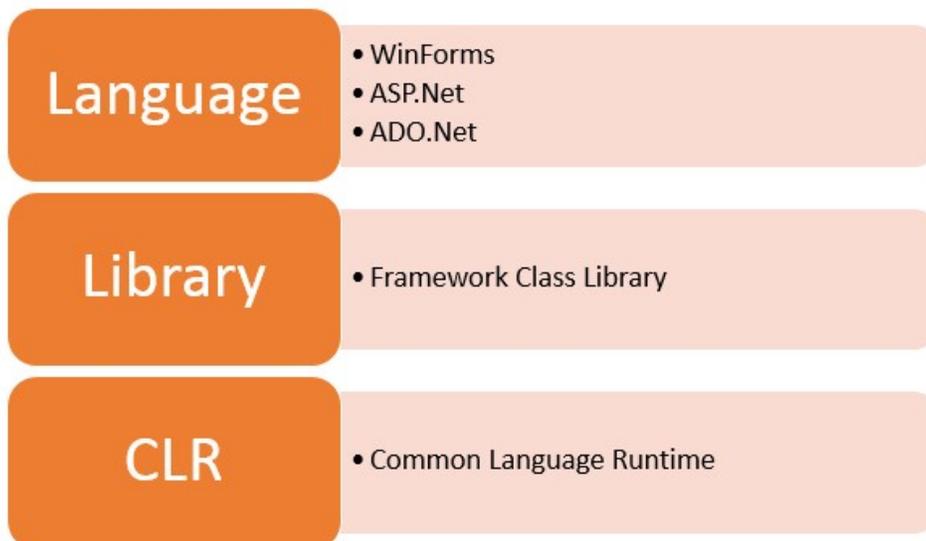
DSG Support Multi Solution

- **बड़ा फुटप्रिंट** : .NET फ्रेमवर्क की स्थापना अपेक्षाकृत बड़ी है, जो सीमित भंडारण या बैंडविड्थ वाले उपकरणों पर चलने वाले अनुप्रयोगों के लिए चिंता का विषय हो सकता है।
- **लाइसेंसिंग लागत**: .NET फ्रेमवर्क के कुछ संस्करणों के लिए सशुल्क लाइसेंस की आवश्यकता हो सकती है, जिससे विकास और परिनियोजन की लागत बढ़ जाती है।
- **प्रदर्शन संबंधी विचार** : यद्यपि .NET अधिकांश अनुप्रयोगों के लिए उत्कृष्ट प्रदर्शन प्रदान करता है, यह उच्च-प्रदर्शन परिदृश्यों के लिए आदर्श विकल्प नहीं हो सकता है जहां निम्न-स्तरीय हार्डवेयर इंटरैक्शन या जटिल एल्गोरिदम की आवश्यकता होती है।
- **सीखने की प्रक्रिया** : यद्यपि .NET को उपयोग में आसान बनाने के लिए डिज़ाइन किया गया है, लेकिन इस प्लेटफॉर्म पर नए डेवलपर्स को विशेष रूप से ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग के साथ सीखने की प्रक्रिया का सामना करना पड़ सकता है।

.नेट फ्रेमवर्क आर्किटेक्चर

.नेट फ्रेमवर्क आर्किटेक्चर .नेट प्लेटफॉर्म के लिए एक प्रोग्रामिंग मॉडल है जो विभिन्न विंडोज और डेस्कटॉप अनुप्रयोगों के सरल विकास और परिनियोजन के लिए विभिन्न प्रोग्रामिंग भाषाओं के साथ एक निष्पादन वातावरण और एकीकरण प्रदान करता है। इसमें क्लास लाइब्रेरी और पुनः प्रयोज्य घटक शामिल हैं।

.Net फ्रेमवर्क की मूल संरचना नीचे दर्शाई गई है।



.नेट फ्रेमवर्क आर्किटेक्चर आरेख

DSG Support Multi Solution

.NET घटक

.Net फ्रेमवर्क की वास्तुकला निम्नलिखित प्रमुख घटकों पर आधारित है;

1. सामान्य भाषा रनटाइम

"कॉमन लैंग्वेज इन्फ्रास्ट्रक्चर" या सीएलआई .NET आर्किटेक्चर में एक प्लेटफॉर्म है जिस पर .NET प्रोग्राम निष्पादित होते हैं।

सीएलआई की निम्नलिखित प्रमुख विशेषताएं हैं:

अपवाद प्रबंधन - अपवाद वे त्रुटियाँ हैं जो अनुप्रयोग के निष्पादन के समय उत्पन्न होती हैं।

अपवादों के उदाहरण हैं:

यदि कोई एप्लिकेशन स्थानीय मशीन पर फ़ाइल खोलने का प्रयास करता है, लेकिन फ़ाइल मौजूद नहीं है।

यदि अनुप्रयोग किसी [डेटाबेस](#) से कुछ रिकॉर्ड लाने का प्रयास करता है , लेकिन डेटाबेस से कनेक्शन वैध नहीं है।

कचरा संग्रहण - कचरा संग्रहण अवांछित संसाधनों को हटाने की प्रक्रिया है, जब उनकी आवश्यकता नहीं रह जाती।

कचरा संग्रहण के उदाहरण हैं

एक फ़ाइल हैंडल जिसकी अब आवश्यकता नहीं है। यदि एप्लिकेशन ने किसी फ़ाइल पर सभी ऑपरेशन समाप्त कर दिए हैं, तो फ़ाइल हैंडल की अब आवश्यकता नहीं रह सकती है।

डेटाबेस कनेक्शन की अब आवश्यकता नहीं है। यदि एप्लिकेशन ने डेटाबेस पर सभी ऑपरेशन समाप्त कर दिए हैं, तो डेटाबेस कनेक्शन की अब आवश्यकता नहीं हो सकती है।

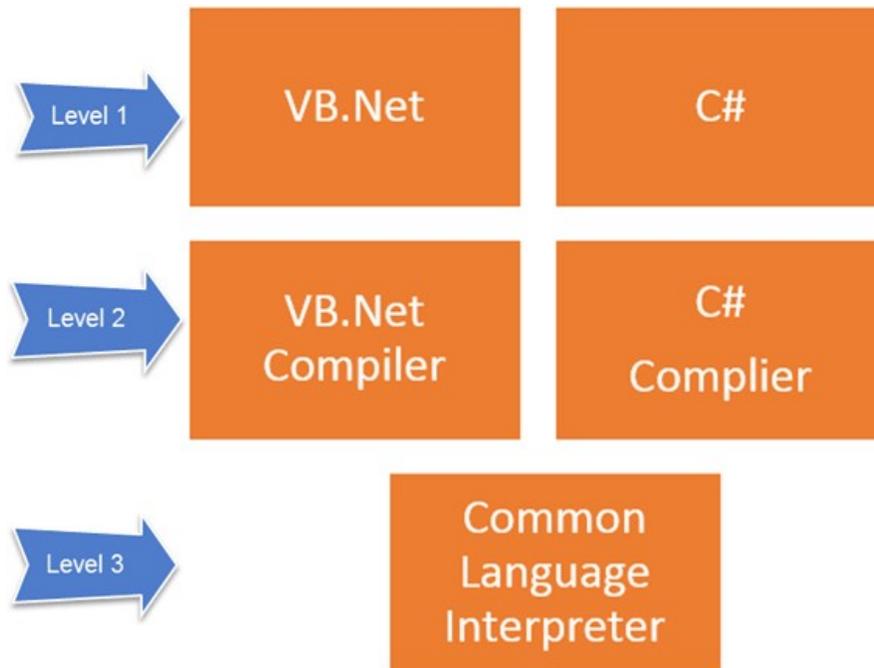
विभिन्न प्रोग्रामिंग भाषाओं के साथ कार्य करना -

जैसा कि पहले अनुभाग में बताया गया है, एक डेवलपर विभिन्न .Net प्रोग्रामिंग भाषाओं में एप्लिकेशन विकसित कर सकता है।

भाषा - पहला स्तर प्रोग्रामिंग भाषा ही है, सबसे आम VB.Net और C# हैं।

DSG Support Multi Solution

2. **कंपाइलर** - प्रत्येक प्रोग्रामिंग भाषा के लिए एक अलग कंपाइलर होगा। इसलिए VB.Net भाषा के अंतर्गत, एक अलग VB.Net कंपाइलर होगा। इसी तरह, C# के लिए, आपके पास एक और कंपाइलर होगा।
3. **कॉमन लैंग्वेज इंटरप्रेटर** - यह .Net में अंतिम परत है जिसका उपयोग किसी भी प्रोग्रामिंग भाषा में विकसित .net प्रोग्राम को चलाने के लिए किया जाएगा। इसलिए बाद का कंपाइलर .Net एप्लिकेशन को चलाने के लिए प्रोग्राम को CLI लेयर पर भेजेगा।



2. Class Library

.NET फ्रेमवर्क में मानक क्लास लाइब्रेरी का एक सेट शामिल है। क्लास लाइब्रेरी विधियों और फंक्शन का एक संग्रह है जिसका उपयोग मुख्य उद्देश्य के लिए किया जा सकता है।

उदाहरण के लिए, सभी फ़ाइल-स्तरीय ऑपरेशन को संभालने के लिए विधियों के साथ एक क्लास लाइब्रेरी है। तो एक विधि है जिसका उपयोग फ़ाइल से टेक्स्ट पढ़ने के लिए किया जा सकता है। इसी तरह, फ़ाइल में टेक्स्ट लिखने के लिए भी एक विधि है।

अधिकांश विधियाँ या तो System.* या Microsoft.* नामस्थानों में विभाजित हैं। (तारांकन चिह्न * का अर्थ है कि सिस्टम या Microsoft नामस्थान के अंतर्गत आने वाली सभी विधियों का संदर्भ)

DSG Support Multi Solution

नामस्थान विधियों का तार्किक पृथक्करण है। हम इन नामस्थानों के बारे में अगले अध्यायों में और विस्तार से जानेंगे।

3. भाषाएँ

.Net फ्रेमवर्क में बनाए जा सकने वाले अनुप्रयोगों के प्रकारों को मोटे तौर पर निम्नलिखित श्रेणियों में वर्गीकृत किया गया है।

WinForms - इसका उपयोग फॉर्म-आधारित एप्लिकेशन विकसित करने के लिए किया जाता है, जो अंतिम उपयोगकर्ता मशीन पर चलेगा। नोटपैड क्लाइंट-आधारित एप्लिकेशन का एक उदाहरण है।

ASP.Net - इसका उपयोग वेब-आधारित अनुप्रयोगों को विकसित करने के लिए किया जाता है, जिन्हें किसी भी ब्राउज़र जैसे इंटरनेट एक्सप्लोरर, क्रोम या फ़ायरफ़ॉक्स पर चलाने के लिए बनाया जाता है।

वेब अनुप्रयोग को एक सर्वर पर संसाधित किया जाएगा, जिसमें इंटरनेट सूचना सेवाएं स्थापित होंगी।

इंटरनेट सूचना सेवा या IIS एक माइक्रोसॉफ्ट घटक है जिसका उपयोग [Asp.Net](#) अनुप्रयोग को निष्पादित करने के लिए किया जाता है।

इसके बाद निष्पादन का परिणाम क्लाइंट मशीनों को भेजा जाता है, और आउटपुट ब्राउज़र में दिखाया जाता है।

ADO.Net - इस तकनीक का उपयोग **Oracle** या **Microsoft SQL** Server जैसे डेटाबेस के साथ इंटरैक्ट करने के लिए अनुप्रयोगों को विकसित करने के लिए किया जाता है ।

माइक्रोसॉफ्ट हमेशा यह सुनिश्चित करता है कि .Net फ्रेमवर्क सभी समर्थित विंडोज ऑपरेटिंग सिस्टम के अनुरूप हो।

.नेट फ्रेमवर्क डिज़ाइन सिद्धांत

अब इस .Net आर्किटेक्चर ट्यूटोरियल में, हम .Net फ्रेमवर्क के डिज़ाइन सिद्धांतों को सीखेंगे। .Net फ्रेमवर्क के निम्नलिखित डिज़ाइन सिद्धांत इसे .Net आधारित एप्लिकेशन बनाने के लिए बहुत प्रासंगिक बनाते हैं।

1) इंटरऑपरेबिलिटी - .नेट फ्रेमवर्क बहुत सारे बैकवर्ड सपोर्ट प्रदान करता है। मान लीजिए कि आपके पास [.नेट फ्रेमवर्क के पुराने संस्करण](#), जैसे 2.0 पर बना कोई एप्लिकेशन है। और यदि आप उसी एप्लिकेशन को किसी ऐसी मशीन पर चलाने की कोशिश करते हैं जिसमें .नेट फ्रेमवर्क का उच्च संस्करण है, जैसे 3.5। तो एप्लिकेशन फिर भी काम करेगा। ऐसा इसलिए है क्योंकि हर रिलीज़ के साथ, Microsoft यह सुनिश्चित करता है कि पुराने फ्रेमवर्क संस्करण नवीनतम संस्करण के साथ अच्छी तरह से मेल खाते हों।

2) पोर्टेबिलिटी - .नेट फ्रेमवर्क पर बनाए गए एप्लिकेशन को किसी भी विंडोज प्लेटफॉर्म पर काम करने के लिए बनाया जा सकता है। और अब हाल के दिनों में, माइक्रोसॉफ्ट भी माइक्रोसॉफ्ट उत्पादों को अन्य प्लेटफॉर्म, जैसे कि आईओएस और [लिनक्स](#) पर काम करने के लिए तैयार कर रहा है।

3) सुरक्षा - .NET फ्रेमवर्क में एक अच्छा सुरक्षा तंत्र है। इनबिल्ट सुरक्षा तंत्र एप्लीकेशन के सत्यापन और सत्यापन दोनों में मदद करता है। प्रत्येक एप्लीकेशन अपने सुरक्षा तंत्र को स्पष्ट रूप से परिभाषित कर सकता है। प्रत्येक सुरक्षा तंत्र का उपयोग उपयोगकर्ता को कोड या चल रहे प्रोग्राम तक पहुँच प्रदान करने के लिए किया जाता है।

4) मेमोरी मैनेजमेंट - कॉमन लैंग्वेज रनटाइम [मेमोरी मैनेजमेंट का](#) सारा काम करता है। .नेट फ्रेमवर्क में उन संसाधनों को देखने की पूरी क्षमता है, जिनका उपयोग किसी चल रहे प्रोग्राम द्वारा नहीं किया जाता है। फिर यह उन संसाधनों को तदनुसार जारी करेगा। यह "गारबेज कलेक्टर" नामक एक प्रोग्राम के माध्यम से किया जाता है जो .नेट फ्रेमवर्क के हिस्से के रूप में

DSG Support Multi Solution

चलता है। गारबेज कलेक्टर नियमित अंतराल पर चलता रहता है और जाँचता रहता है कि कौन से सिस्टम संसाधन उपयोग में नहीं हैं, और उन्हें तदनुसार मुक्त करता है।

5) सरलीकृत परिनियोजन - .Net फ्रेमवर्क में ऐसे उपकरण भी होते हैं, जिनका उपयोग .Net फ्रेमवर्क पर निर्मित अनुप्रयोगों को पैकेज करने के लिए किया जा सकता है। इन पैकेजों को फिर क्लाइट मशीनों में वितरित किया जा सकता है। फिर पैकेज स्वचालित रूप से अनुप्रयोग स्थापित कर देंगे।

CLR

कॉमन लैंग्वेज रनटाइम (CLR) Microsoft .NET फ्रेमवर्क का एक घटक है जो .NET अनुप्रयोगों के निष्पादन का प्रबंधन करता है। यह C#, VB.NET, F# और अन्य सहित विभिन्न .NET प्रोग्रामिंग भाषाओं में लिखे गए कोड को लोड करने और निष्पादित करने के लिए जिम्मेदार है।

जब **C#** प्रोग्राम संकलित किया जाता है, तो परिणामी निष्पादन योग्य कोड एक मध्यवर्ती भाषा में होता है जिसे कॉमन इंटरमीडिएट लैंग्वेज (CIL) या Microsoft इंटरमीडिएट लैंग्वेज (MSIL) कहा जाता है। यह कोड मशीन-विशिष्ट नहीं है, और यह किसी भी प्लेटफॉर्म पर चल सकता है जिसमें CLR स्थापित है। जब CIL कोड निष्पादित होता है, तो CLR इसे मशीन कोड में संकलित करता है जिसे प्रोसेसर द्वारा निष्पादित किया जा सकता है।

सी.एल.आर. की कार्यप्रणाली

1. संकलन और निष्पादन:

- जब हम C# प्रोग्राम लिखते हैं, तो वह इंटरमीडिएट लैंग्वेज (IL) कोड में संकलित हो जाता है, जो प्लेटफॉर्म-स्वतंत्र होता है।
- इसके बाद CLR, प्रोग्राम के चलने के दौरान IL कोड को मशीन-विशिष्ट कोड में परिवर्तित करने के लिए जस्ट-इन-टाइम (JIT) कंपाइलर का उपयोग करता है।

2. सीएलआर द्वारा प्रदान की गई सेवाएं:

- सी.एल.आर. गारबेज कलेक्शन के माध्यम से स्वचालित मेमोरी प्रबंधन को संभालता है, तथा मेमोरी लीक को रोकता है।
- यह सुनिश्चित करता है कि डेटा प्रकारों का सही और सुरक्षित उपयोग किया जाए।
- सी.एल.आर., आईएल कोड को चलाने से पहले सुरक्षा जोखिमों के लिए उसकी जांच करता है।

3. अंतर-भाषा एकीकरण:

DSG Support Multi Solution

सीएलआर विभिन्न .NET भाषाओं (C#, VB.NET, F#) के कोड को कॉमन टाइप सिस्टम (CTS) के माध्यम से एक साथ सहजता से काम करने की अनुमति देता है।

सी.एल.आर. के प्रमुख घटक

जैसा कि शब्द से स्पष्ट है, कॉमन का अर्थ है कि CLR एक सामान्य रनटाइम या निष्पादन वातावरण प्रदान करता है क्योंकि 60 से अधिक .NET प्रोग्रामिंग भाषाएं हैं।

- **सामान्य भाषा विनिर्देश (सीएलएस)** : यह सामान्य नियमों को परिभाषित करता है ताकि विभिन्न भाषाओं में लिखा गया कोड परस्पर संचालित हो सके।
 - प्रबंधित कोड: MSIL कोड जिसे CLR द्वारा प्रबंधित किया जाता है उसे प्रबंधित कोड के रूप में जाना जाता है। प्रबंधित कोड के लिए CLR तीन .NET सुविधाएँ प्रदान करता है:
 - अप्रबंधित कोड: .NET विकास से पहले, .COM घटक और Win32 API जैसी प्रोग्रामिंग भाषाएँ MSIL कोड उत्पन्न नहीं करती थीं। इसलिए इन्हें CLR द्वारा प्रबंधित नहीं किया जाता बल्कि ऑपरेटिंग सिस्टम द्वारा प्रबंधित किया जाता है।
- **जेआईटी कम्पाइलर** : यह आईएल कोड को रनटाइम पर सिस्टम के लिए विशिष्ट मशीन कोड में परिवर्तित करता है, जिससे निष्पादन अनुकूलित होता है।
- **कचरा संग्रहकर्ता** : यह अप्रयुक्त वस्तुओं को मुक्त करके स्वचालित रूप से मेमोरी का प्रबंधन करता है, जिससे मैनुअल मेमोरी प्रबंधन की आवश्यकता कम हो जाती है।
- **कॉमन टाइप सिस्टम (CTS)** : यह सुनिश्चित करता है कि भाषाओं में अलग-अलग डेटा प्रकार CLR द्वारा समझे जा सकें और एक साथ काम कर सकें। CTS के 2 प्रकार हैं जो हर .NET प्रोग्रामिंग भाषा में होते हैं:
 - **वैल्यू टाइप**: वैल्यू टाइप वैल्यू को सीधे मेमोरी लोकेशन में स्टोर करेंगे। ये प्रकार केवल स्टैक मैकेनिज्म के साथ काम करते हैं। CLR संकलन समय पर इनके लिए मेमोरी की अनुमति देता है।
 - **संदर्भ प्रकार** : संदर्भ प्रकारों में मान का मेमोरी पता होगा क्योंकि संदर्भ प्रकार चर मान को सीधे मेमोरी में संग्रहीत नहीं करेंगे। ये प्रकार हीप तंत्र के साथ काम करते हैं। CLR रनटाइम पर इनके लिए मेमोरी आवंटित करता है।

