

**बी.सी.एस.टी.**



# कनक कम्प्यूटर एज्युकेशन

मारवनलाल चतुर्वेदी विश्वविद्यालय से संबद्ध

**Mob: 9589995353, 9926326401**

**website :- [www.bestkanak.in](http://www.bestkanak.in)**

**DCA**

1 वर्षीय

कृष्णा टाकिज के पीछे, राऊत भवन के सामने मुलताई

**PGDCA**

1 वर्षीय



# PGDCA Semester -1

Author : Narendra Tiwari

प्रोग्रामिंग विद VB.net



प्रोग्रामिंग विद VB.net

Narendra Publication

## .net का परिचय

.net यह Microsoft कंपनी ने विकसित किया है, विंडो पर आधारित प्रोग्रामिंग करने एवं सॉफ्टवेयर के विकास के लिए नया प्लेटफॉर्म बनाया है। वर्तमान में प्रोग्रामिंग क्षेत्र के व्यक्तियों के लिए डॉट नेट यह बहुत महत्वपूर्ण टूल हो गया है। .net ना सिर्फ प्रोग्रामिंग क्षेत्र में बहुत प्रयोग हो रहा है, अपितु इंटरनेट में विभिन्न कार्य हेतु अधिक प्रयोग हो रहा है। वस्तुतः “.” (dot) यह शब्द इंटरनेट के वेबसाईट में प्रयोग होने के कारण ही “.” डाट नाम दिया गया है। डॉट नेट एक प्लेटफॉर्म है, उसके अनुरूप मायक्रोसॉफ्ट ने कई भाषाओं को विकसित किया है, यह भाषाएं इस प्लेटफॉर्म का अनुकरण करती है उन्हें हम डॉट नेट प्रोग्रामिंग भाषा कहते हैं। .net यह अप्लिकेशन प्रोग्राम एवं ऑपरेटिंग सिस्टम के बीच में कार्य करती है।

इसमें विभिन्न कार्यों के लिए कोड की बहुत बड़ी लायब्ररी है। इस प्लेटफॉर्म का सबसे महत्वपूर्ण बिंदु यह है, कि इसमें विभिन्न प्रोग्रामिंग भाषा के कोड एक दूसरे से साझा कर सकते हैं। इस प्लेटफॉर्म में प्रोग्रामिंग भाषा में लिखे सॉफ्टवेयर का मेमोरी प्रबंधन, डाटा की सुरक्षा व्यवस्था, आदि को सपोर्ट करती है। .net framework यह object oriented programming पर आधारित है। इस व्यवस्था में प्रोग्रामर को बहुतसे कार्य के कोड readymade प्राप्त होते हैं। इस प्लेटफॉर्म पर visual basic, C# आदि प्रोग्रामिंग भाषा में कोड बना सकते हैं। जो प्रोग्राम .net के लिए विकसित किये जाते हैं, तथा उसके अंदर चलाये जाते हैं, वह .net द्वारा नियंत्रित होते हैं। .net यह प्रोग्राम का सफल एवं सुरक्षित क्रियान्वयन सुनिश्चित करता है। vb.net यह .net framework का ही एक हिस्सा है।

विंडो ऑपरेटिंग सिस्टम यह .net framework का आधारभूत घटक है, जो उन्नत अनुप्रयोग एवं वेब आधारित अप्लिकेशन निर्माण में सहायक है। .net framework को निम्न उद्देश्यों को पूर्ण करने के लिए बनाया गया है,

1. ऑब्जेक्ट ओरियंटेड प्रोग्रामिंग के लिए उपयुक्त एवं सक्षम environment प्रदान करना।
2. कोड के क्रियान्वयन के लिए ऐसे पद्धति का निर्माण करना, जिसमें इच्छित कार्य के लिए कम से कम कोड का प्रयोग हो, तथा कोड का क्रियान्वयन बिना किसी बाधा के सुचारु रूप से पूर्ण हो।
3. कोड क्रियान्वयन के लिए ऐसे पद्धति का निर्माण करना, जिसमें कोड सुरक्षित रूप से कार्य कर सके, तथा अन्य कम्प्यूटर पर भी सही परिणाम दे।
4. प्रोग्रामर को विभिन्न प्रकार के अप्लिकेशन के निर्माण के लिए एक ही जगह से कार्य पूर्ण करने की सुविधा हो।

### डॉट नेट की आवश्यकता (Need of .net )

वर्तमान में बहुतसी प्रोग्रामिंग भाषा का प्रयोग हो रहा है, जैसे C, C