नीचे दी गई तालिका .NET फ्रेमवर्क में CLR संस्करण को दर्शाती है।

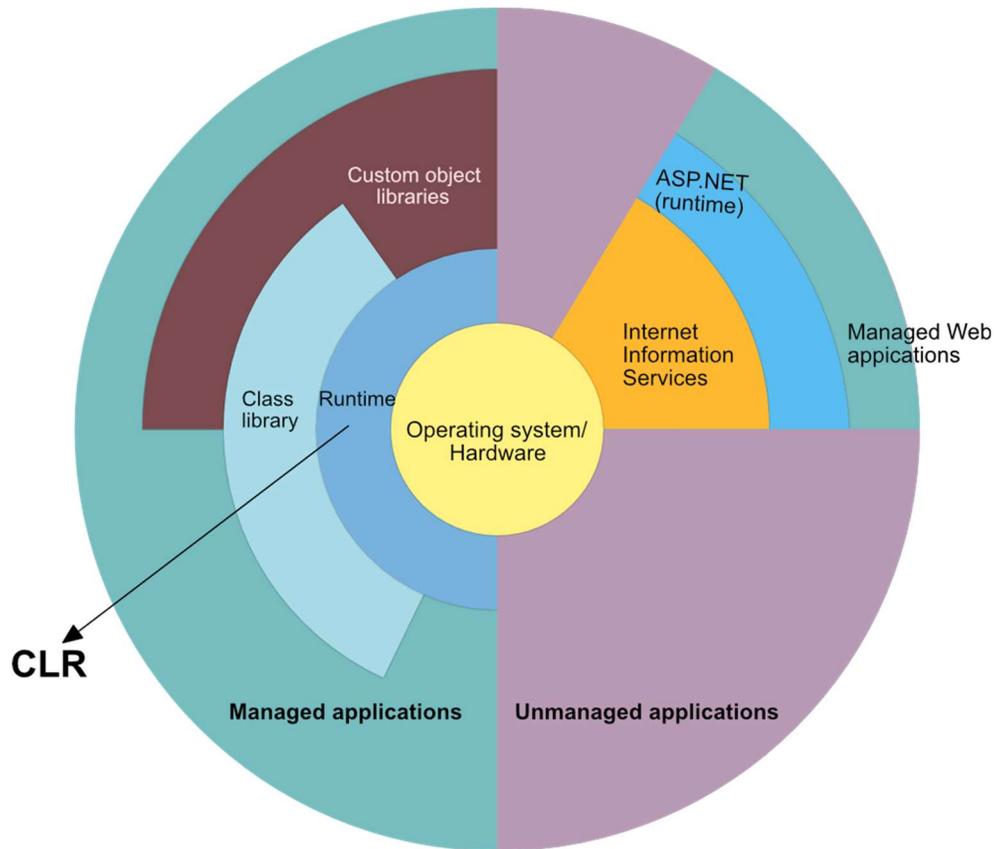
सीएलआर संस्करण	.NET फ्रेमवर्क संस्करण
1.0	1.0

DSG Support Multi Solution

सीएलआर संस्करण	.NET फ्रेमवर्क संस्करण
1.1	1.1
2.0	2.0
2.0	3.0
2.0	3.5
4	4
4	4.5 (4.5.1 और 4.5.2 भी)
4	4.6 (4.6.1 और 4.6.2 भी)
4	4.7 (4.7.1 और 4.7.2 भी)
4	4.8 (4.8.1 भी)
-	6
-	8.0
-	9.0

नीचे दिया गया आरेख दर्शाता है कि CLR ऑपरेटिंग सिस्टम/हार्डवेयर के साथ-साथ क्लास लाइब्रेरी से कैसे जुड़ा हुआ है। यहाँ, रनटाइम वास्तव में CLR है।

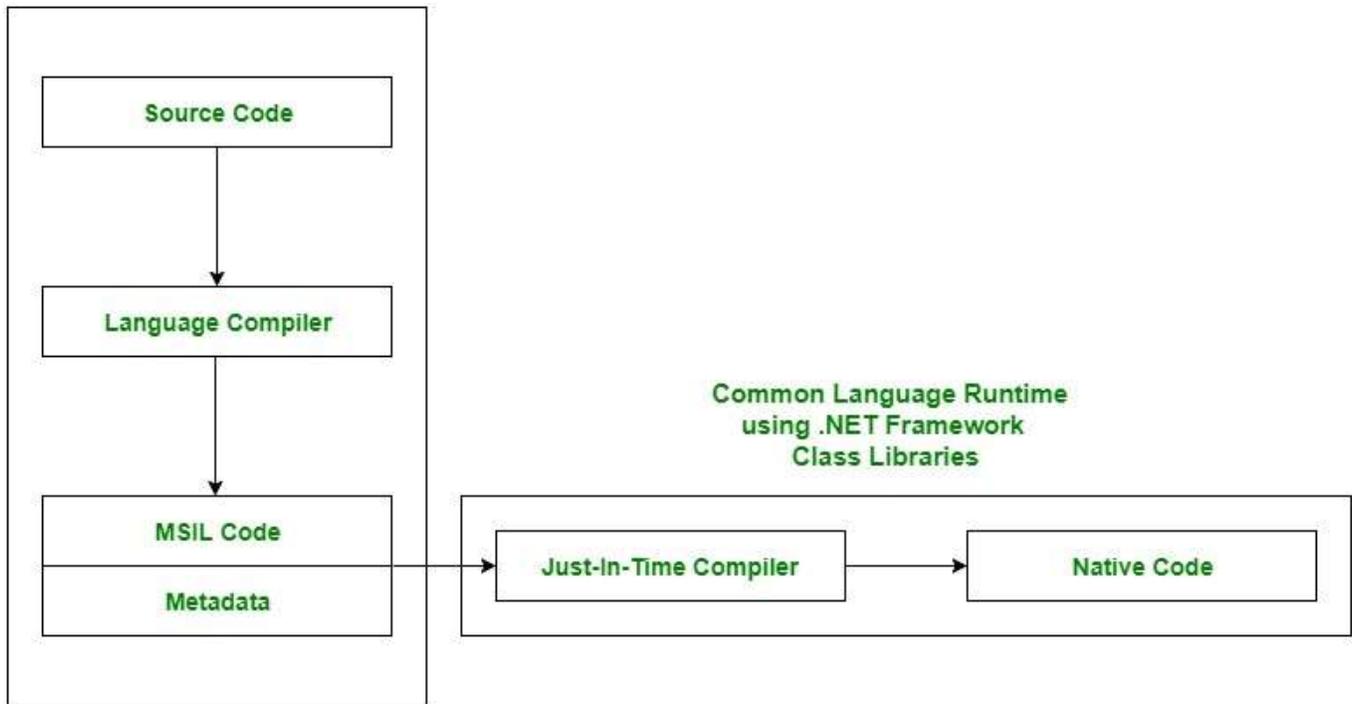
DSG Support Multi Solution



C# प्रोग्राम के निष्पादन में CLR की भूमिका

- मान लीजिए हमने एक C# प्रोग्राम लिखा है और उसे एक फाइल में सेव कर दिया है जिसे सोर्स कोड के नाम से जाना जाता है।
- भाषा विशिष्ट संकलक स्रोत कोड को MSIL (Microsoft इंटरमीडिएट भाषा) में संकलित करता है जिसे CIL (कॉमन इंटरमीडिएट भाषा) या IL (इंटरमीडिएट भाषा) के रूप में भी जाना जाता है, साथ ही इसके मेटाडेटा को भी संकलित करता है। मेटाडेटा में प्रोग्राम के प्रत्येक फंक्शन के सभी प्रकार, वास्तविक कार्यान्वयन शामिल हैं। MSIL मशीन-स्वतंत्र कोड है।
- अब CLR अस्तित्व में आता है। CLR MSIL को सेवाएँ और रनटाइम वातावरण प्रदान करता है। आंतरिक रूप से CLR में JIT (जस्ट-इन-टाइम) कंपाइलर शामिल होता है जो MSIL कोड को मशीन कोड में परिवर्तित करता है जिसे आगे CPU द्वारा निष्पादित किया जाता है। CLR .NET फ्रेमवर्क क्लास लाइब्रेरी का भी उपयोग करता है। मेटाडेटा CLR को प्रोग्रामिंग भाषा, वातावरण, संस्करण और क्लास लाइब्रेरी के बारे में जानकारी प्रदान करता है जिसके द्वारा CLR MSIL कोड को संभालता है। चूंकि CLR आम है इसलिए यह किसी अन्य भाषा में लिखे गए क्लास के इंस्टेंस को किसी अन्य भाषा में लिखे गए क्लास के मेथड को कॉल करने की अनुमति देता है।

DSG Support Multi Solution



सी.एल.आर. के लाभ

- यह रनटाइम निष्पादन और प्रोग्राम इंटरैक्शन को अनुकूलित करता है।
- ये प्रोग्राम किसी भी सिस्टम पर बिना पुनर्संकलन की आवश्यकता के चल सकते हैं जो CLR का समर्थन करता है।
- यह प्रकार सुरक्षा प्रदान करता है और कोड में संभावित सुरक्षा जोखिमों की जांच करता है।
- यह स्वचालित रूप से मेमोरी आवंटन और सफाई का काम संभालता है।
- यह विभिन्न भाषाओं में लिखे कोड को एक साथ काम करने की अनुमति देता है।
- इससे स्केलेबल, मल्टीथ्रेडेड अनुप्रयोगों का निर्माण आसान हो जाता है।

MSIL

MSIL (Microsoft Intermediate Language) जिसे **CIL (Common Intermediate Language)** भी कहा जाता है, स्रोत कोड से संकलक द्वारा उत्पन्न निर्देशों का एक सेट है। MSIL प्लेटफॉर्म-स्वतंत्र है, इसका मतलब है कि यह किसी भी ऐसे वातावरण पर चल सकता है जो .NET रनटाइम का समर्थन करता है। MSIL कोड को निष्पादित करने से पहले, इसे उस सिस्टम के लिए विशिष्ट मशीन कोड में परिवर्तित किया जाना चाहिए जिस पर यह चल रहा है। यह JIT कंपाइलर (जस्ट-इन-टाइम कंपाइलर) द्वारा किया जाता है, जो रनटाइम के दौरान आवश्यकतानुसार MSIL को संकलित करता है।

DSG Support Multi Solution

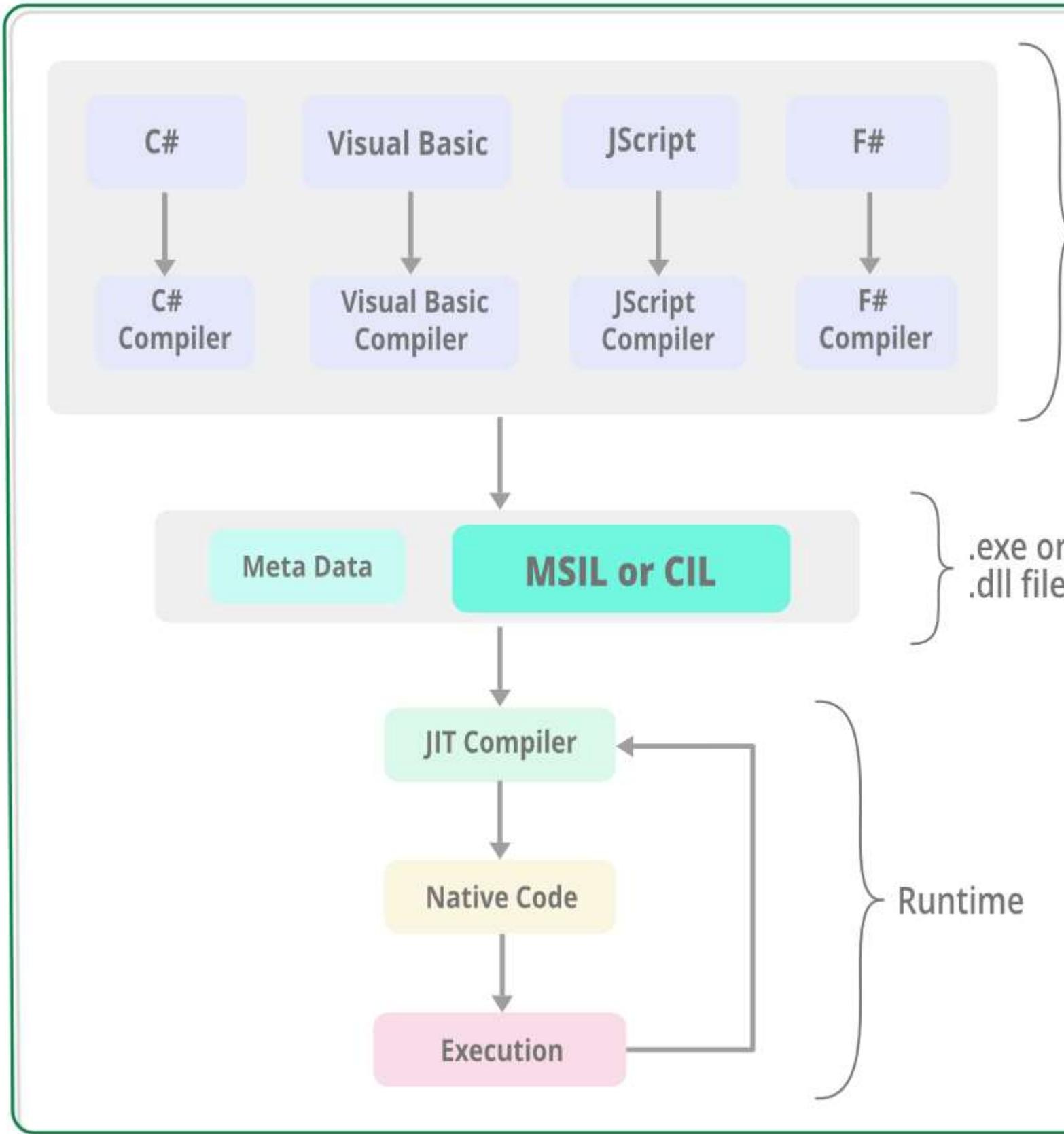
सामान्य भाषा रनटाइम (CLR) में निष्पादन प्रक्रिया

निष्पादन प्रक्रिया में MSIL का निर्माण और JIT कंपाइलर द्वारा MSIL को मशीन कोड में परिवर्तित करना शामिल है। इस प्रक्रिया में निम्नलिखित चरण शामिल हैं:

1. **MSIL निर्माण** : स्रोत कोड को भाषा-विशिष्ट संकलक द्वारा MSIL में संकलित किया जाता है। MSIL के साथ, मेटाडेटा उत्पन्न होता है, जिसमें प्रकार परिभाषाएँ, हस्ताक्षर और रनटाइम जानकारी जैसी जानकारी होती है।
2. **असेंबली निर्माण** : MSIL को कॉमन लैंग्वेज इंफ्रास्ट्रक्चर (CLI) असेंबली में पैक किया जाता है, जो सुरक्षा, परिनियोजन और संस्करण के लिए उपयोग की जाने वाली संकलित कोड लाइब्रेरी है। असेंबली के दो प्रकार हैं: प्रोसेस असेंबली (EXE) और लाइब्रेरी असेंबली (DLL)।
3. **JIT संकलन** : JIT संकलक MSIL को सिस्टम के लिए विशिष्ट मशीन कोड में परिवर्तित करता है। यह MSIL को ऑन-डिमांड संकलित करता है, जिसका अर्थ है कि रनटाइम पर केवल आवश्यक भागों को संकलित किया जाता है।
4. **निष्पादन**: JIT कंपाइलर द्वारा उत्पन्न मशीन कोड को प्रोसेसर द्वारा निष्पादित किया जाता है।



DSG Support Multi Solution



उदाहरण: सरल C# कोड और उसका MSIL

DSG Support Multi Solution

MSIL को नीचे दिए गए स्रोत कोड से भाषा-विशिष्ट संकलक द्वारा तैयार किया जाता है। MSIL को विस्तार से समझने के लिए, डेमो क्लास के साथ सरल C# स्रोत कोड जो "GeeksforGeeks" प्रिंट करता है, नीचे दिया गया है:

```
using System;

public class Demo {
    public static void Main()
    {
        Console.WriteLine("GeeksforGeeks");
    }
}
```

उपर दिए गए कोड के लिए C# कंपाइलर द्वारा बनाया गया MSIL निम्नानुसार है:

```
// ===== क्लास सदस्यों की घोषणा =====

.class public auto ansi beforefieldinit Demo
    extends [mscorlib]System.Object
{
    .method public hidebysig static void Main() cil managed
    {
        //
        .maxstack 8
        IL_0000: nop
        IL_0001: ldstr "GeeksforGeeks"
        IL_0006: call void [mscorlib]System.Console::WriteLine(string)
        IL_000b: nop
        IL_000c: ret
    } // विधि Demo::Main का अंत

    .method public hidebysig specialname rtspecialname
        इंस्टेंस void .ctor() cil managed
    {
        //
        .maxstack 8
        IL_0000: ldarg.0
        IL_0001: call इंस्टेंस void [mscorlib]System.Object::.ctor()
        IL_0006: ret
    } // विधि Demo::.ctor का अंत

} // वर्ग Demo का अंत

// =====
```

DSG Support Multi Solution

Visual Studio

Visual Studio एक एकीकृत विकास वातावरण (IDE) है जिसे Microsoft द्वारा डेस्कटॉप एप्लिकेशन, GUI (ग्राफिकल यूजर इंटरफ़ेस), कंसोल, वेब एप्लिकेशन, मोबाइल एप्लिकेशन, क्लाउड और वेब सेवाएँ आदि विकसित करने के लिए विकसित किया गया है। इस IDE की मदद से आप प्रबंधित कोड के साथ-साथ मूल कोड भी बना सकते हैं। यह Microsoft सॉफ्टवेयर डेवलपमेंट सॉफ्टवेयर के विभिन्न प्लेटफॉर्म जैसे Windows स्टोर, Microsoft Silverlight और Windows API आदि का उपयोग करता है। यह एक भाषा-विशिष्ट IDE नहीं है क्योंकि आप इसका उपयोग C#, C++, VB (विजुअल बेसिक), पायथन, जावास्क्रिप्ट और कई अन्य भाषाओं में कोड लिखने के लिए कर सकते हैं। यह 36 विभिन्न प्रोग्रामिंग भाषाओं के लिए समर्थन प्रदान करता है। यह Windows के साथ-साथ macOS के लिए भी उपलब्ध है।

विजुअल स्टूडियो का विकास

VS (विजुअल स्टूडियो) का पहला संस्करण 1997 में जारी किया गया था, जिसका नाम विजुअल स्टूडियो 97 था और इसका संस्करण नंबर 5.0 था। विजुअल स्टूडियो का नवीनतम संस्करण 15.0 है जिसे 7 मार्च, 2017 को जारी किया गया था। इसे विजुअल स्टूडियो 2017 भी कहा जाता है। नवीनतम विजुअल स्टूडियो में समर्थित .Net फ्रेमवर्क संस्करण 3.5 से 4.7 है। विजुअल स्टूडियो के पुराने संस्करणों में जावा का समर्थन किया गया था, लेकिन नवीनतम संस्करण में जावा भाषा के लिए कोई समर्थन नहीं दिया गया है।

विजुअल स्टूडियो संस्करण

माइक्रोसॉफ्ट विजुअल स्टूडियो के 3 संस्करण इस प्रकार हैं:

1. समुदाय

यह एक निःशुल्क संस्करण है जिसकी घोषणा 2014 में की गई थी। अन्य सभी संस्करण सशुल्क हैं। इसमें प्रोफेशनल संस्करण के समान सुविधाएँ हैं। इस संस्करण का उपयोग करके, कोई भी व्यक्तिगत डेवलपर अपने स्वयं के निःशुल्क या सशुल्क ऐप जैसे .Net एप्लिकेशन, वेब एप्लिकेशन और कई अन्य विकसित कर सकता है। एक उद्यम संगठन में, इस संस्करण की कुछ सीमाएँ हैं। उदाहरण के लिए, यदि आपके संगठन में 250 से अधिक पीसी हैं और वार्षिक राजस्व \$1 मिलियन (यूएस डॉलर) से अधिक है, तो आपको इस संस्करण का उपयोग करने की अनुमति नहीं है। एक गैर-उद्यम संगठन में, अधिकतम पाँच उपयोगकर्ता इस संस्करण का उपयोग कर सकते हैं। इसका मुख्य उद्देश्य इकोसिस्टम (हजारों एक्सटेंशन तक पहुँच) और भाषाओं (आप C#, VB, F#, C++, HTML, JavaScript, Python, आदि में कोड कर सकते हैं) का समर्थन प्रदान करना है।

DSG Support Multi Solution

2. पेशेवर

यह Visual Studio का व्यावसायिक संस्करण है। यह Visual Studio 2010 और उसके बाद के संस्करणों में उपलब्ध है। यह XML और XSLT संपादन के लिए सहायता प्रदान करता है और इसमें सर्वर एक्सप्लोरर और Microsoft SQL सर्वर के साथ एकीकरण जैसे उपकरण शामिल हैं। Microsoft इस संस्करण का निःशुल्क परीक्षण प्रदान करता है और परीक्षण अवधि के बाद, उपयोगकर्ता को इसका उपयोग जारी रखने के लिए भुगतान करना पड़ता है। इसका मुख्य उद्देश्य लचीलापन (किसी भी प्रकार के एप्लिकेशन के निर्माण के लिए पेशेवर डेवलपर उपकरण), उत्पादकता (कोडलेंस जैसी शक्तिशाली सुविधाएँ आपकी टीम की उत्पादकता में सुधार करती हैं), सहयोग (एजाइल प्रोजेक्ट प्लानिंग टूल, चार्ट, आदि) और Microsoft सॉफ्टवेयर, प्लस Azure, Pluralsight, आदि जैसे सब्सक्राइबर लाभ प्रदान करना है।

3. उद्यम

यह किसी भी आकार की टीमों के लिए एक एकीकृत, एंड टू एंड समाधान है, जिसकी मांग गुणवत्ता और पैमाने की आवश्यकताओं के साथ है। Microsoft इस संस्करण का 90-दिन का निःशुल्क परीक्षण प्रदान करता है और परीक्षण अवधि के बाद, उपयोगकर्ता को इसका उपयोग जारी रखने के लिए भुगतान करना होगा। इस संस्करण का मुख्य लाभ यह है कि यह अत्यधिक स्केलेबल है और उच्च गुणवत्ता वाला सॉफ्टवेयर प्रदान करता है।

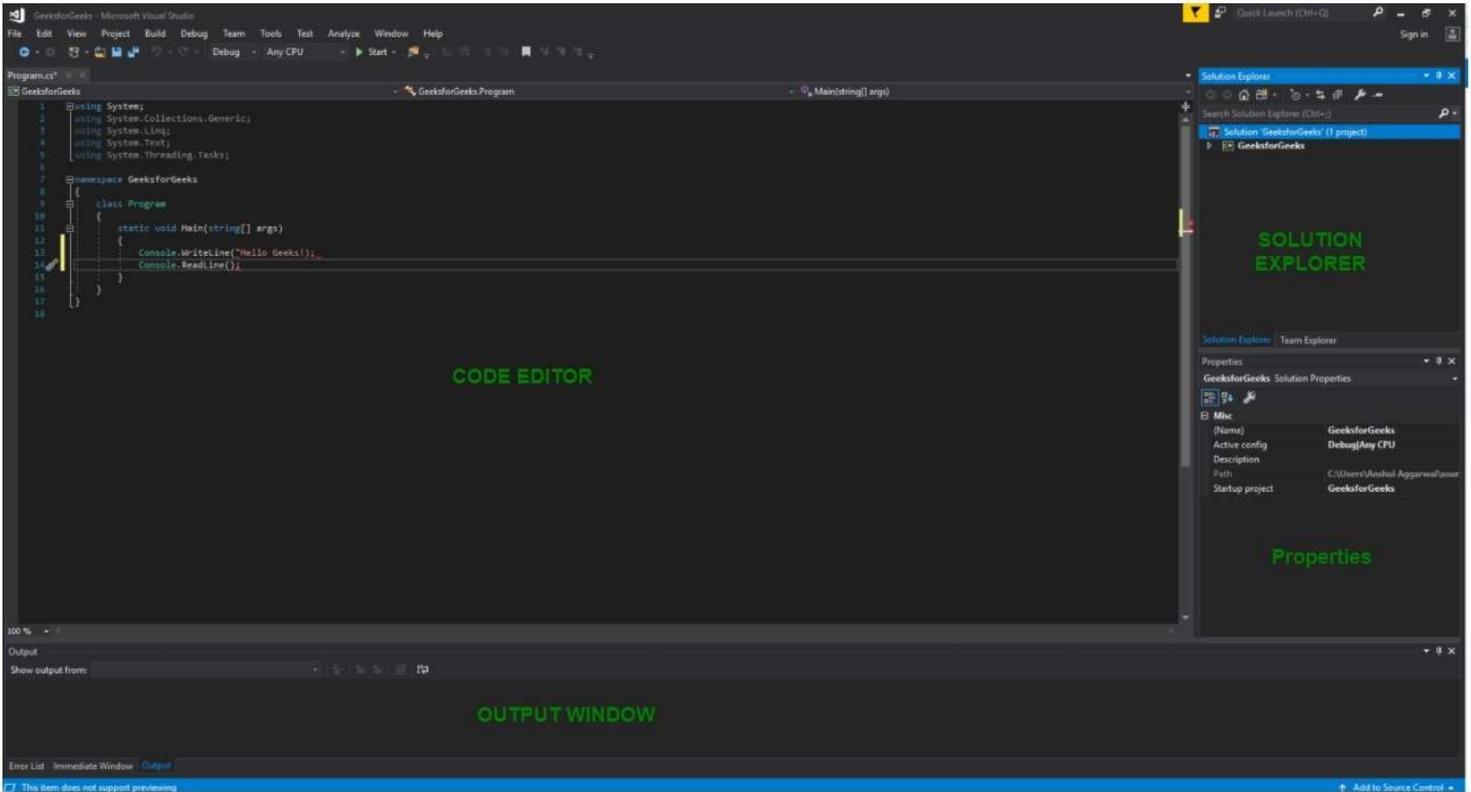
Visual Studio 2017 के साथ आरंभ करना

- सबसे पहले, आपको Visual Studio डाउनलोड और इंस्टॉल करना होगा। इसके लिए, आप [Visual Studio 2017 डाउनलोड और इंस्टॉल करना](#) देख सकते हैं। VS 2017 की स्थापना के दौरान .NET को रिकलोड का चयन करना न भूलें। यदि आप भूल जाते हैं तो आपको इंस्टॉलेशन को [संशोधित करना](#) होगा।

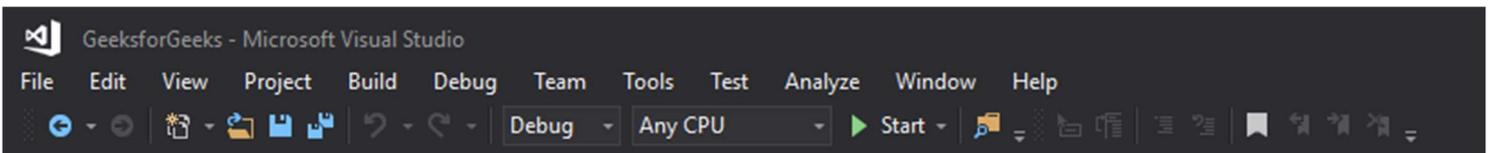


DSG Support Multi Solution

- जब आप विजुअल स्टूडियो खोलेंगे और अपना पहला प्रोग्राम लिखना शुरू करेंगे तो आपको कई टूल विंडो दिखाई देंगी, जैसे:



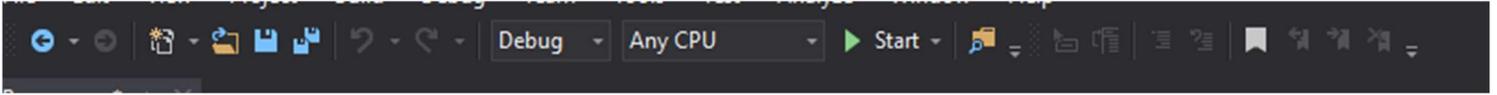
1. **कोड संपादक:** जहां उपयोगकर्ता कोड लिखेगा।
 2. **आउटपुट विंडो:** यहां विजुअल स्टूडियो आउटपुट, कंपाइलर चेतावनियां, त्रुटि संदेश और डिबगिंग जानकारी दिखाता है।
 3. **समाधान एक्सप्लोरर:** यह उन फ़ाइलों को दिखाता है जिन पर उपयोगकर्ता वर्तमान में काम कर रहा है।
 4. **गुण:** यह वर्तमान परियोजना के चयनित भागों के बारे में अतिरिक्त जानकारी और संदर्भ देगा।
- **उपयोगकर्ता व्यू मेनू से चुनकर अपनी आवश्यकता के अनुसार विंडो भी जोड़ सकता है।** विजुअल स्टूडियो में टूल विंडो को कस्टमाइज़ किया जा सकता है क्योंकि उपयोगकर्ता अधिक विंडो जोड़ सकता है, मौजूदा खुली विंडो को हटा सकता है या अपनी सुविधानुसार विंडो को इधर-उधर कर सकता है।
 - **विजुअल स्टूडियो में विभिन्न मेनू:** उपयोगकर्ता विजुअल स्टूडियो की शीर्ष स्क्रीन पर बहुत सारे मेनू पा सकता है जैसा कि नीचे दिखाया गया है



1. प्रोजेक्ट बनाएं, खोलें और सहेजें आदेश फ़ाइल मेनू में निहित हैं।

DSG Support Multi Solution

2. कोड खोजना, संशोधित करना, पुनर्रचना करना आदि आदेश संपादन मेनू में समाहित होते हैं।
 3. दृश्य मेनू का उपयोग विजुअल स्टूडियो में अतिरिक्त टूल विंडो खोलने के लिए किया जाता है।
 4. प्रोजेक्ट मेनू का उपयोग प्रोजेक्ट में कुछ फ़ाइलें और निर्भरताएँ जोड़ने के लिए किया जाता है।
 5. सेटिंग्स बदलने के लिए, एक्सटेंशन के माध्यम से विजुअल स्टूडियो में कार्यक्षमता जोड़ने के लिए, तथा विभिन्न विजुअल स्टूडियो टूल्स तक पहुंचने के लिए टूल्स मेनू का उपयोग किया जा सकता है।
- नीचे दिए गए मेनू को टूलबार के नाम से जाना जाता है जो सबसे ज़्यादा इस्तेमाल किए जाने वाले कमांड तक त्वरित पहुँच प्रदान करता है। आप **View** → **Customize** पर जाकर कमांड जोड़ और हटा सकते हैं



विजुअल स्टूडियो IDE का उपयोग करने के लाभ

- कई ऑपरेटिंग सिस्टम, वेब और क्लाउड के लिए एक पूर्ण-विशेषताओं वाला प्रोग्रामिंग प्लेटफ़ॉर्म, विजुअल स्टूडियो IDE उपलब्ध है। उपयोगकर्ता आसानी से UI ब्राउज़ कर सकते हैं ताकि वे अपना कोड तेज़ी से और सटीक रूप से लिख सकें।
- डेवलपर्स को कोड में संभावित त्रुटियों को शीघ्रता से पहचानने में मदद करने के लिए, विजुअल स्टूडियो एक मजबूत डिबगिंग टूल प्रदान करता है।
- डेवलपर्स अपने एप्लिकेशन को सर्वर पर विश्वास के साथ होस्ट कर सकते हैं, क्योंकि उन्होंने ऐसी किसी भी चीज को समाप्त कर दिया है जो प्रदर्शन संबंधी समस्याएं पैदा कर सकती थी।
- डेवलपर्स चाहे कोई भी प्रोग्रामिंग भाषा इस्तेमाल कर रहे हों, विजुअल स्टूडियो के उपयोगकर्ता लाइव कोडिंग सहायता प्राप्त कर सकते हैं। तेज़ विकास के लिए, प्लेटफ़ॉर्म एक स्वतः पूर्ण विकल्प प्रदान करता है। अंतर्निहित बुद्धिमान प्रणाली API के लिए विवरण और सुझाव प्रदान करती है।
- Visual Studio IDE के ज़रिए आप एक ही प्रोजेक्ट में अपने साथियों के साथ आसानी से सहयोग कर सकते हैं। यह IDE डेवलपर्स को अपने साथियों के साथ अपना कोड शेयर करने, पुश करने और खींचने में मदद करता है।
- विजुअल स्टूडियो के हर उपयोगकर्ता के पास इसे कस्टमाइज़ करने की क्षमता होती है। उनके पास अपनी ज़रूरतों के हिसाब से सुविधाएँ जोड़ने का विकल्प होता है। उदाहरण के लिए, वे ऐड-ऑन डाउनलोड कर सकते हैं और अपने IDE में एक्सटेंशन इंस्टॉल कर सकते हैं। यहाँ तक कि प्रोग्रामर भी अपने खुद के एक्सटेंशन सबमिट कर सकते हैं।

DSG Support Multi Solution

IDE

IDE (एकीकृत विकास वातावरण) एक ऐसा सॉफ्टवेयर है जो आम तौर पर इस्तेमाल किए जाने वाले डेवलपर टूल को एक कॉम्पैक्ट GUI (ग्राफिकल यूजर इंटरफ़ेस) एप्लिकेशन में जोड़ता है। यह एक एकीकृत टर्मिनल के साथ कोड एडिटर, कोड कंपाइलर और कोड डीबगर जैसे टूल का संयोजन है।

सॉफ्टवेयर संपादन, निर्माण, परीक्षण और पैकेजिंग जैसी सुविधाओं को एक सरल-से-उपयोग उपकरण में एकीकृत करके, IDE डेवलपर उत्पादकता को बढ़ाने में मदद करते हैं। IDE का उपयोग आमतौर पर प्रोग्रामर और सॉफ्टवेयर डेवलपर्स द्वारा उनकी प्रोग्रामिंग यात्रा को आसान बनाने के लिए किया जाता है।

आईडीई क्यों महत्वपूर्ण हैं?

किसी को निश्चित रूप से कोड लिखने या एप्लिकेशन विकसित करने के लिए IDE की आवश्यकता नहीं होती है। कोड लिखने के लिए नोटपैड जैसे सरल टेक्स्ट एडिटर का भी उपयोग किया जा सकता है। हालाँकि, IDE कुछ आश्चर्यजनक सुविधाएँ प्रदान करते हैं जो सामान्य संपादन से परे हैं। एक सरल इंटरफ़ेस में अक्सर उपयोग किए जाने वाले डेवलपर टूल प्रदान करके, कोई भी व्यक्ति विकास वातावरण को मैन्युअल रूप से कॉन्फ़िगर करने और एकीकृत करने की कठिनाई से गुज़रे बिना सीधे अपने एप्लिकेशन बनाना शुरू कर सकता है।

आईडीई (एकीकृत विकास पर्यावरण) की सामान्य विशेषताएं

आईडीई कई प्रकार की सुविधाएं प्रदान करते हैं जिनमें आमतौर पर निम्नलिखित शामिल हैं:

- **संपादक:** आमतौर पर एक पाठ संपादक दृश्य संकेतों के साथ वाक्यविन्यास को उजागर करके, भाषा-विशिष्ट स्वतः-पूर्णता प्रदान करके, और टाइप करते समय बग की जांच करके आपको सॉफ्टवेयर कोड लिखने में मदद कर सकता है।
- **कंपाइलर:** एक कंपाइलर मानव-पठनीय कोड को मशीन-विशिष्ट कोड में परिवर्तित करता है जिसे **लिनक्स**, **विंडोज** या **मैक ओएस** जैसे विभिन्न ऑपरेटिंग सिस्टम पर निष्पादित किया जा सकता है। अधिकांश IDE आमतौर पर उस भाषा के लिए बिल्ट-इन कंपाइलर के साथ आते हैं जिसका वे समर्थन करते हैं।
- **डीबगर:** एक उपकरण जो डेवलपर्स को उनके अनुप्रयोगों का परीक्षण और डीबग करने में सहायता कर सकता है और यदि कोई बग या त्रुटि हो तो उसके स्थानों को ग्राफिक रूप से इंगित कर सकता है।
- **बिल्ट-इन टर्मिनल:** टर्मिनल एक टेक्स्ट-आधारित इंटरफ़ेस है जिसका उपयोग मशीन के ऑपरेटिंग सिस्टम के साथ बातचीत करने के लिए किया जा सकता है। डेवलपर्स एक बिल्ट-इन टर्मिनल/कंसोल के साथ सीधे IDE के भीतर स्क्रिप्ट या कमांड चला सकते हैं।

DSG Support Multi Solution

- **संस्करण नियंत्रण:** संस्करण नियंत्रण सॉफ्टवेयर के विकास में स्पष्टता लाने में मदद करता है। कुछ IDEs **Git** जैसे संस्करण नियंत्रण उपकरणों का भी समर्थन करते हैं, जिसके माध्यम से उपयोगकर्ता सॉफ्टवेयर कोड में परिवर्तनों को ट्रैक और प्रबंधित कर सकता है।
- **कोड स्निपेट:** IDE कोड स्निपेट का समर्थन करते हैं जिनका उपयोग आमतौर पर एकल कार्य को पूरा करने के लिए किया जाता है और यह कुछ हद तक अनावश्यक कार्य को भी कम कर सकता है।
- **एक्सटेंशन और प्लगइन्स:** एक्सटेंशन और प्लगइन्स का उपयोग विशिष्ट प्रोग्रामिंग भाषाओं के संबंध में IDE की कार्यक्षमता को बढ़ाने के लिए किया जाता है ।
- **कोड नेविगेशन:** IDE में कोड फोल्डिंग, क्लास और मेथड नेविगेशन, तथा रीफैक्टरिंग टूल जैसे उपकरण होते हैं, जो कोड को पढ़ना और उसका विश्लेषण करना सरल बनाते हैं।

डेवलपर्स IDE (इंटीग्रेटेड डेवलपमेंट एनवायरनमेंट) का उपयोग क्यों करते हैं ?

विकास प्रक्रिया के सभी पहलुओं के प्रबंधन के लिए एकल, एकीकृत वातावरण प्रदान करके, IDE डेवलपर की उत्पादकता, कोड गुणवत्ता और समग्र विकास अनुभव को बेहतर बनाने में मदद कर सकता है।

- **उत्पादकता :** कोड संपादन, निष्पादनयोग्य फाइलें बनाना, डिबगिंग और परीक्षण जैसी सामान्य गतिविधियों को सॉफ्टवेयर/अनुप्रयोग विकास के भाग के रूप में संयोजित करके, IDEs समय को कम करने और समग्र उत्पादकता को बढ़ाने में मदद करते हैं।
- **कोड गुणवत्ता :** IDEs एक ही **GUI** के हिस्से के रूप में बिल्ट-इन टूल के साथ आते हैं, जिसके माध्यम से डेवलपर्स अनुप्रयोगों के बीच स्विच किए बिना क्रियाएँ निष्पादित कर सकते हैं। इसके अलावा, यह सिंटैक्स हाइलाइटिंग, कोड रीफैक्टरिंग और कोड विश्लेषण में भी मदद कर सकता है जिससे समग्र कोड गुणवत्ता में वृद्धि होती है।
- **एकीकृत वातावरण :** IDE विकास उपकरणों के संयोजन के साथ पहले से निर्मित आते हैं जो डेवलपर्स को नए अनुप्रयोगों को जल्दी से प्रोग्रामिंग शुरू करने की अनुमति देते हैं। IDE के साथ, सेटअप प्रक्रिया के हिस्से के रूप में कई उपयोगिताओं को मैन्युअल रूप से कॉन्फिगर और एकीकृत करना अब आवश्यक नहीं है। इसके अतिरिक्त, चूंकि प्रत्येक उपयोगिता एक ही कार्यक्षेत्र पर उपलब्ध है, इसलिए डेवलपर्स को प्रत्येक को अलग-अलग उपयोग करने का तरीका सीखने में घंटों खर्च नहीं करने पड़ते हैं।
- **अनुकूलनशीलता :** कस्टम रंग योजनाओं और कीबोर्ड शॉर्टकट से लेकर अद्वितीय लेआउट, विभिन्न प्लगइन्स और ऐड-ऑन चुनने तक के अनुकूलन विकल्पों को शामिल करके; IDE डेवलपर्स को अपने वातावरण को अपनी अनूठी आवश्यकताओं और स्वाद के अनुसार अनुकूलित करने में सक्षम बनाता है, जिससे विकास प्रक्रिया की सुविधा और दक्षता में सुधार होता है।

DSG Support Multi Solution

आईडीई के प्रकार

IDE कई तरह के होते हैं, कुछ खास भाषा के लिए काम करने के लिए डिज़ाइन किए जाते हैं जबकि कुछ मोबाइल डिवाइस जैसे खास प्लेटफ़ॉर्म के लिए लक्षित होते हैं। इसलिए, अपनी ज़रूरतों के हिसाब से सबसे उपयुक्त IDE चुनना भी उतना ही महत्वपूर्ण हो जाता है।

1. डेस्कटॉप आईडीई

इस प्रकार में वे IDE शामिल हैं जिन्हें स्थानीय रूप से कॉन्फ़िगर किया जा सकता है। प्रोग्राम बनाने/चलाने के लिए उन्हें सक्रिय इंटरनेट कनेक्शन की आवश्यकता नहीं होती है। डेस्कटॉप IDE डेवलपर-विशिष्ट आवश्यकताओं के अनुरूप अत्यधिक अनुकूलन योग्य होते हैं और इंटरनेट की गति के बावजूद प्रदर्शन प्रदान करते हैं। **उदाहरण: Microsoft Visual Studio, Eclipse, Netbeans, आदि** शामिल हैं।

2. क्लाउड आईडीई

क्लाउड IDE स्थानीय रूप से सॉफ़्टवेयर को कॉन्फ़िगर करने के ओवरहेड को समाप्त करते हैं। वे दूरस्थ सर्वर पर चलते हैं और डेस्कटॉप ब्राउज़र के माध्यम से एक्सेस किए जा सकते हैं। क्लाउड IDE बेहतर पहुँच और प्लेटफ़ॉर्म स्वतंत्रता प्रदान करते हैं, जिससे उन्हें क्लाउंट ऑपरेटिंग सिस्टम की परवाह किए बिना इंटरनेट के माध्यम से एक्सेस किया जा सकता है। क्लाउड IDE का उपयोग स्वतंत्र रूप से किया जा सकता है या प्रदाता के संबंध में मूल्य निर्धारण मॉडल हो सकते हैं। **उदाहरण: Gitpod, AWS Cloud 9, Replit, आदि** शामिल करें।

3. मोबाइल ऐप डेवलपमेंट आईडीई

ये IDE खास तौर पर मोबाइल एप्लीकेशन बनाने के लिए डिज़ाइन किए गए हैं। इनमें मोबाइल एप्लीकेशन विकसित करने और उनका परीक्षण करने के लिए एमुलेटर सपोर्ट और इंटीग्रेशन जैसी सुविधाएँ शामिल हैं। मोबाइल डेवलपमेंट IDE डेस्कटॉप IDE या क्लाउड IDE के रूप में हो सकते हैं। **उदाहरण: Android Studio, Flutlab.io, आदि** शामिल करें।

4. डेटाबेस-विशिष्ट आईडीई

ये IDE (एकीकृत विकास वातावरण) विशेष रूप से डेटाबेस के साथ काम करने के लिए डिज़ाइन किए गए हैं। इनमें क्वेरी बिल्डर्स और मोबाइल एप्लीकेशन विकसित करने और परीक्षण करने जैसी सुविधाएँ शामिल हैं। मोबाइल-डेवलपमेंट IDE की तरह, डेटाबेस IDE भी डेस्कटॉप IDE या क्लाउड IDE के रूप में हो सकते हैं। **उदाहरण: MySQL वर्कबेंच, Oracle SQL डेवलपर, आदि** शामिल करें।

DSG Support Multi Solution

मुझे आईडीई का चयन कैसे करना चाहिए?

बाजार में कई IDE उपलब्ध हैं, जबकि कुछ ओपन-सोर्स और उपयोग करने के लिए मुफ्त हैं, अन्य सब्सक्रिप्शन मॉडल के साथ काम कर सकते हैं। लोकप्रिय IDE में Visual Studio, PyCharm, Android Studio आदि शामिल हैं।

1. प्रोग्रामिंग भाषा : IDE (एकीकृत विकास वातावरण) का चुनाव मुख्य रूप से आपके द्वारा चुनी गई प्रोग्रामिंग भाषा पर निर्भर करता है। समर्पित IDE में स्वचालित सुविधाएँ होती हैं जो उन प्रोग्रामिंग भाषाओं के लिए सबसे उपयुक्त होती हैं जिनका वे समर्थन करते हैं। जबकि ऐसे IDE भी मौजूद हैं जो कई प्रोग्रामिंग भाषाओं का समर्थन करते हैं।

2. ऑपरेटिंग सिस्टम : ज्यादातर सभी IDE अलग-अलग ऑपरेटिंग सिस्टम के साथ काम करते हैं, लेकिन कुछ IDE ऐसे भी होते हैं जो सिर्फ़ खास प्लैटफ़ॉर्म पर ही सबसे बेहतर काम करते हैं। आप IDE की अलग-अलग प्लैटफ़ॉर्म और ऑपरेटिंग सिस्टम के साथ संगतता को ध्यान में रख सकते हैं।

3. मूल्य निर्धारण मॉडल: कुछ IDE मुफ्त में उपलब्ध हो सकते हैं जबकि कुछ को अधिकांश सुविधाओं को अनलॉक करने के लिए प्रीमियम सदस्यता मॉडल की आवश्यकता हो सकती है। अपने बजट और अपनी ज़रूरतों को ध्यान में रखते हुए, उस IDE को चुनें जो आपको सबसे अच्छा लगे।

4. विशेषताएं: संस्करण नियंत्रण, डिबगर, कोड हाइलाइटर, एकीकृत टर्मिनल और कोड रिफैक्टरिंग जैसी विशेषताएं कुछ ऐसी विशेषताएं हैं जिन्हें आईडीई चुनते समय ध्यान में रखा जा सकता है।

आईडीई का उपयोग करने के लाभ

आईडीई या एकीकृत विकास वातावरण के कुछ सामान्य लाभ इस प्रकार हैं:

- **प्रोजेक्ट प्रबंधन:** IDE प्रोजेक्ट निर्देशिका संरचना को देखना बहुत आसान बनाता है जो कई फ़ाइलों के साथ काम करने के थकाऊ कार्यों को तोड़ देता है। आम तौर पर IDE में कई उपकरण भी शामिल होते हैं, जैसे कि सिंटैक्स हाइलाइटिंग, कोड पूर्णता और कोड फोल्डिंग, जो समग्र प्रोजेक्ट को प्रबंधित करना आसान बनाता है।
- **बहुत समय और प्रयास की बचत:** IDE में अक्सर कई तरह के उपकरण शामिल होते हैं जो न केवल आपके कोड को व्यवस्थित करने में आपकी सहायता कर सकते हैं बल्कि आपके कोड में त्रुटि को भी तुरंत उजागर कर सकते हैं। इनमें आपके कोड को संकलित करने, बनाने, परीक्षण करने और तैनात करने के लिए उपकरणों का एक लचीला संयोजन भी शामिल है, जिसके माध्यम से आप इन कार्यों को स्वचालित कर सकते हैं और प्रत्येक कार्य को मैन्युअल रूप से चलाने की आवश्यकता को समाप्त कर सकते हैं।
- **उत्पादकता:** IDE आपकी सभी विकास आवश्यकताओं के लिए एक केंद्रीकृत स्थान प्रदान करके आपकी उत्पादकता बढ़ा सकते हैं, कोड लिखने और संपादित करने से लेकर डिबगिंग और तैनाती तक। IDE

DSG Support Multi Solution

द्वारा प्रदान की जाने वाली कुछ कार्यक्षमता को टेक्स्ट एडिटर और कमांड-लाइन टूल के साथ भी पूरा किया जा सकता है, लेकिन IDE आमतौर पर प्रक्रिया को अधिक सुविधाजनक और उपयोगकर्ता के अनुकूल बनाते हैं।

आईडीई का उपयोग करने के नुकसान

आईडीई या एकीकृत विकास वातावरण के कुछ सामान्य नुकसान इस प्रकार हैं:

- **शुरुआती लोगों के लिए अनुकूल नहीं:** IDE जटिल उपकरण हैं और आम तौर पर काफी डराने वाला UI प्रदान करते हैं, जिसे समझना शुरुआती लोगों के लिए थोड़ा कठिन हो सकता है। उनके लाभ को अधिकतम करने के लिए आम तौर पर एक समर्पित प्रयास की आवश्यकता होती है।
- **लगातार अपडेट:** नए नमूने, टेम्पलेट्स और सुविधाओं जैसे लगातार अपडेट को बनाए रखना मुश्किल है।
- **केवल कोड लिखने में सहायता करें:** एक उपकरण के रूप में, IDE (एकीकृत विकास वातावरण) केवल सहायक होने तक ही सीमित हैं। वे कोड नहीं लिख सकते हैं या स्वचालित रूप से त्रुटियों को ठीक नहीं कर सकते हैं, फिर भी साफ कोड लिखने के लिए ज्ञान की आवश्यकता होती है।
- **लागत:** कुछ IDE महंगे होते हैं या कुछ ऐड-ऑन को अनलॉक करने के लिए अतिरिक्त लागत की आवश्यकता होती है, जो छोटे पैमाने की कंपनियों या सीखने वाले व्यक्तियों के लिए बाधा बन सकती है।

Event Driven Programming

इवेंट-ड्रिवन प्रोग्रामिंग एक प्रतिमान है जहाँ प्रोग्राम का निष्पादन उपयोगकर्ता क्रियाओं या संदेशों जैसी घटनाओं द्वारा निर्धारित होता है। प्रोग्राम पूर्वनिर्धारित क्रियाओं के साथ घटनाओं पर प्रतिक्रिया करते हैं, जिससे अतुल्यकालिक और प्रतिक्रियाशील व्यवहार की अनुमति मिलती है, जिसे अक्सर GUI अनुप्रयोगों और वितरित प्रणालियों में देखा जाता है।

इवेंट-संचालित प्रोग्रामिंग प्रतिमान के लाभ

- अतुल्यकालिक प्रसंस्करण को सक्षम बनाता है, संसाधन उपयोग और प्रतिक्रियाशीलता को अनुकूलित करता है, जो वास्तविक समय अनुप्रयोगों और उपयोगकर्ता इंटरफेस के लिए महत्वपूर्ण है।
- मॉड्यूलर कोड डिजाइन को प्रोत्साहित करता है, चिंताओं को अलग करके रखरखाव और मापनीयता को सरल बनाता है और कोड पुनः प्रयोज्यता को बढ़ावा देता है।
- उपयोगकर्ता के इनपुट पर तुरंत प्रतिक्रिया देकर उपयोगकर्ता अनुभव को बढ़ाता है, तथा एक सहज और अधिक इंटरैक्टिव इंटरफेस प्रदान करता है।
- यह नई सुविधाओं या संशोधनों के आसान एकीकरण की सुविधा प्रदान करता है, तथा गतिशील वातावरण में बदलती आवश्यकताओं के लिए अनुकूलनशीलता को बढ़ावा देता है।

DSG Support Multi Solution

- घटक घटनाओं के माध्यम से संचार करते हैं, जिससे निर्भरता कम होती है और सिस्टम का लचीलापन बढ़ता है, जिससे रखरखाव और संशोधन आसान हो जाता है।

इवेंट-ड्रिवेन प्रोग्रामिंग प्रतिमान के नुकसान

- घटना-संचालित प्रणालियों को उनकी अतुल्यकालिक प्रकृति के कारण डीबग करना चुनौतीपूर्ण हो सकता है, जिससे त्रुटियों का पता लगाना अधिक कठिन हो जाता है।
- समवर्ती घटनाएं रेस स्थितियां उत्पन्न कर सकती हैं, जिससे अप्रत्याशित व्यवहार उत्पन्न हो सकता है तथा डिबगिंग और सिंक्रोनाइजेशन जटिल हो सकता है।
- घटना-संचालित प्रणालियां नियंत्रण को उलट सकती हैं, जिससे डिजाइन से अपरिचित डेवलपर्स के लिए कोड का अनुसरण करना और समझना कठिन हो जाता है।
- परस्पर संबद्ध घटनाओं की श्रृंखला से व्यापक प्रभाव उत्पन्न हो सकता है, जिससे परिणाम की भविष्यवाणी करना और सिस्टम स्थिति का प्रबंधन करना कठिन हो जाता है।
- घटनाओं को लगातार सुनने से सिस्टम संसाधनों का उपभोग हो सकता है, जिससे संसाधन उपयोग में संभावित अकुशलताएं उत्पन्न हो सकती हैं और समग्र सिस्टम प्रदर्शन प्रभावित हो सकता है।

इवेंट-ड्रिवेन आर्किटेक्चर (EDA)

इवेंट-ड्रिवेन आर्किटेक्चर (EDA) बटन क्लिक जैसी क्रियाओं पर प्रतिक्रिया करता है। इस प्रतिमान में, "बटन क्लिक" जैसी घटनाएँ पूर्वनिर्धारित क्रियाओं को ट्रिगर करती हैं। उपयोगकर्ता इंटरफ़ेस के साथ इंटरैक्ट करता है, और सिस्टम, इवेंट-ड्रिवेन प्रोग्रामिंग मॉडल का पालन करते हुए, एक उत्तरदायी और आकर्षक उपयोगकर्ता अनुभव सुनिश्चित करने के लिए गतिशील रूप से प्रतिक्रिया करता है।

उदाहरण: जावास्क्रिप्ट इवेंटडिस्पैचर के माध्यम से बुनियादी इवेंट हैंडलिंग का प्रदर्शन करने के लिए, इवेंट-ड्रिवेन आर्किटेक्चर का उपयोग करके अलर्ट का संकेत देना।

इवेंट-संचालित प्रोग्रामिंग प्रतिमान में कॉलबैक फ़ंक्शन

इवेंट-ड्रिवेन प्रोग्रामिंग प्रतिमान में, हम कॉलबैक फ़ंक्शन का उपयोग कर रहे हैं। इसमें बटन क्लिक जैसे विशिष्ट इवेंट को संभालने के लिए फ़ंक्शन को परिभाषित करना शामिल है। इवेंट श्रोताओं के साथ इन कॉलबैक को पंजीकृत करके, प्रोग्राम उपयोगकर्ता क्रियाओं पर गतिशील रूप से प्रतिक्रिया करता है, जो इंटरैक्टिव और उत्तरदायी अनुप्रयोगों के लिए इवेंट-ड्रिवेन आर्किटेक्चर के सार को मूर्त रूप देता है।

VB.Net & VB Difference

VB.NET का मतलब है Visual Basic. नेटवर्क सक्षम तकनीकें। Microsoft ने 2001 में .NET प्लेटफ़ॉर्म जारी किया, जो Visual Basic .NET का समर्थन करता है जो VB प्रोग्रामिंग भाषा के अंतिम संस्करण का अपग्रेड

DSG Support Multi Solution

है। यह [Microsoft .NET फ्रेमवर्क](#) के लिए एक उच्च-स्तरीय प्रोग्रामिंग भाषा है । Linux और MAC ऑपरेटिंग सिस्टम पर VB.NET चलाना भी संभव है।

विजुअल बेसिक या VB एक प्रोग्रामिंग भाषा है जिसे पहली बार 1991 में रिलीज़ किया गया था। यह पहली प्रोग्रामिंग भाषा थी जो भाषा-आपूर्ति वाली वस्तुओं का उपयोग करके सीधे प्रोग्राम करने योग्य ग्राफ़िकल यूज़र इंटरफ़ेस का समर्थन करती थी। उस समय से लेकर 2002 तक, इसके 7 अन्य संस्करण रिलीज़ हुए, जिनमें से प्रत्येक संस्करण में ऐसी विशेषताएँ थीं जो भाषा की शक्ति को बढ़ाती थीं।

VB.NET और विजुअल बेसिक के बीच अंतर:

पैरामीटर	वीबी .NET	मूल दृश्य
प्रोग्रामिंग भाषा प्रतिमान	VB.NET एक ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग भाषा है।	विजुअल बेसिक इवेंट संचालित प्रोग्रामिंग भाषा है।
इंटेलिसेंस अनुकूलता	VB.NET IntelliSense संगतता का समर्थन करता है।	यह IntelliSense संगतता का समर्थन नहीं करता है।
प्रकार	यह एक संकलित टाइप की गई भाषा है	यह एक दुभाषिया आधारित भाषा है
बहु	यह मल्टीथ्रेडेड की अवधारणा का समर्थन करता है।	यह बहु-थ्रेडेड अवधारणा का समर्थन नहीं करता है।
प्रकार-सुरक्षित	यह एक प्रकार-सुरक्षित भाषा है।	यह एक प्रकार-सुरक्षित भाषा नहीं है।
डेटा प्रकार में परिवर्तन	VB.NET में, VARIANT और CURRENCY वैरिएबल हटा दिए गए	Visual Basic VARIANT और CURRENCY में, चरों का उपयोग किया जाता है तथा DECIMAL चर को शामिल नहीं किया जाता है।

DSG Support Multi Solution

पैरामीटर	वीबी .NET	मूल दृश्य
	हैं और DECIMAL वेरिएबल पेश किया गया है।	
पैरामीटर पासिंग	VB.NET में पैरामीटर्स को मान द्वारा पास किया जाता है।	विजुअल बेसिक में पैरामीटर्स को संदर्भ द्वारा पास किया जाता है।
डेटा संधारण	VB.NET में, डेटा को ADO.net का उपयोग करके प्रबंधित किया जाता है	VB में डेटा को DAO प्रोटोकॉल का उपयोग करके नियंत्रित किया जाता है, उसके बाद RDO और उसके बाद ADO का उपयोग किया जाता है।
पृष्ठभूमि संकलन	VB.NET में प्रत्येक वैध कथन के लिए पृष्ठभूमि संकलन निरंतर होता रहता है।	विजुअल बेसिक एक व्याख्याकृत भाषा है, इसमें प्रत्येक कथन को पहले ऑब्जेक्ट कोड में परिवर्तित किया जाता है, फिर स्रोत कोड में परिवर्तित किया जाता है, जिससे पृष्ठभूमि संकलन की आवश्यकता नहीं होती।
रनटाइम वातावरण	VB.NET कॉमन लैंग्वेज रनटाइम वातावरण का उपयोग करता है।	Visual Basic सामान्य Visual Basic रनटाइम वातावरण का उपयोग करता है।
फॉर्म	VB.NET फॉर्म में कोई आकार नियंत्रण और रेखा नियंत्रण नहीं है।	विजुअल बेसिक फॉर्म में आकृति नियंत्रण और रेखा नियंत्रण होता है।
शॉर्ट सर्किट लॉजिक	VB.NET में, शॉर्ट सर्किट लॉजिक उपलब्ध है।	विजुअल बेसिक में शॉर्ट सर्किट लॉजिक की कोई अवधारणा नहीं है।
अनुप्रयोग विकास	VB.NET में हम विभिन्न प्रकार के अनुप्रयोग बना सकते हैं जैसे कंसोल	विजुअल बेसिक विभिन्न प्रकार के अनुप्रयोगों के निर्माण के लिए बहुमुखी नहीं है।

DSG Support Multi Solution

पैरामीटर	वीबी .NET	मूल दृश्य
	अनुप्रयोग, वेब अनुप्रयोग, विंडोज अनुप्रयोग।	

VB.Net Variables

एक चर कुछ और नहीं बल्कि एक स्टोरेज एरिया को दिया गया नाम है जिसे हमारे प्रोग्राम मैनिपुलेट कर सकते हैं। VB.Net में प्रत्येक चर का एक विशिष्ट प्रकार होता है, जो चर की मेमोरी के आकार और लेआउट को निर्धारित करता है; उस मेमोरी के भीतर संग्रहीत किए जा सकने वाले मानों की सीमा; और चर पर लागू किए जा सकने वाले संचालनों का सेट।

हमने पहले ही विभिन्न डेटा प्रकारों पर चर्चा की है। VB.Net में प्रदान किए गए मूल मान प्रकारों को इस प्रकार वर्गीकृत किया जा सकता है -

Type	Example
Integral types	SByte, Byte, Short, UShort, Integer, UInteger, Long, ULong and Char
Floating point types	Single and Double
Decimal types	Decimal
Boolean types	True or False values, as assigned
Date types	Date

VB.Net Enum जैसे चर के अन्य मान प्रकारों और Class जैसे चर के संदर्भ प्रकारों को परिभाषित करने की भी अनुमति देता है। हम अगले अध्यायों में दिनांक प्रकारों और क्लासों पर चर्चा करेंगे।

VB.Net में चर घोषणा

डिम स्टेटमेंट का उपयोग एक या अधिक वैरिएबल के लिए वैरिएबल घोषणा और स्टोरेज आवंटन के लिए किया जाता है। डिम स्टेटमेंट का उपयोग मॉड्यूल, क्लास, संरचना, प्रक्रिया या ब्लॉक स्तर पर किया जाता है।

DSG Support Multi Solution

VB.Net में चर घोषणा के लिए सिंटैक्स है -

```
[ < attributelist > ] [ accessmodifier ] [[ Shared ] [ Shadows ] | [ Static ]]  
[ ReadOnly ] Dim [ WithEvents ] variablelist  
कहाँ,
```

attributelist उन विशेषताओं की सूची है जो चर पर लागू होती हैं। वैकल्पिक।

accessmodifier चरों के एक्सेस स्तरों को परिभाषित करता है, इसके मान इस प्रकार हैं - सार्वजनिक, संरक्षित, मित्र, संरक्षित मित्र और निजी। वैकल्पिक।

साझा एक साझा चर घोषित करता है, जो किसी वर्ग या संरचना के किसी विशिष्ट उदाहरण से संबद्ध नहीं होता है, बल्कि वर्ग या संरचना के सभी उदाहरणों के लिए उपलब्ध होता है। वैकल्पिक।

छाया यह संकेत देती है कि चर किसी आधार वर्ग में समान नाम वाले तत्व या अतिभारित तत्वों के समूह को पुनः घोषित करता है और छिपाता है। वैकल्पिक।

स्टैटिक यह इंगित करता है कि चर अपना मान बनाए रखेगा, भले ही वह जिस प्रक्रिया में घोषित किया गया है, उसकी समाप्ति के बाद भी। वैकल्पिक।

ReadOnly का अर्थ है कि चर को पढ़ा जा सकता है, लेकिन लिखा नहीं जा सकता। वैकल्पिक।

WithEvents निर्दिष्ट करता है कि चर का उपयोग चर को निर्दिष्ट इंस्टेंस द्वारा उठाए गए ईवेंट का जवाब देने के लिए किया जाता है। वैकल्पिक।

वेरिएबललिस्ट घोषित वेरिएबल्स की सूची प्रदान करता है।

चर सूची में प्रत्येक चर का निम्नलिखित सिंटैक्स और भाग होता है -

DSG Support Multi Solution

variablename[([boundslist])] [As [New] datatype] [= initializer]

कहाँ,

variablename – चर का नाम है

boundslist – वैकल्पिक। यह एक सरणी चर के प्रत्येक आयाम की सीमाओं की सूची प्रदान करता है।

नया – वैकल्पिक। यह Dim कथन चलाने पर क्लास का एक नया उदाहरण बनाता है।

डेटाटाइप – यदि ऑप्शन स्ट्रिक्ट चालू है तो आवश्यक है। यह चर के डेटा प्रकार को निर्दिष्ट करता है।

इनिशियलाइज़र – वैकल्पिक यदि नया निर्दिष्ट नहीं है। अभिव्यक्ति जिसका मूल्यांकन किया जाता है और चर को तब असाइन किया जाता है जब इसे बनाया जाता है।

कुछ मान्य चर घोषणाएँ उनकी परिभाषा के साथ यहाँ दर्शाई गई हैं –

```
Dim StudentID As Integer
Dim StudentName As String
Dim Salary As Double
Dim count1, count2 As Integer
Dim status As Boolean
Dim exitButton As New System.Windows.Forms.Button
Dim lastTime, nextTime As Date
```

VB.Net में चर आरंभीकरण

चरों को बराबर चिह्न के साथ आरंभीकृत किया जाता है (एक मान निर्दिष्ट किया जाता है) जिसके बाद एक स्थिर अभिव्यक्ति होती है। आरंभीकरण का सामान्य रूप है –

```
variable_name = value;
```

उदाहरण के लिए,

```
Dim pi As Double
pi = 3.14159
```

DSG Support Multi Solution

आप घोषणा के समय एक चर को निम्न प्रकार से आरंभ कर सकते हैं -

```
Dim StudentID As Integer = 100
Dim StudentName As String = "Bill Smith"
```

उदाहरण

निम्नलिखित उदाहरण को आजमाएं जो विभिन्न प्रकार के चरों का उपयोग करता है -

```
Module variablesNdatatypes
Sub Main()
    Dim a As Short
    Dim b As Integer
    Dim c As Double

    a = 10
    b = 20
    c = a + b
    Console.WriteLine("a = {0}, b = {1}, c = {2}", a, b, c)
    Console.ReadLine()
End Sub
End Module
```

जब उपरोक्त कोड संकलित और निष्पादित किया जाता है, तो यह निम्नलिखित परिणाम उत्पन्न करता है -

a = 10, b = 20, c = 30

उपयोगकर्ता से मान स्वीकार करना

सिस्टम नामस्थान में कंसोल क्लास उपयोगकर्ता से इनपुट स्वीकार करने और उसे एक चर में संग्रहीत करने के लिए एक फ़ंक्शन `ReadLine` प्रदान करता है। उदाहरण के लिए,

```
Dim message As String
message = Console.ReadLine
```

निम्नलिखित उदाहरण इसे प्रदर्शित करता है -

```
Module variablesNdatatypes
Sub Main()
    Dim message As String
    Console.Write("Enter message: ")
```

DSG Support Multi Solution

```
message = Console.ReadLine  
Console.WriteLine()  
Console.WriteLine("Your Message: {0}", message)  
Console.ReadLine()  
End Sub  
End Module
```

जब उपरोक्त कोड संकलित और निष्पादित किया जाता है, तो यह निम्नलिखित परिणाम उत्पन्न करता है (मान लें कि उपयोगकर्ता Hello World इनपुट करता है) -

```
Enter message: Hello World  
Your Message: Hello World
```

एलवैल्यू और आरवैल्यू

अभिव्यक्ति दो प्रकार की होती है -

lvalue – एक अभिव्यक्ति जो एक lvalue है, असाइनमेंट के बाएं या दाएं हाथ के रूप में दिखाई दे सकती है।

आरवैल्यू – एक अभिव्यक्ति जो आरवैल्यू है, असाइनमेंट के दाईं ओर दिखाई दे सकती है, लेकिन बाईं ओर नहीं।

वेरिएबल्स lvalues हैं और इसलिए असाइनमेंट के बाएं हाथ की ओर दिखाई दे सकते हैं। संख्यात्मक लिटरल rvalues हैं और इसलिए उन्हें असाइन नहीं किया जा सकता है और वे बाएं हाथ की ओर दिखाई नहीं दे सकते हैं। निम्नलिखित एक मान्य कथन है -

```
Dim g As Integer = 20
```

लेकिन निम्नलिखित कथन मान्य नहीं है और संकलन-समय त्रुटि उत्पन्न करेगा -

```
20 = g
```

VB.Net Data Types

कंप्यूटर में, हम पूर्णांक और स्ट्रिंग आदि के बीच अंतर करने के लिए डेटा प्रकारों का उपयोग करते हैं। डेटा प्रकार जो हम किसी चर को सौंपेंगे, वह तय करेगा कि उस चर में क्या संग्रहीत किया जाना चाहिए, यानी, चर में संग्रहीत किए जाने वाले मान चर के डेटा प्रकार पर निर्भर करते हैं। VBA में, डेटा प्रकारों को दो प्रमुख प्रकारों में विभाजित किया जाता है:

1. संख्यात्मक डेटा प्रकार: संख्यात्मक डेटा प्रकारों का उपयोग गणितीय संचालन जैसे जोड़, घटाव आदि करने के लिए किया जाता है। इसका उपयोग विभिन्न प्रतिनिधित्व प्रारूप में संख्याओं को संभालने के लिए

DSG Support Multi Solution

किया जाता है। संख्यात्मक डेटा प्रकार में, इंटीग्रल प्रकार केवल पूर्ण संख्याओं (शून्य, सकारात्मक और नकारात्मक) का प्रतिनिधित्व करता है। गैर-इंटीग्रल प्रकार पूर्णांक और भिन्नात्मक भाग दोनों का प्रतिनिधित्व करते हैं।

Data Types	Memory Size	Value Range
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	Negative Values (-3.402823E+38 to -1.401298E-45) & Positive Values (1.401298E-45 to 3.402823E+38)
Double	8 bytes	Negative Values (-1.79769313486232e+308 to -4.94065645841247E-324) & Positive Values (4.94065645841247E-324 to 1.79769313486232e+308)
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Decimal	12 bytes	No Decimal Places (+/- 79,228,162,514,264,337,593,543,950,335) & Up to 28 Decimal Places (+/- 7.9228162514264337593543950335)

2. नॉन-न्यूमेरिक डेटा टाइप: नॉन-न्यूमेरिक डेटा टाइप को अंकगणितीय ऑपरेटरों द्वारा नियंत्रित नहीं किया जाता है। इनमें टेक्स्ट, डेटा आदि शामिल होते हैं।

Data Types	Memory Size	Value Range
String(fixed size/length)	Equivalent to String's length(in bytes)	1 to 65,400 characters
String(variable length)	String's length + 10 bytes	0 to 2 billion characters
Boolean	2 bytes	True/False
Object	4 bytes	Embedded object

DSG Support Multi Solution

Data	8 bytes	January 1, 100 to December 31, 9999
Variant(numeric)	16 bytes	Any value
Variant (text)	Text's length + 22 bytes	0 to 2 billion characters

2. Reference Types (संदर्भ प्रकार)

ये वैल्यू को सीधे स्टोर नहीं करते बल्कि मेमोरी के किसी स्थान का पता (reference) स्टोर करते हैं।

डाटा टाइप

विवरण

String	टेक्स्ट स्टोर करने के लिए, जैसे नाम, पता
Object	सभी डाटा टाइप्स का बेस क्लास होता है
Array	एक जैसे टाइप के कई वैल्यू को संग्रहित करता है
Class	कस्टम ऑब्जेक्ट बनाने के लिए
Interface	इंटरफेस डिफाइन करता है

◆ 3. Type Conversion (प्रकार रूपांतरण)

VB.NET में डाटा टाइप्स को बदला जा सकता है:

- **Implicit Conversion (स्वतः रूपांतरण)** – छोटे से बड़े टाइप में
- **Explicit Conversion (स्पष्ट रूपांतरण)** – बड़े से छोटे टाइप में (CType, Convert, आदि का उपयोग)

```
vb.net
CopyEdit
Dim x As Integer = 10
Dim y As Double = x      ' Implicit conversion
```

DSG Support Multi Solution

Dim z As Integer = CType(y, Integer) ' Explicit conversion

◆ 4. Type Inference (प्रकार अनुमान)

VB.NET में आप Dim के साथ वैरिएबल घोषित करके टाइप को स्वतः अनुमानित कर सकते हैं:

```
vb.net
CopyEdit
Dim name = "Rahul" ' VB.NET इसे String मानेगा
Dim age = 25 ' VB.NET इसे Integer मानेगा
```

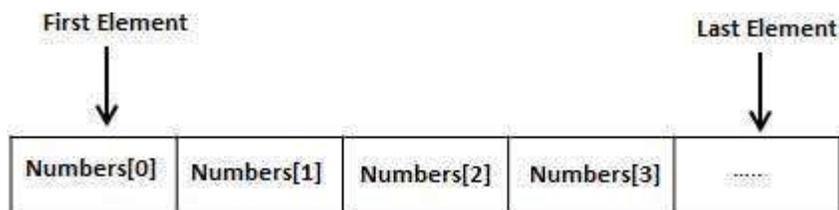
◆ 5. Nullable Types (शून्य-संभव प्रकार)

कुछ वैल्यू टाइप्स में null स्टोर करने के लिए ? या Nullable का उपयोग करते हैं:

```
vb.net
CopyEdit
Dim age As Nullable(Of Integer) = Nothing
' या
Dim age As Integer? = Nothing
```

Array

ऐरे एक रैखिक डेटा संरचना है जहाँ सभी तत्व क्रमिक रूप से व्यवस्थित होते हैं। यह एक ही डेटा प्रकार के तत्वों का संग्रह है जो सन्नहित मेमोरी स्थानों पर संग्रहीत होता है।



सरलता के लिए, हम एक सरणी को सीढ़ियों की एक उड़ान के रूप में सोच सकते हैं जहाँ प्रत्येक चरण पर एक मान रखा जाता है (मान लीजिए कि आपका कोई मित्र)। यहाँ, आप अपने किसी भी मित्र के स्थान की पहचान केवल उस चरण की गिनती जानकर कर सकते हैं जिस पर वे हैं।

DSG Support Multi Solution

इससे प्रत्येक तत्व की स्थिति की गणना करना आसान हो जाता है, बस एक बेस वैल्यू में ऑफसेट जोड़कर, यानी, सरणी के पहले तत्व का मेमोरी स्थान (आमतौर पर सरणी के नाम से दर्शाया जाता है)। बेस वैल्यू इंडेक्स 0 है और दो इंडेक्स के बीच का अंतर ऑफसेट है।

क्या सरणी सदैव एक निश्चित आकार की होती है?

मूल रूप से ऐरे केवल निश्चित आकार के होते हैं, लेकिन अधिकांश भाषाएँ अंतर्निहित निश्चित आकार के ऐरे का उपयोग करके गतिशील आकार के ऐरे प्रदान करती हैं। उदाहरण के लिए, C++ में वेक्टर, जावा में ArrayList और पायथन में सूची। C भाषा में, ऐरे का एक निश्चित आकार होता है जिसका अर्थ है कि एक बार आकार दिए जाने के बाद, इसे बदला नहीं जा सकता है यानी आप इसे छोटा नहीं कर सकते हैं और न ही आप इसे बड़ा कर सकते हैं।

There are majorly 4 types of arrays

1. Fixed Size Array
2. Dynamic Sized Array
3. 1-Dimensional Array
4. Multi-Dimensional Array

Classification of Types of Arrays

However, these array types can be broadly classified in two ways:

1. On the basis of Size
2. On the basis of Dimensions

Dynamic Arrays

Dynamic arrays are arrays that can be dimensioned and re-dimensioned as per the need of the program. You can declare a dynamic array using the **ReDim** statement.

Syntax for ReDim statement –

```
ReDim [Preserve] arrayname(subscripts)
```

Where,

The **Preserve** keyword helps to preserve the data in an existing array, when you resize it.

arrayname is the name of the array to re-dimension.

subscripts specifies the new dimension.

```
Module arrayApl
Sub Main()
    Dim marks() As Integer
    ReDim marks(2)
```

DSG Support Multi Solution

```
marks(0) = 85
```

```
marks(1) = 75
```

```
marks(2) = 90
```

```
ReDim Preserve marks(10)
```

```
marks(3) = 80
```

```
marks(4) = 76
```

```
marks(5) = 92
```

```
marks(6) = 99
```

```
marks(7) = 79
```

```
marks(8) = 75
```

```
For i = 0 To 10
```

```
    Console.WriteLine(i & vbTab & marks(i))
```

```
Next i
```

```
Console.ReadKey()
```

```
End Sub
```

```
End Module
```

When the above code is compiled and executed, it produces the following result –

```
0      85
1      75
2      90
3      80
4      76
5      92
6      99
7      79
8      75
9      0
10     0
```

Multi-Dimensional Arrays

VB.Net allows multidimensional arrays. Multidimensional arrays are also called rectangular arrays.

You can declare a 2-dimensional array of strings as –

```
Dim twoDStringArray(10, 20) As String
```

DSG Support Multi Solution

or, a 3-dimensional array of Integer variables –

```
Dim threeDIntArray(10, 10, 10) As Integer
```

The following program demonstrates creating and using a 2-dimensional array –

```
Module arrayApl
    Sub Main()
        ' an array with 5 rows and 2 columns
        Dim a(,) As Integer = {{0, 0}, {1, 2}, {2, 4}, {3, 6}, {4, 8}}
        Dim i, j As Integer
        ' output each array element's value '

        For i = 0 To 4
            For j = 0 To 1
                Console.WriteLine("a[{0},{1}] = {2}", i, j, a(i, j))
            Next j
        Next i
        Console.ReadKey()
    End Sub
End Module
```

When the above code is compiled and executed, it produces the following result –

```
a[0,0]: 0
a[0,1]: 0
a[1,0]: 1
a[1,1]: 2
a[2,0]: 2
a[2,1]: 4
a[3,0]: 3
a[3,1]: 6
a[4,0]: 4
a[4,1]: 8
```

Jagged Array

A Jagged array is an array of arrays. The following code shows declaring a jagged array named `scores` of Integers –

```
Dim scores As Integer()() = New Integer(5)(){}
```

DSG Support Multi Solution

The following example illustrates using a jagged array –

```
Module arrayApl
Sub Main()
    'a jagged array of 5 array of integers
    Dim a As Integer()() = New Integer(4)() {}
    a(0) = New Integer() {0, 0}
    a(1) = New Integer() {1, 2}
    a(2) = New Integer() {2, 4}
    a(3) = New Integer() {3, 6}
    a(4) = New Integer() {4, 8}
    Dim i, j As Integer
    ' output each array element's value

    For i = 0 To 4
        For j = 0 To 1
            Console.WriteLine("a[{0},{1}] = {2}", i, j, a(i)(j))
        Next j
    Next i
    Console.ReadKey()
End Sub
End Module
```

When the above code is compiled and executed, it produces the following result –

```
a[0][0]: 0
a[0][1]: 0
a[1][0]: 1
a[1][1]: 2
a[2][0]: 2
a[2][1]: 4
a[3][0]: 3
a[3][1]: 6
a[4][0]: 4
a[4][1]: 8
```

The Array Class

The Array class is the base class for all the arrays in VB.Net. It is defined in the System namespace. The Array class provides various properties and methods to work with arrays.

DSG Support Multi Solution

Properties of the Array Class

The following table provides some of the most commonly used **properties** of the **Array** class –

Sr.No	Property Name & Description
1	IsFixedSize Gets a value indicating whether the Array has a fixed size.
2	IsReadOnly Gets a value indicating whether the Array is read-only.
3	Length Gets a 32-bit integer that represents the total number of elements in all the dimensions of the Array.
4	LongLength Gets a 64-bit integer that represents the total number of elements in all the dimensions of the Array.
5	Rank Gets the rank (number of dimensions) of the Array.

VB.Net Function

A procedure is a collection of statements that, when called, complete a task. After the procedure is completed, control returns to the statement that is called the procedure. There are two kinds of procedures in VB.Net –

- Functions- return a value
- Sub procedures or Subs- do not return a value

Defining a Function

The Function statement is used to declare a function's name, parameters, and body. The Function statement has the following syntax–

```
[Modifiers] Function FunctionName [(ParameterList)] As ReturnType
```

```
[Statements]
```

```
End Function
```

DSG Support Multi Solution

Where,

- **Modifiers** – specify the function’s access level; possible values are Public, Private, Friend, Protected, Protected Friend and information regarding overloading, sharing, overriding, and shadowing.
- **FunctionName** – indicates the name of the function
- **ParameterList** – used to specify a list of the parameters
- **ReturnType** – used to specify the data type of the variable that the function returns

Function Return a Value

A function in VB.Net can return a value to the caller code in two ways:

- by using the return statement
- by assigning the value to the function name

Recursive Function

A recursive function can call itself. This is called recursion.

Param Arrays

When declaring a function or subroutine, you may be unsure of the number of arguments supplied as parameters. At this point, VB.Net param arrays (or parameter arrays) come in handy.

An array can be passed as a function argument in VB.Net.

VB.Net Statements

Introduction to Conditional Statements

Conditional statements are fundamental constructs in programming that allow developers to execute different blocks of code based on certain conditions. In VB.NET, conditional statements enable the creation of dynamic and responsive applications by making decisions at runtime. Understanding how to effectively use these statements is essential for controlling the flow of execution and implementing logic in your programs.

DSG Support Multi Solution

If...Then...Else Statement

The **If...Then...Else** statement is the most basic form of conditional logic in VB.NET. It evaluates a condition and executes a block of code if the condition is true. Optionally, it can execute another block of code if the condition is false.

```
Module IfThenElseExample
Sub Main()
    Dim score As Integer = 85

    If score >= 90 Then
        Console.WriteLine("Grade: A")
    ElseIf score >= 80 Then
        Console.WriteLine("Grade: B")
    ElseIf score >= 70 Then
        Console.WriteLine("Grade: C")
    Else
        Console.WriteLine("Grade: F")
    End If
End Sub
End Module
VB.Net
Copy
```

Grade: B

Explanation: In this example, the program evaluates the value of `score`. Since 85 is greater than or equal to 80 but less than 90, it prints "Grade: B".

Select Case Statement

The **Select Case** statement provides a cleaner and more readable alternative to multiple If...Then...Else statements when evaluating a single expression against multiple conditions.

```
Module SelectCaseExample
Sub Main()
    Dim day As String = "Wednesday"

    Select Case day
        Case "Monday"
            Console.WriteLine("Start of the work week.")
        Case "Wednesday"
            Console.WriteLine("Midweek progress.")
        Case "Friday"
            Console.WriteLine("End of the work week.")
        Case Else
            Console.WriteLine("It's the weekend!")
    End Select
End Sub
End Module
VB.Net
```

DSG Support Multi Solution

Copy

```
Midweek progress.
```

Explanation: The `Select Case` statement evaluates the `day` variable. Since `day` is "Wednesday", it matches the second case and prints "Midweek progress."

Nested Conditional Statements

Nested conditional statements are **If** or **Select Case** statements placed within another conditional statement. They allow for more complex decision-making by evaluating additional conditions based on previous results.

```
Module NestedConditionsExample
Sub Main()
    Dim age As Integer = 25
    Dim hasLicense As Boolean = True

    If age >= 18 Then
        If hasLicense Then
            Console.WriteLine("Eligible to drive.")
        Else
            Console.WriteLine("You need a driver's license to drive.")
        End If
    Else
        Console.WriteLine("You must be at least 18 years old to drive.")
    End If
End Sub
End Module
VB.Net
Copy
```

```
Eligible to drive.
```

Explanation: The outer `If` statement checks if `age` is 18 or older. Since `age` is 25, it proceeds to the nested `If` statement, which checks if `hasLicense` is `True`. Both conditions are met, so it prints "Eligible to drive."

Boolean Logic in Conditions

Boolean logic allows for the combination of multiple conditions using logical operators. In VB.NET, common logical operators include `And`, `Or`, `Not`, `AndAlso`, and `OrElse`.

```
Module BooleanLogicExample
Sub Main()
    Dim isMember As Boolean = True
    Dim purchaseAmount As Decimal = 150D
```

DSG Support Multi Solution

```
If isMember AndAlso purchaseAmount > 100D Then
    Console.WriteLine("Eligible for a discount.")
Else
    Console.WriteLine("No discount available.")
End If
End Sub
End Module
VB.Net
```

Copy

Eligible for a discount.

Windows Form

विंडोज फॉर्म (WinForms) विंडोज डेस्कटॉप एप्लिकेशन बनाने के लिए एक GUI क्लास लाइब्रेरी है। मूल रूप से .NET फ्रेमवर्क का हिस्सा, यह अब **.NET Core**, **.NET 5+**, **.NET 8 (LTS)** और **.NET 9** में समर्थित है। इसका मुख्य उद्देश्य डेस्कटॉप, टैबलेट और पीसी के लिए एप्लिकेशन विकसित करने के लिए एक आसान इंटरफ़ेस प्रदान करना है। इसे **WinForms** भी कहा जाता है।

विंडोज फॉर्म या विनफॉर्म का उपयोग करके विकसित किए गए एप्लिकेशन को **विंडोज फॉर्म एप्लिकेशन** के रूप में जाना जाता है जो डेस्कटॉप कंप्यूटर पर चलते हैं। विनफॉर्म का उपयोग केवल विंडोज फॉर्म एप्लिकेशन विकसित करने के लिए किया जा सकता है।

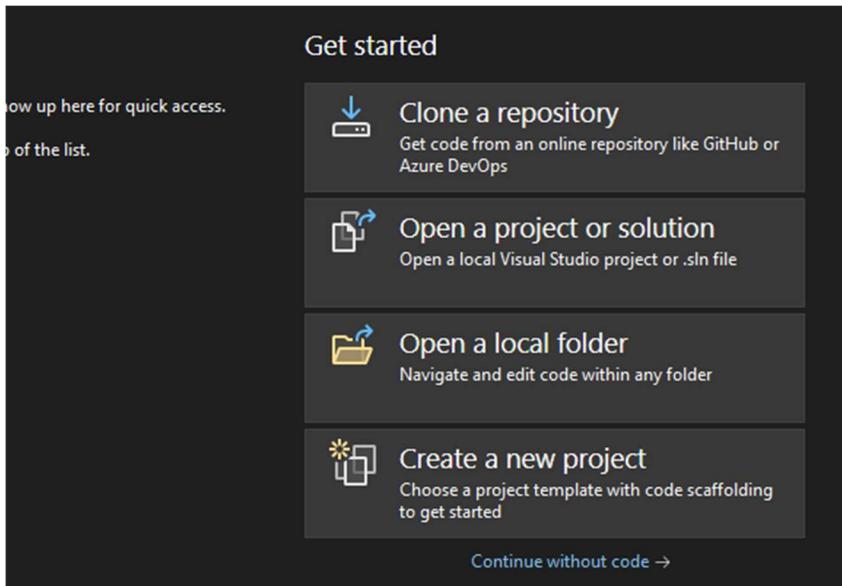
पहला विंडोज फॉर्म बनाना

आइए पहला विंडोज फॉर्म बनाएं, यहां हम Visual Studio Community 2022 का उपयोग कर रहे हैं। इस लेख को देखें जिसमें [Visual Studio को डाउनलोड करने](#) की पूरी प्रक्रिया है।

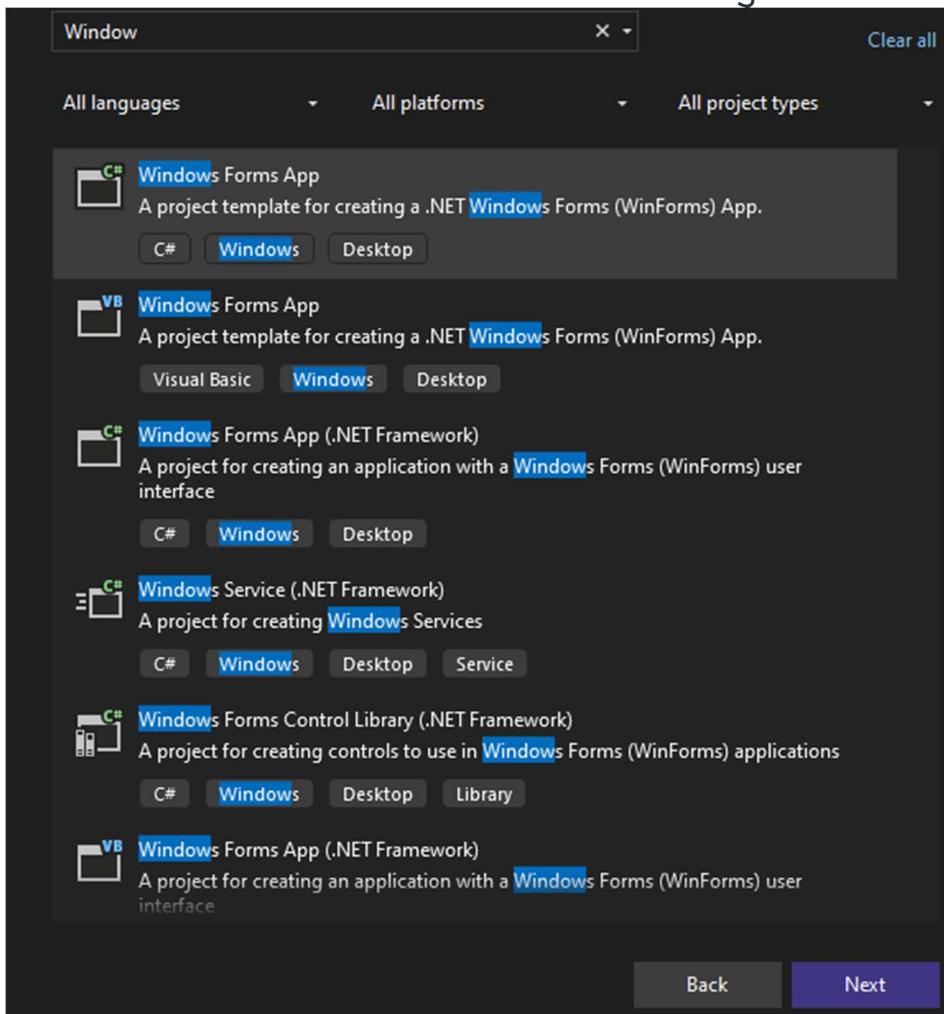
सबसे पहले हम एक साधारण फॉर्म बनाते हैं जिसमें एक बटन होता है और जब हम उस बटन पर क्लिक करते हैं तो यह एक संदेश दिखाता है कि इसे बनाएं। विंडोज फॉर्म बनाने के लिए ये महत्वपूर्ण चरण हैं, नीचे बताए गए चरणों का पालन करें:

चरण 1: अब विजुअल स्टूडियो खोलें और वहां विभिन्न विकल्प उपलब्ध हैं।

DSG Support Multi Solution

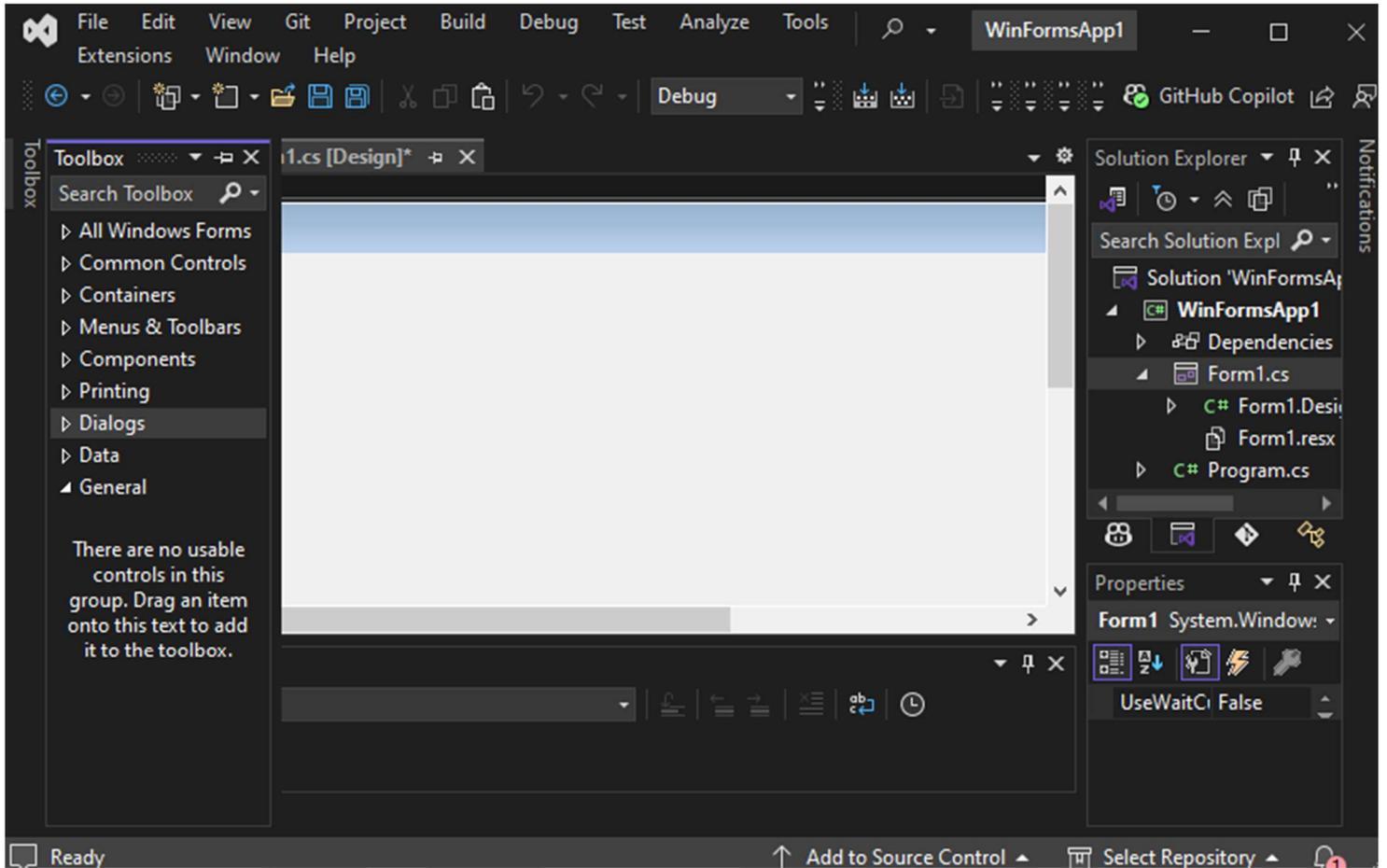


चरण 2 : अब C# के साथ Windows Forms ऐप चुनें और अगले बटन पर क्लिक करें।



DSG Support Multi Solution

चरण 3: अब नाम के साथ प्रोजेक्ट का पता लगाएं, यहां हम डिफॉल्ट नाम का उपयोग कर रहे हैं जो कि फॉर्म 1 है और यह संपादक में एक फॉर्म खोलेगा जिसे हम आगे संशोधित कर सकते हैं।



आइए Windows फॉर्म के लिए Visual Studio Community 2022 के इंटरफ़ेस को समझें

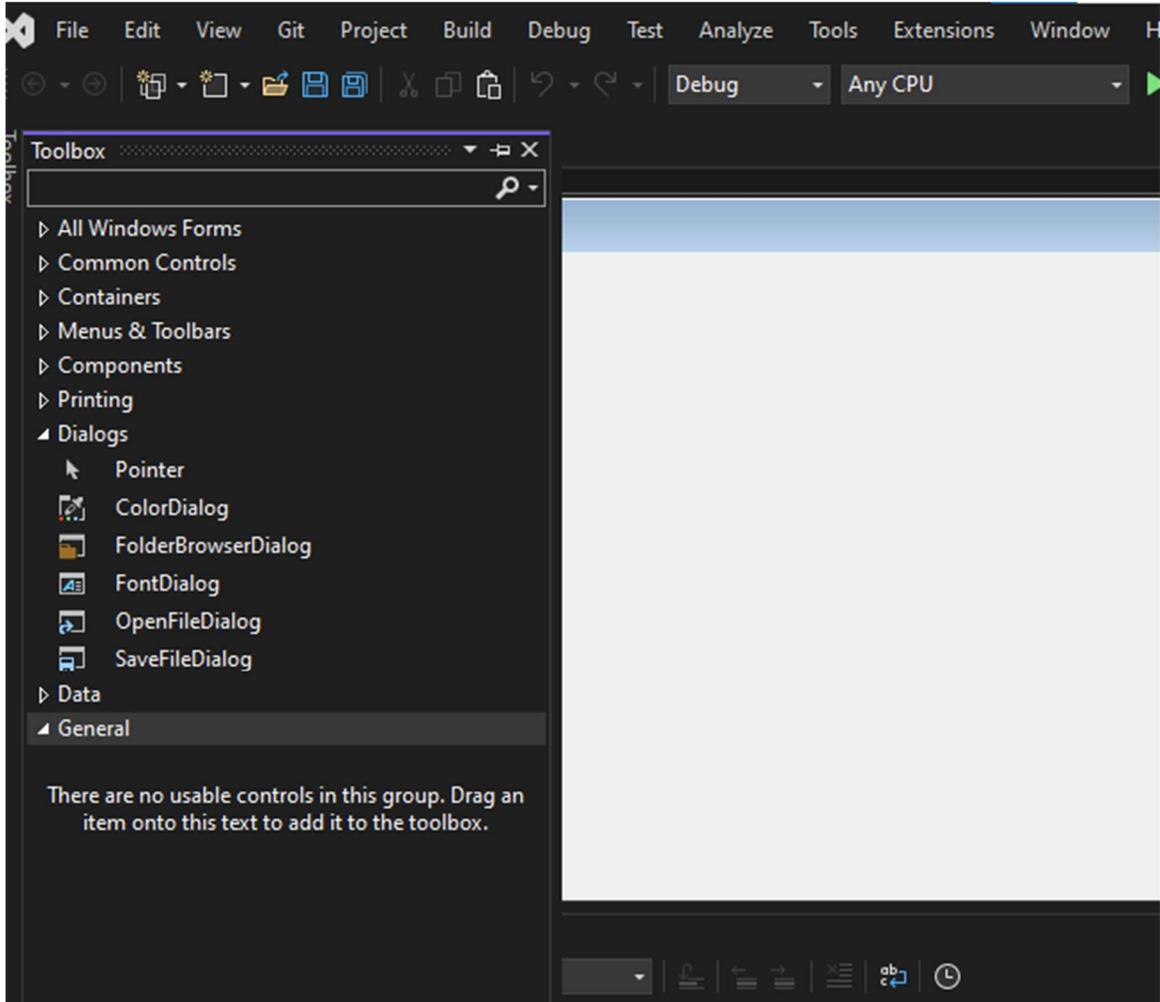
- **एडिटर विंडो (मुख्य विंडो):** यहाँ, हम फॉर्म और कोड संपादन के साथ काम करेंगे। फॉर्म के लेआउट पर ध्यान दें जो अब खाली है। अगर हम फॉर्म पर डबल-क्लिक करते हैं तो यह उसके लिए कोड खोल देगा।
- **समाधान एक्सप्लोरर :** आपके प्रोजेक्ट के लिए एक फ़ाइल प्रबंधक जहाँ आप फॉर्म, संसाधन और सेटिंग्स के बीच नेविगेट कर सकते हैं। उदाहरण के लिए, यदि हम इस विंडो से कोई फ़ाइल चुनेंगे तो प्रॉपर्टी विंडो में विशेष जानकारी प्रदर्शित होगी।
- **गुण विंडो:** आपको रंग, टेक्स्ट संरेखण और दृश्यता जैसे UI तत्वों को संशोधित करने की अनुमति देता है। साथ ही, हम उन घटकों या नियंत्रणों के गुणों को बदल सकते हैं जो फॉर्म में जोड़े जाएँगे।

छवि में, हमारे पास दो खुली हुई फ़ाइलें हैं, एक डिज़ाइन और दूसरी **Form1.cs**, ये दोनों एक प्रमुख भूमिका निभाते हैं। हम कस्टम लॉजिक के लिए Form 1.cs फ़ाइल का उपयोग करते हैं।

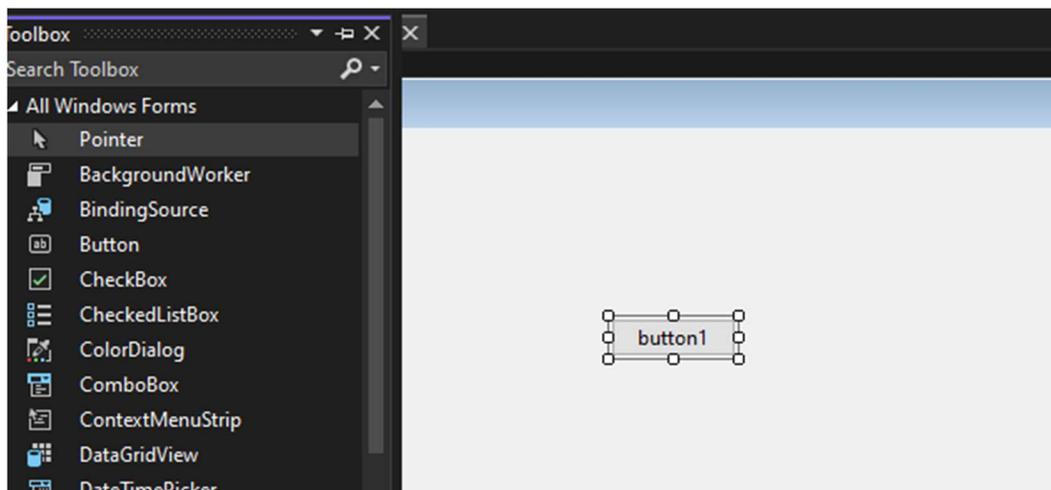
चरण 4: अब एक टूलबॉक्स खोलें, दृश्य > टूलबॉक्स या **ctrl + alt + x** पर जाएं ।

DSG Support Multi Solution

टूलबॉक्स: UI डिज़ाइन करने के लिए बटन, टेक्स्ट बॉक्स और लेबल जैसे ड्रैग-एंड-ड्रॉप नियंत्रण प्रदान करता है।

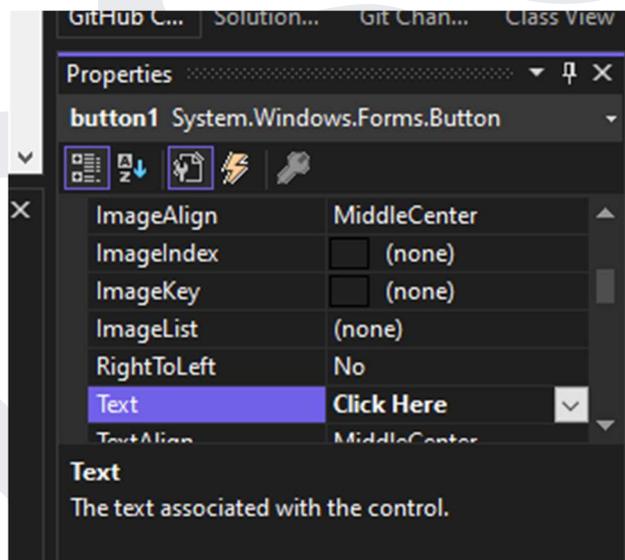
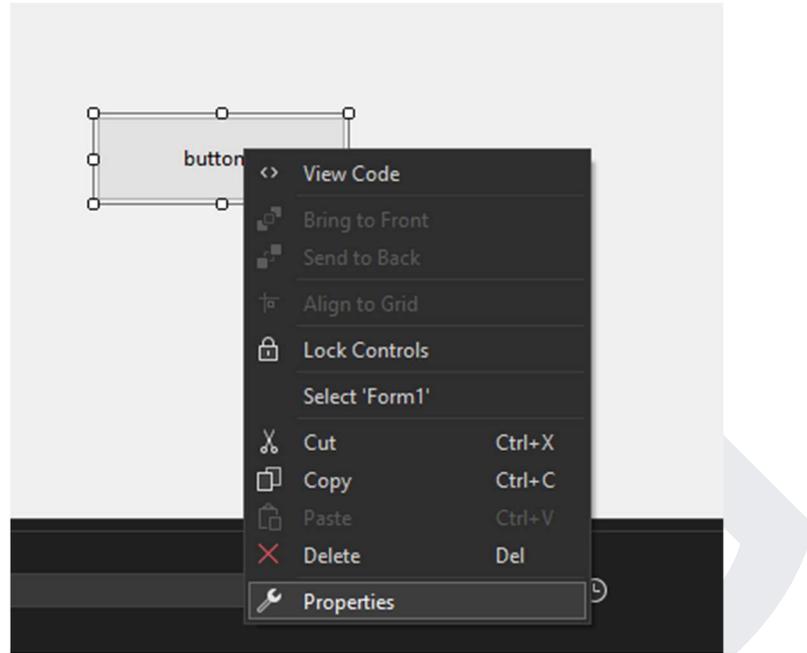


चरण 5. अब सामान्य नियंत्रण खोलें और बटन को फॉर्म पर उस स्थान पर खींचें और छोड़ें जहां हम इसे रखना चाहते हैं।



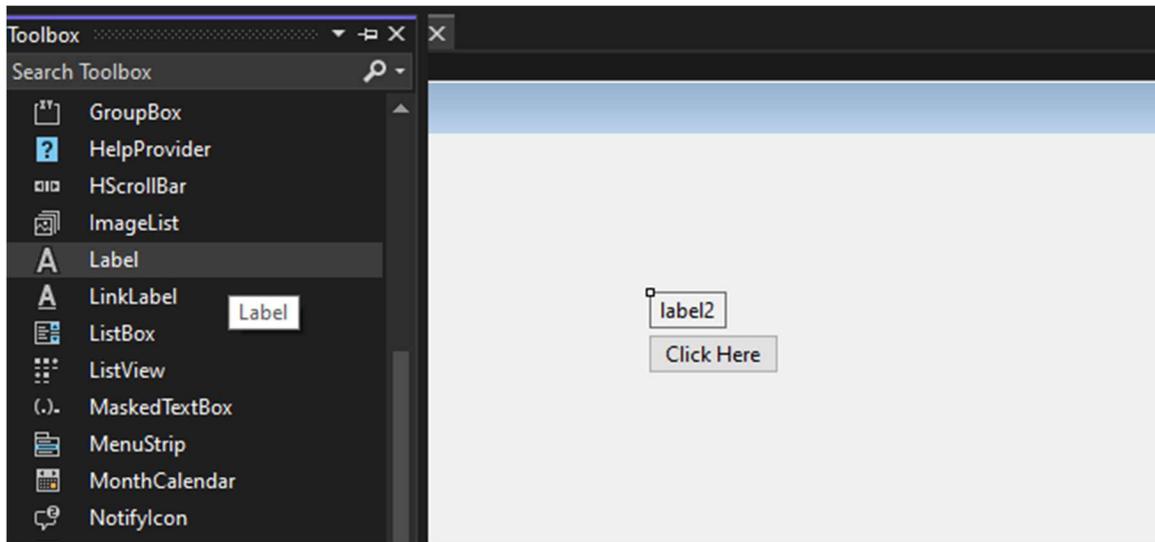
DSG Support Multi Solution

चरण 6. अब बटन के गुण खोलें, बटन पर राइट-क्लिक करें और यह गुण समाधान एक्सप्लोरर खोल देगा अब हम बटन का नाम बदलकर यहां क्लिक करें कर सकते हैं।



चरण 7: अब फिर से टूलबॉक्स खोलें और फिर लेबल पर क्लिक करें और उस स्थान पर रखें जहां हम संदेश दिखाना चाहते हैं।

DSG Support Multi Solution



नोट: Form1.Designer.cs फ़ाइल स्वचालित रूप से Windows Forms के लिए UI कोड जेनरेट करती है। आपको इस फ़ाइल को मैनुअल रूप से संपादित नहीं करना चाहिए, क्योंकि हो सकता है कि इसमें किए गए बदलाव खो जाएँ। इसके बजाय, Form1.cs में इवेंट-हैंडलिंग लॉजिक जोड़ें

नोट: उपयोगकर्ता इंटरैक्शन को संभालने के लिए, Form1.cs में क्लिक इवेंट जेनरेट करने के लिए बटन पर डबल-क्लिक करें। यहां, आप बटन क्लिक होने पर क्रियाएँ परिभाषित कर सकते हैं।

चरण 8. अब फिर से बटन पर डबल क्लिक करें और यह Form1.cs फ़ाइल खोलेगा और बटन 1 क्लिक क्रिया ट्रिगर होने पर पाठ जोड़ देगा।

```
InitializeComponent(),
}

1 reference
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello Geeks!");
}

0 references
private void button2_Click(object sender, EventArgs e)
```

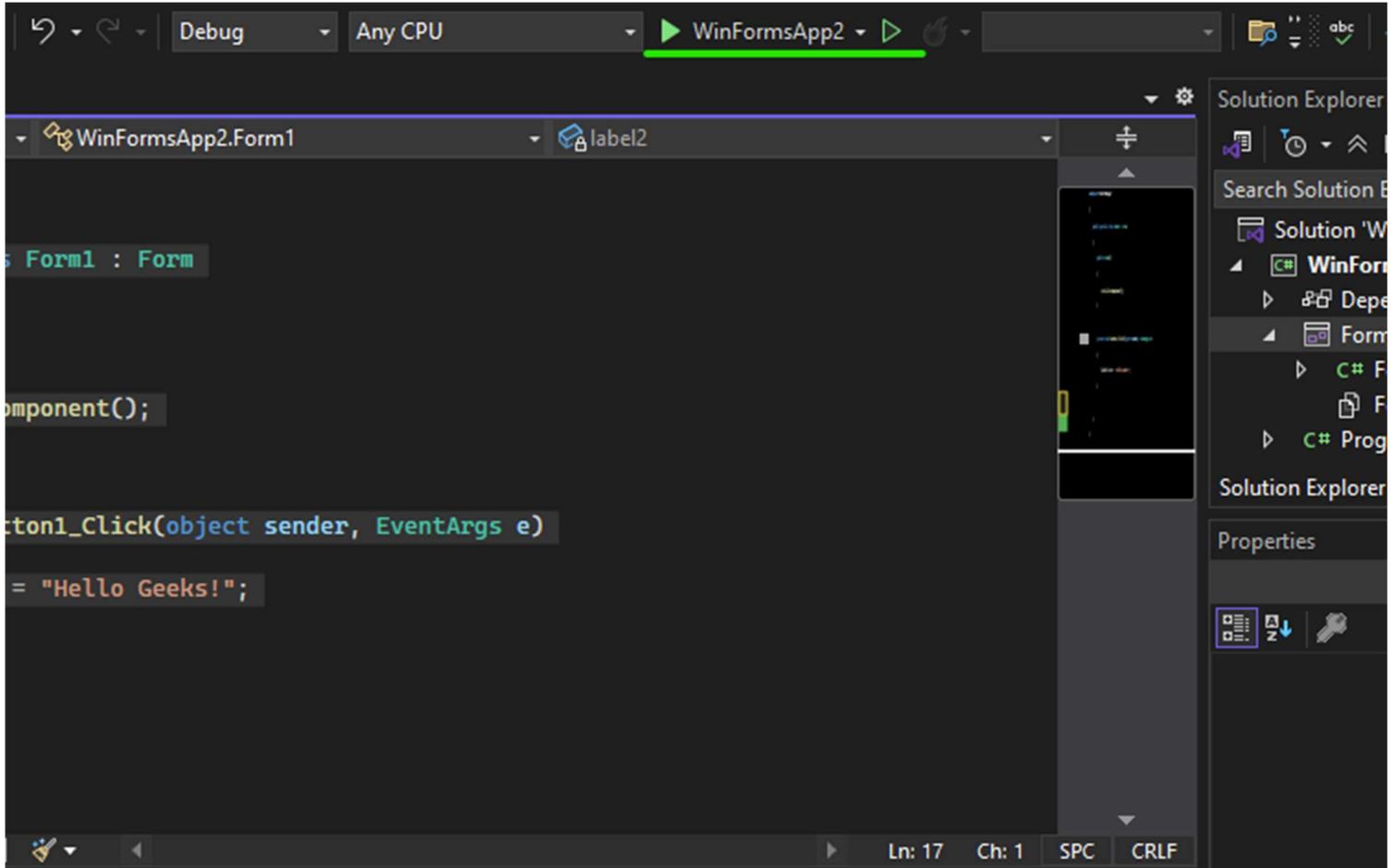
Form1.cs फ़ाइल:

```
namespace WinFormsApp2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            label1.Text = "Hello Geeks!";
        }
    }
}
```

DSG Support Multi Solution

चरण 9: अब प्रोग्राम को निष्पादित करने के लिए स्टार्ट बटन पर क्लिक करें या **F5** दबाएं।



अब यह प्रोग्राम के साथ इंटरैक्ट करने के लिए एक विंडोज फॉर्म खोलेगा।

OOPS

जैसा कि नाम से पता चलता है, ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग या OOPs उन भाषाओं को संदर्भित करता है जो प्रोग्रामिंग में ऑब्जेक्ट का उपयोग करते हैं। ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग का उद्देश्य प्रोग्रामिंग में इनहेरिटेन्स, हिडिंग, पॉलीमॉर्फिज्म आदि जैसी वास्तविक दुनिया की संस्थाओं को लागू करना है। OOP का मुख्य उद्देश्य डेटा और उन पर काम करने वाले फ़ंक्शन को एक साथ बांधना है ताकि कोड का कोई अन्य भाग उस फ़ंक्शन को छोड़कर इस डेटा तक न पहुँच सके।

ओ.ओ.पी.एस. अवधारणाएँ:

- कक्षा

DSG Support Multi Solution

- वस्तुओं
- डेटा अमूर्तन
- कैप्सूलीकरण
- विरासत
- बहुरूपता
- गतिशील बाइंडिंग
- संदेश भेजना

1. वर्ग:

क्लास एक उपयोगकर्ता-परिभाषित डेटा प्रकार है। इसमें डेटा सदस्य और सदस्य फ़ंक्शन शामिल होते हैं, जिन्हें उस क्लास का उदाहरण बनाकर एक्सेस और उपयोग किया जा सकता है। यह उन गुणों या विधियों के समूह का प्रतिनिधित्व करता है जो एक प्रकार के सभी ऑब्जेक्ट के लिए सामान्य हैं। एक क्लास किसी ऑब्जेक्ट के लिए एक ब्लूप्रिंट की तरह है।

उदाहरण के लिए: कारों के वर्ग पर विचार करें। अलग-अलग नाम और ब्रांड वाली कई कारें हो सकती हैं, लेकिन उन सभी में कुछ सामान्य गुण होंगे जैसे कि उन सभी में 4 पहिए, गति सीमा, माइलेज रेंज आदि होंगे। तो यहाँ, कार वर्ग है, और पहिए, गति सीमा, माइलेज उनके गुण हैं।

2. वस्तु:

यह ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग की एक बुनियादी इकाई है और वास्तविक जीवन की संस्थाओं का प्रतिनिधित्व करती है। एक ऑब्जेक्ट एक क्लास का उदाहरण है। जब एक क्लास परिभाषित की जाती है, तो कोई मेमोरी आवंटित नहीं की जाती है, लेकिन जब इसे इंस्टैंटिएट किया जाता है (यानी एक ऑब्जेक्ट बनाया जाता है) तो मेमोरी आवंटित की जाती है। एक ऑब्जेक्ट की एक पहचान, स्थिति और व्यवहार होता है। प्रत्येक ऑब्जेक्ट में डेटा और कोड होता है जिससे डेटा में हेरफेर किया जा सकता है। ऑब्जेक्ट एक दूसरे के डेटा या कोड के विवरण को जाने बिना भी बातचीत कर सकते हैं, यह जानना पर्याप्त है कि ऑब्जेक्ट द्वारा स्वीकार किए गए संदेश का प्रकार और प्रतिक्रिया का प्रकार क्या है।

उदाहरण के लिए "कुत्ता" एक वास्तविक वस्तु है, जिसमें रंग, नस्ल, भौंकना, सोना और खाना जैसी कुछ विशेषताएं हैं।



वस्तु

DSG Support Multi Solution

3. डेटा अमूर्तन:

डेटा एब्सट्रैक्शन ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग की सबसे ज़रूरी और महत्वपूर्ण विशेषताओं में से एक है। डेटा एब्सट्रैक्शन का मतलब है बाहरी दुनिया को डेटा के बारे में सिर्फ़ ज़रूरी जानकारी देना, बैकग्राउंड डिटेल या इम्प्लीमेंटेशन को छिपाना। एक असल ज़िंदगी के उदाहरण पर विचार करें जिसमें एक आदमी कार चला रहा है। आदमी सिर्फ़ इतना जानता है कि एक्सीलेटर दबाने से कार की रफ़्तार बढ़ जाएगी या ब्रेक लगाने से कार रुक जाएगी, लेकिन उसे यह नहीं पता कि एक्सीलेटर दबाने पर कार की रफ़्तार कैसे बढ़ रही है, उसे कार के अंदरूनी मैकेनिज्म या कार में एक्सीलेटर, ब्रेक आदि के इम्प्लीमेंटेशन के बारे में नहीं पता। यही एब्सट्रैक्शन है।

4. एनकैप्सुलेशन:

एनकैप्सुलेशन को डेटा को एक ही इकाई के अंतर्गत लपेटने के रूप में परिभाषित किया जाता है। यह वह तंत्र है जो कोड और उस डेटा को एक साथ बांधता है जिसे वह हेरफेर करता है। एनकैप्सुलेशन में, किसी क्लास के वैरिएबल या डेटा को किसी अन्य क्लास से छिपाया जाता है और उन्हें केवल उस क्लास के किसी भी सदस्य फ़ंक्शन के माध्यम से एक्सेस किया जा सकता है जिसमें उन्हें घोषित किया गया है। एनकैप्सुलेशन की तरह, क्लास में डेटा अन्य क्लास से छिपा होता है, इसलिए इसे **डेटा-हाइडिंग** के रूप में भी जाना जाता है।

Encapsulation



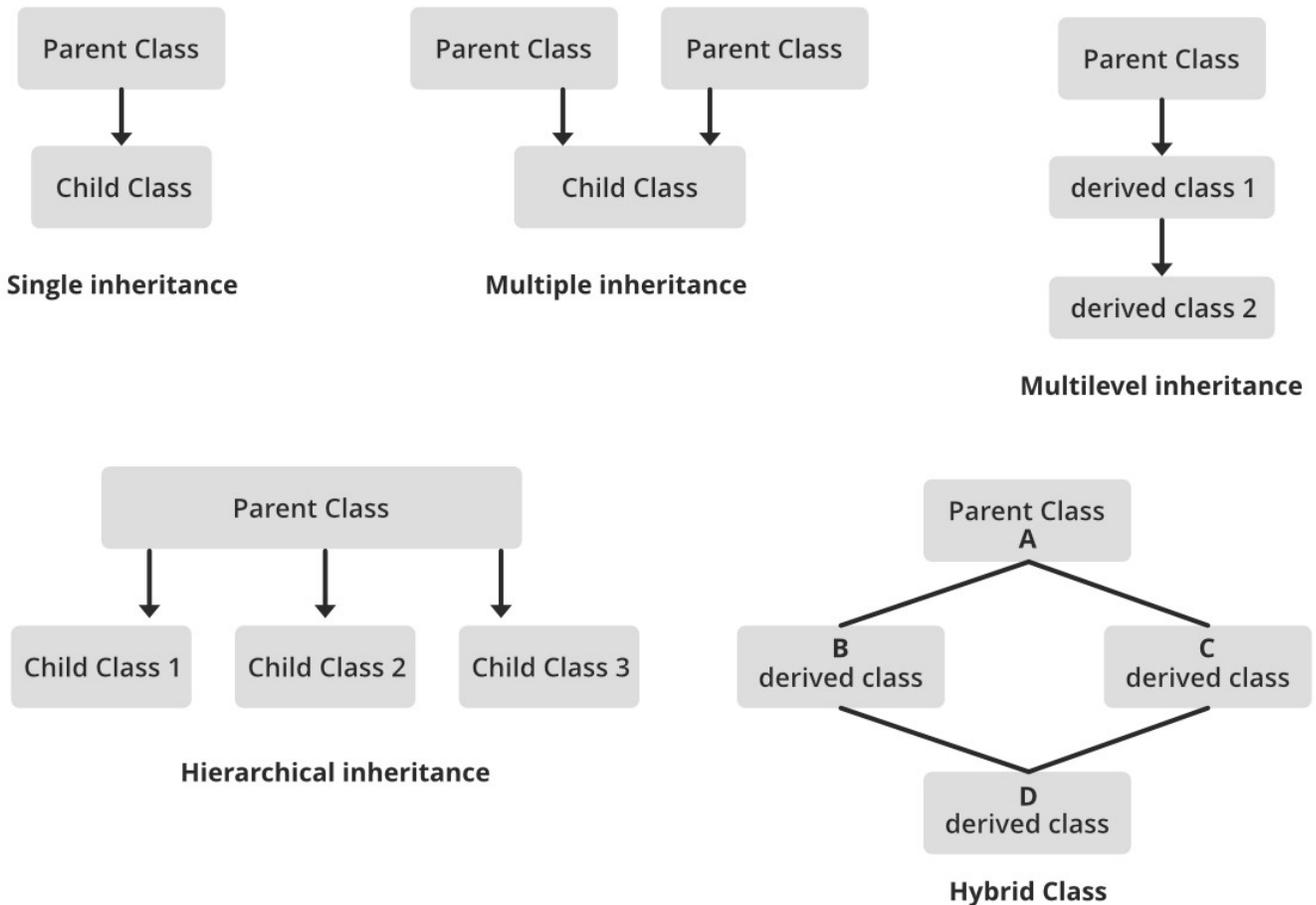
एनकैप्सुलेशन का एक वास्तविक उदाहरण लें, एक कंपनी में अलग-अलग सेक्शन होते हैं जैसे अकाउंट्स सेक्शन, फाइनेंस सेक्शन, सेल्स सेक्शन आदि। फाइनेंस सेक्शन सभी वित्तीय लेन-देन को संभालता है और फाइनेंस से जुड़े सभी डेटा का रिकॉर्ड रखता है। इसी तरह सेल्स सेक्शन सभी बिक्री से जुड़ी गतिविधियों को

DSG Support Multi Solution

संभालता है और सभी बिक्री का रिकॉर्ड रखता है। अब ऐसी स्थिति आ सकती है जब किसी कारण से फाइनेंस सेक्शन के किसी अधिकारी को किसी विशेष महीने में बिक्री के बारे में सभी डेटा की आवश्यकता हो। इस मामले में, उसे सेल्स सेक्शन के डेटा तक सीधे पहुंचने की अनुमति नहीं है। उसे पहले सेल्स सेक्शन के किसी अन्य अधिकारी से संपर्क करना होगा और फिर उससे विशेष डेटा देने का अनुरोध करना होगा। एनकैप्सुलेशन यही है। यहां सेल्स सेक्शन का डेटा और उन्हें मैनिपुलेट करने वाले कर्मचारी एक ही नाम "सेल्स सेक्शन" के तहत लपेटे जाते हैं।

5. उत्तराधिकार:

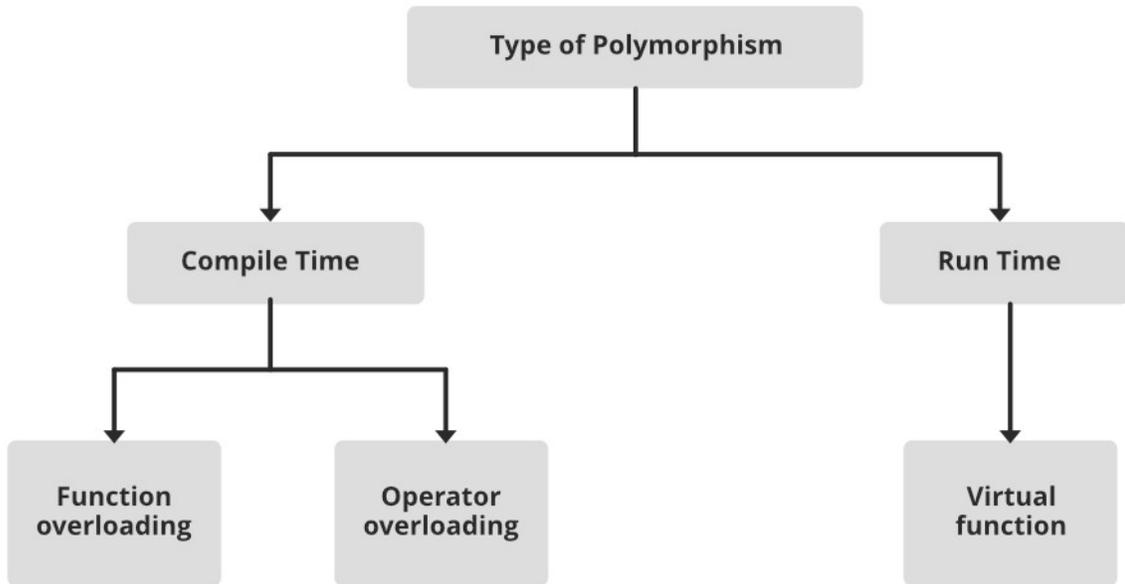
इनहेरिटेंस OOP (ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग) का एक महत्वपूर्ण स्तंभ है। किसी क्लास की किसी दूसरी क्लास से गुण और विशेषताएँ प्राप्त करने की क्षमता को इनहेरिटेंस कहा जाता है। जब हम कोई क्लास लिखते हैं, तो हम दूसरी क्लास से गुण प्राप्त करते हैं। इसलिए जब हम कोई क्लास बनाते हैं, तो हमें बार-बार सभी गुण और फंक्शन लिखने की ज़रूरत नहीं होती, क्योंकि इन्हें उस क्लास से प्राप्त किया जा सकता है। इनहेरिटेंस उपयोगकर्ता को जब भी संभव हो कोड का पुनः उपयोग करने और इसकी अतिरेकता को कम करने की अनुमति देता है।



DSG Support Multi Solution

6. बहुरूपता:

बहुरूपता शब्द का अर्थ है कई रूप होना। सरल शब्दों में, हम बहुरूपता को एक संदेश की एक से अधिक रूपों में प्रदर्शित होने की क्षमता के रूप में परिभाषित कर सकते हैं। उदाहरण के लिए, एक व्यक्ति में एक ही समय में अलग-अलग विशेषताएँ हो सकती हैं। जैसे एक व्यक्ति एक ही समय में पिता, पति, कर्मचारी हो सकता है। तो एक ही व्यक्ति अलग-अलग परिस्थितियों में अलग-अलग व्यवहार रखता है। इसे बहुरूपता कहते हैं।



7. गतिशील बंधन:

डायनेमिक बाइंडिंग में, फंक्शन कॉल के जवाब में निष्पादित किया जाने वाला कोड रनटाइम पर तय किया जाता है। डायनेमिक बाइंडिंग का मतलब है कि किसी दिए गए प्रक्रिया कॉल से जुड़ा कोड रन टाइम पर कॉल के समय तक ज्ञात नहीं होता है। डायनेमिक मेथड बाइंडिंग इनहेरिटेन्स के मुख्य लाभों में से एक यह है कि कुछ व्युत्पन्न वर्ग D में उसके बेस क्लास B के सभी सदस्य होते हैं। एक बार जब D B के किसी भी सार्वजनिक सदस्य को छिपा नहीं रहा होता है, तो D का कोई ऑब्जेक्ट किसी भी संदर्भ में B का प्रतिनिधित्व कर सकता है जहाँ B का उपयोग किया जा सकता है। इस विशेषता को उपप्रकार बहुरूपता के रूप में जाना जाता है।

8. संदेश भेजना:

यह संचार का एक रूप है जिसका उपयोग ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग के साथ-साथ समानांतर प्रोग्रामिंग में भी किया जाता है। ऑब्जेक्ट एक दूसरे को सूचना भेजकर और प्राप्त करके एक दूसरे से संवाद करते हैं।

DSG Support Multi Solution

किसी ऑब्जेक्ट के लिए संदेश एक प्रक्रिया के निष्पादन के लिए एक अनुरोध है और इसलिए प्राप्त करने वाले ऑब्जेक्ट में एक फंक्शन को आमंत्रित करेगा जो वांछित परिणाम उत्पन्न करता है। संदेश पास करने में ऑब्जेक्ट का नाम, फंक्शन का नाम और भेजी जाने वाली जानकारी निर्दिष्ट करना शामिल है।

हमें ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग की आवश्यकता क्यों है?

- परियोजनाओं के विकास और रखरखाव को और अधिक सरल बनाना।
- डेटा छिपाने की सुविधा प्रदान करना जो सुरक्षा चिंताओं के लिए अच्छा है।
- यदि हम ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग का उपयोग कर रहे हैं तो हम वास्तविक दुनिया की समस्याओं को हल कर सकते हैं।
- यह कोड की पुनः प्रयोज्यता सुनिश्चित करता है।
- यह हमें सामान्य कोड लिखने की सुविधा देता है: जो विभिन्न प्रकार के डेटा के साथ काम करेगा, इसलिए हमें बार-बार बुनियादी चीजें लिखने की जरूरत नहीं होगी।

Constructors

कंस्ट्रक्टर क्लास का एक विशेष तरीका है जो क्लास का इंस्टेंस बनने पर अपने आप लागू हो जाता है। C# में कंस्ट्रक्टर ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग के मूलभूत घटक हैं। विधियों की तरह, इसमें उन निर्देशों का संग्रह होता है जिन्हें ऑब्जेक्ट निर्माण के समय निष्पादित किया जाता है। इसका उपयोग उसी क्लास के डेटा सदस्यों को प्रारंभिक मान निर्दिष्ट करने के लिए किया जाता है। कंस्ट्रक्टर के बारे में कुछ महत्वपूर्ण बिंदु नीचे दिए गए हैं:

- किसी क्लास के कंस्ट्रक्टर का नाम उस क्लास के नाम के समान होना चाहिए जिसमें वह स्थित है।
- एक क्लास में किसी भी संख्या में कंस्ट्रक्टर्स हो सकते हैं।
- एक कंस्ट्रक्टर अमूर्त, अंतिम और सिंक्रोनाइज्ड नहीं हो सकता।
- कंस्ट्रक्टर का कोई रिटर्न प्रकार नहीं होता।

- *Class Student*

- Private sub1Mark As Integer*

- Private sub2Mark As Integer*

- Private sub3Mark As Integer*

- Private totalMarks As Integer*

- Public Sub New()*

- totalMarks = 0*

- End Sub*

DSG Support Multi Solution

Sub Main()

Dim student1 As Student = New Student () ' यहाँ पे constructor कॉल होगा

- ' और student1 ऑब्जेक्ट के totalMarks variable को 0 वैल्यू असाइन होगी।

End Sub

End Class

The constructor is a special method of a class created with a `New` keyword and does not have a return type. The main purpose of the constructor is to initialize the memory, allocated for the object, where its fields will be stored.

There are various types of constructors.

- [Default Constructor](#)
- [Parameterized Constructor](#)
- [Copy Constructor](#)
- [Private Constructor](#)

Default Constructor#

If you don't provide a constructor for your class, VB.NET creates one by default that instantiates the object and sets member variables to the default values.

Here is an example of the most simplified parameterless constructor in the `CustomerInfo` class.

```
Public Sub New()  
    Console.WriteLine("Default Constructor Called")  
End Sub
```

The following code shows how to call a default parameterless constructor.

```
Dim customer As CustomerInfo = New CustomerInfo()
```

In the above example, using the `New` keyword, we call the constructor of the class `CustomerInfo`. The memory is allocated for the newly created object of the `CustomerInfo` type.

Parameterized Constructor#

Similar to the methods, if we need extra data to create an object, the constructor gets it through a parameter list. The following example passes two strings variables `name` and `address`, as a parameter to the constructor.

DSG Support Multi Solution

```
Public Sub New(ByVal name As String, ByVal address As String)
    Me.Name = name
    Me.Address = address
End Sub
```

Similarly, the call of a constructor with parameters is done in the same way as the call of a method with parameters. The required values are supplied as a list separated with commas.

```
Dim customer1 As CustomerInfo = New CustomerInfo("John", "11 wall street")
```

There is no limit to parameters. You can have as many parameters as you want.

Copy Constructor#

The constructor which creates an object by copying variables from another object is called a copy constructor. The purpose of a copy constructor is to initialize a new instance to the values of an existing instance.

```
Public Sub New(ByVal customer As CustomerInfo)
    Me.Name = customer.Name
    Me.Address = customer.Address
End Sub
```

The copy constructor is called by passing the object as a parameter of the same class.

```
Dim customer1 As CustomerInfo = New CustomerInfo("John", "11 wall street")
Dim customer2 As CustomerInfo = New CustomerInfo(customer1)

customer1.Print()
customer2.Print()
```

Let's execute the above code, and you will see the same values for both objects.

```
Name: John, Address: 11 wall street
Name: John, Address: 11 wall street
```

Private Constructor#

A private constructor is declared by specifying the `Private` access modifier. It is used to prevent creating instances of a class when we have only `Shared` members.

```
Public Class Counter
    Private Sub New()
    End Sub

    Public Shared currentCount As Integer

    Public Shared Function IncrementCount() As Integer
```

DSG Support Multi Solution

```
        currentCount += 1
        Return currentCount
    End Function
End Class
```

You can call the members using the class name instead of creating its object.

```
Counter.currentCount = 9
Counter.IncrementCount()
Console.WriteLine("New count: {0}", Counter.currentCount)
```

Let's execute the above code and you will see the following output.

```
New count: 10
```

Access Modifiers

VB.NET में Access Modifiers का उपयोग किसी Class, Method, Variable आदि की Accessibility (पहुँच या एक्सेस) को नियंत्रित करने के लिए किया जाता है। यानी, हम यह तय करते हैं कि कौन-सा कोड किस हिस्से को एक्सेस कर सकता है।

नीचे VB.NET के सभी Access Modifiers की हिंदी में पूरी जानकारी दी गई है:

1. Public (सार्वजनिक)

- यह सबसे ओपन एक्सेस मोडिफायर है।
- किसी भी Class या Project से Access किया जा सकता है।

```
vb.net
CopyEdit
Public Class MyClass
    Public Sub Show()
        Console.WriteLine("Hello from Public method")
    End Sub
End Class
```

उपयोग: जब आपको कोई डेटा या फंक्शन हर जगह से एक्सेस करवाना हो।

DSG Support Multi Solution

2. Private (निजी)

- यह सबसे Restrictive होता है।
- केवल उसी Class के अंदर एक्सेस किया जा सकता है जिसमें इसे डिफाइन किया गया है।

```
vb.net
CopyEdit
Public Class MyClass
    Private Sub Show()
        Console.WriteLine("Hello from Private method")
    End Sub
End Class
```

 **उपयोग:** जब आप चाहते हैं कि Method या Variable केवल उसी Class के अंदर काम करें।

3. Protected (संरक्षित)

- यह Private जैसा ही होता है लेकिन Inheritance में इस्तेमाल होता है।
- इसे उसी Class या उसकी Derived (उत्तराधिकार प्राप्त) Class से Access किया जा सकता है।

```
vb.net
CopyEdit
Public Class Parent
    Protected Sub Show()
        Console.WriteLine("Hello from Protected method")
    End Sub
End Class

Public Class Child
    Inherits Parent
    Public Sub Display()
        Show() ' यहाँ यह एक्सेसिबल है
    End Sub
End Class
```

 **उपयोग:** जब आप Child Class को Access देना चाहते हैं, लेकिन बाकी किसी को नहीं।

DSG Support Multi Solution

□ 4. Friend (मित्र)

- केवल उसी Assembly (Project) के अंदर Access किया जा सकता है।
- किसी दूसरी Assembly से Access नहीं किया जा सकता।

```
vb
CopyEdit
Friend Class MyClass
    Friend Sub Show()
        Console.WriteLine("Hello from Friend method")
    End Sub
End Class
```

□ **उपयोग:** जब आप चाहते हैं कि कोड केवल उसी प्रोजेक्ट के अंदर से एक्सेस हो।

5. Protected Friend

- इसे **Protected** और **Friend** दोनों का कॉम्बिनेशन कहा जा सकता है।
- Derived Class और उसी Assembly से Access किया जा सकता है।

```
vb.net
CopyEdit
Public Class MyClass
    Protected Friend Sub Show()
        Console.WriteLine("Hello from Protected Friend method")
    End Sub
End Class
```

● **उपयोग:** जब आप चाहते हैं कि कोड Derived Class और उसी प्रोजेक्ट में एक्सेस हो।

ADO.NET

.NET फ्रेमवर्क में अपनी खुद की डेटा एक्सेस तकनीक शामिल है, यानी ADO.NET। ADO.NET माइक्रोसॉफ्ट की यूनिवर्सल डेटा एक्सेस रणनीति का नवीनतम कार्यान्वयन है। इसमें प्रबंधित क्लासेस शामिल हैं जो .NET अनुप्रयोगों को Microsoft SQL Server, Microsoft Access, Oracle, XML, आदि जैसे डेटा स्रोतों से कनेक्ट करने, कमांड निष्पादित करने और डिस्कनेक्ट किए गए डेटा को प्रबंधित करने की अनुमति देते हैं।

DSG Support Multi Solution

Microsoft ADO.NET, ADO पर एक सुधार है। ADO.NET को सबसे पहले .NET फ्रेमवर्क के 1.0 संस्करण में पेश किया गया था और यह विभिन्न मोड में डेटा को संभालने के लिए कई तरह की सुविधाएँ प्रदान करता है, जैसे कनेक्टेड मोड और डिस्कनेक्टेड मोड। कनेक्टेड मोड में, हम लाइव डेटा से निपटते हैं, जबकि डिस्कनेक्टेड मोड में, डेटा को स्टोर किए गए स्रोत से एक्सेस किया जाता है।

ADO.NET को मुख्य रूप से डेटा स्रोतों से प्राप्त डेटा के साथ काम करने के दो सामान्य तरीकों को संबोधित करने के लिए विकसित किया गया था:

- पहले दृष्टिकोण में एक बार डेटा तक पहुंचना और एक ही बार में उसका पुनरावर्तन करना, अर्थात् डेटा को रनटाइम मेमोरी में कैश करना शामिल है।
- दूसरे दृष्टिकोण में डेटा के साथ जुड़े हुए तरीके से काम करना शामिल है, जहां डेटा कैश नहीं किया जाता है, और एप्लिकेशन इसे पुनः प्राप्त करने के लिए हमेशा डेटाबेस से पूछताछ करता है।

ADO.NET Architecture

ADO.NET एक बहुस्तरीय वास्तुकला का उपयोग करता है जो कुछ प्रमुख अवधारणाओं के इर्द-गिर्द घूमती है:

- Connection
- Command
- DataSet objects

ADO.NET की वास्तुकला ADO से थोड़ी भिन्न है, जैसा कि निम्नलिखित चित्र में दिखाया गया है (यहाँ शामिल नहीं है)।

ADO और ADO.NET के बीच मुख्य अंतरों में से एक यह है कि वे विभिन्न डेटा स्रोतों को कैसे संभालते हैं। ADO.NET में, प्रोग्रामर अंतर्निहित डेटा स्रोत की परवाह किए बिना ऑब्जेक्ट के एक सामान्य सेट का उपयोग करते हैं।

उदाहरण के लिए: यदि हम Oracle डेटाबेस से कोई रिकॉर्ड प्राप्त करना चाहते हैं, तो हम उसी कनेक्शन क्लास का उपयोग करते हैं जिसका उपयोग हम SQL सर्वर के साथ करेंगे। ADO.NET इस लचीलेपन को प्राप्त करने के लिए DataSet के साथ-साथ डेटा प्रदाता मॉडल का उपयोग करता है।

ADO.NET की विशेषताएं

इंटरोऑपरेबिलिटी

XML दस्तावेज़ टेक्स्ट-आधारित प्रारूप हैं, जिसका अर्थ है कि उन्हें मानक टेक्स्ट-संपादन उपकरणों का उपयोग करके संपादित किया जा सकता है। ADO.NET सभी डेटा एक्सचेंजों और डेटा के आंतरिक प्रतिनिधित्व के लिए XML का उपयोग करता है।

DSG Support Multi Solution

रख-रखाव

ADO.NET को डेटा लॉजिक और यूजर इंटरफ़ेस के बीच अलगाव के साथ डिज़ाइन किया गया है। यह डेवलपर्स को स्वतंत्र परतों में एप्लिकेशन बनाने की अनुमति देता है, जिससे उन्हें बनाए रखना आसान हो जाता है।

प्रोग्रामेबिलिटी (टाइप्ड प्रोग्रामिंग)

टाइप्ड प्रोग्रामिंग एक ऐसी शैली है जिसमें विशिष्ट उपयोगकर्ता-परिभाषित प्रकारों का उपयोग कथनों के निर्माण या अभिव्यक्तियों का मूल्यांकन करने के लिए किया जाता है। उदाहरण के लिए, "छात्र" तालिका से "कवल" के लिए "अंक" कॉलम का चयन करने के लिए, आप इसका उपयोग करेंगे:

```
डेटासेट.स्टूडेंट["कवल"].माक्स;
```

प्रदर्शन

ADO.NET एक डिस्कनेक्टेड डेटा आर्किटेक्चर का उपयोग करता है, जो डेटाबेस पर लोड को कम करता है और स्केलेबिलिटी में सुधार करता है। चूंकि अधिकांश ऑपरेशन क्लाइंट-साइड पर संभाले जाते हैं, इसलिए समग्र प्रदर्शन बेहतर होता है।

अनुमापकता

स्केलेबिलिटी का मतलब है सिस्टम की क्षमता, बिना प्रदर्शन को कम किए क्लाइंट की बढ़ती संख्या को संभालने की। डिस्कनेक्टेड डेटा एक्सेस का उपयोग करके, एप्लिकेशन लंबे समय तक डेटाबेस कनेक्शन को बनाए नहीं रखते हैं। यह दृष्टिकोण संसाधनों को संरक्षित करता है और कई उपयोगकर्ताओं को एक साथ डेटा एक्सेस करने की अनुमति देता है।

The ADO.NET Architecture is comprised of following important components:

- **Connection**
- **Command**
- **DataReader**
- **DataAdapter**
- **DataSet**

Connection

- It is an important component of ADO.NET Architecture .
- It is used to establish connection.
- It is essential to connect with your backend database.
- Backend database can be SQL Server, Oracle, MySQL, etc.

DSG Support Multi Solution

- To create a connection object, we need at least two things. The first one is the physical address of your database. And the second thing is the security credentials i.e. user name and password
- The first is to create the connection object and the connection object is required to connect the front-end application with the backend data source.
- A connection string is a string of parameters that contains the necessary details for establishing a connection with the data source. It includes information such as the data source location (e.g., server name or file path), credentials (e.g., username and password), and any other required settings specific to the data source.

Command

- It is used to execute a SQL query.
- It serves as the bridge between the application and the data source, allowing developers to interact with the data by issuing SQL queries or executing stored procedures.
- To execute SQL statements or stored procedures, the Command Object provides a range of Execute methods.
- The Execute methods offered by the Command Object allow for various execution scenarios.
- They can be used to retrieve results from the data source, modify data within the data source, or execute non-query commands that do not return any data.
- The choice of the appropriate Execute method depends on the type of SQL command being executed and the expected outcome.

DataReader

- It is used to read the resultset.
- It provides stream-based, forward-only mechanism for retrieving query results from the data source.
- the DataReader is specifically designed for read-only operations and does not support data modification.
- It reads the data row by row, allowing the application to process and consume the data in a sequential fashion.
- the DataReader is optimized for performance and memory efficiency.
- It efficiently streams data from the data source as it is being read, eliminating the need to load the entire result set into memory.

DSG Support Multi Solution

- This makes it highly suitable for working with large datasets where memory consumption is a concern.

Data Adapter

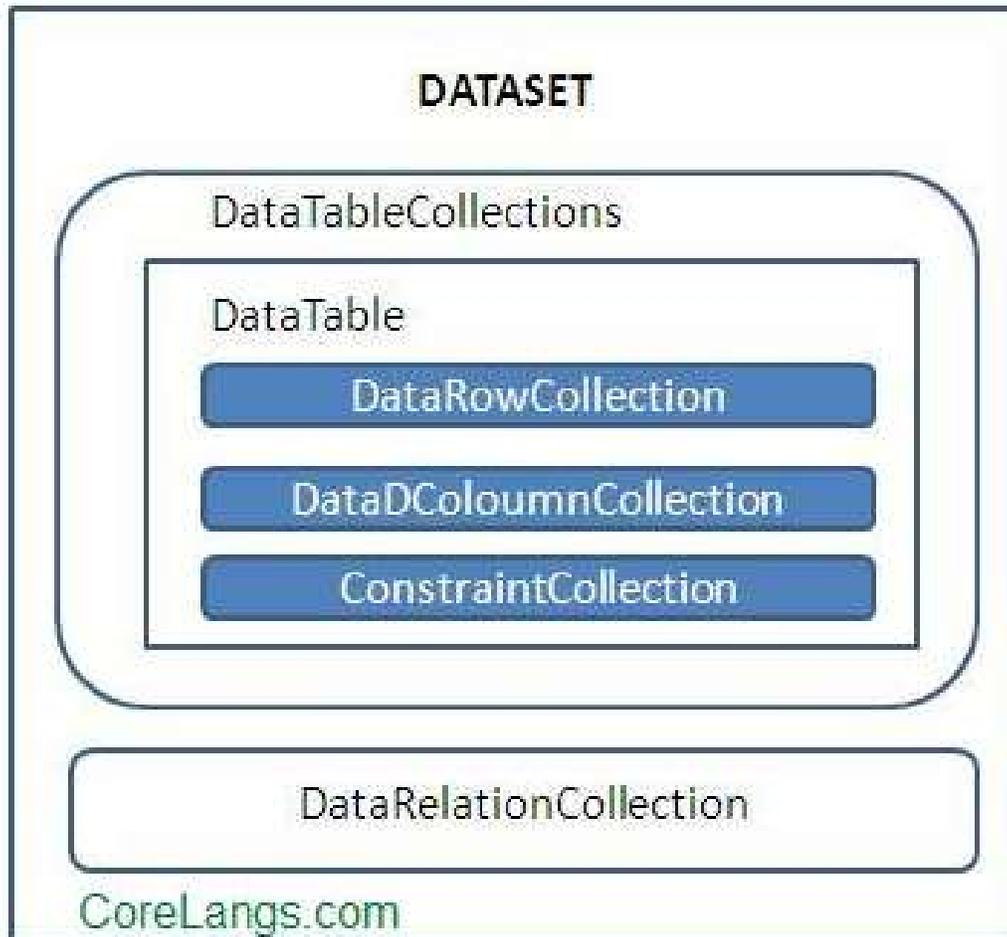
- It populates the Dataset Object with results retrieved from a data source.
- It acts as a bridge between disconnected Dataset objects and the physical data source.
- When working with ADO.NET, the Dataset Object serves as an in-memory representation of data retrieved from a data source.
- To initially populate the Dataset Object with data, the DataAdapter Object comes into play.
- It handles the communication and data retrieval process, utilizing the appropriate Connection Object to establish a connection with the data source.
- Once the connection is established, the DataAdapter executes the necessary SQL statements or stored procedures and retrieves the corresponding data.
- It populates the Dataset's tables, rows, and columns with the retrieved data, ensuring that the in-memory representation aligns with the data structure from the data source.

DataSet

- It provides the disconnected representation of results from the data source.
- The DataSet is entirely independent from the data source.
- It is a Disconnected record set that can be browsed in both i.e. forward and backward mode.
- It is not read-only i.e. you can update the data present in the data set.
- DataSet is a collection of DataTables that holds the data and we can add, update, and delete data in a data table.
- DataSet gets filled by DataAdapter.
- Within the DataSet, data is organized into rows and columns, similar to a traditional database table.
- It also supports defining primary keys, constraints, and relationships between DataTable objects.

CLR

DSG Support Multi Solution



Advantages of ADO.Net over ADO

- There are some similarities and differences between ADO and ADO.NET.
- ADO, being an older technology, relies on COM (Component Object Model) for its implementation. On the other hand, ADO.NET is built on top of the .NET Common Language Runtime and utilizes managed providers defined within the .NET framework.
- One similarity between ADO and ADO.NET is the need for a connection to a data store in order to fetch data. Both models require establishing a connection to the data source, and the code for establishing a connection is similar in many cases.

DSG Support Multi Solution

VB.Net - Dialog Boxes

There are many built-in dialog boxes to be used in Windows forms for various tasks like opening and saving files, printing a page, providing choices for colors, fonts, page setup, etc., to the user of an application. These built-in dialog boxes reduce the developer's time and workload.

All of these dialog box control classes inherit from the **CommonDialog** class and override the **RunDialog()** function of the base class to create the specific dialog box.

The **RunDialog()** function is automatically invoked when a user of a dialog box calls its **ShowDialog()** function.

The **ShowDialog** method is used to display all the dialog box controls at run-time. It returns a value of the type of **DialogResult** enumeration. The values of DialogResult enumeration are –

Abort – returns DialogResult.Abort value, when user clicks an Abort button.

Cancel – returns DialogResult.Cancel, when user clicks a Cancel button.

Ignore – returns DialogResult.Ignore, when user clicks an Ignore button.

No – returns DialogResult.No, when user clicks a No button.

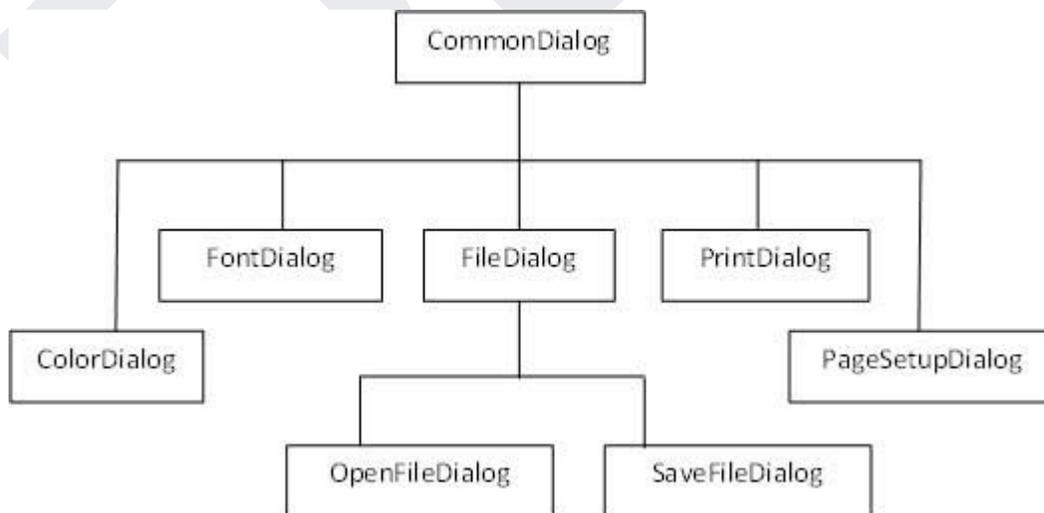
None – returns nothing and the dialog box continues running.

OK – returns DialogResult.OK, when user clicks an OK button

Retry – returns DialogResult.Retry, when user clicks an Retry button

Yes – returns DialogResult.Yes, when user clicks an Yes button

The following diagram shows the common dialog class inheritance –



All these above-mentioned classes have corresponding controls that could be added from the Toolbox during design time. You can include relevant functionality of these classes to your application, either by instantiating the class programmatically or by using relevant controls.

DSG Support Multi Solution

When you double click any of the dialog controls in the toolbox or drag the control onto the form, it appears in the Component tray at the bottom of the Windows Forms Designer, they do not directly show up on the form.

The following table lists the commonly used dialog box controls. Click the following links to check their detail –

Sr.No.	Control & Description
1	<u>ColorDialog</u> It represents a common dialog box that displays available colors along with controls that enable the user to define custom colors.
2	<u>FontDialog</u> It prompts the user to choose a font from among those installed on the local computer and lets the user select the font, font size, and color.
3	<u>OpenFileDialog</u> It prompts the user to open a file and allows the user to select a file to open.
4	<u>SaveFileDialog</u> It prompts the user to select a location for saving a file and allows the user to specify the name of the file to save data.
5	<u>PrintDialog</u> It lets the user to print documents by selecting a printer and choosing which sections of the document to print from a Windows Forms application.

Thanks & Regards

DSGSupport Multi Solution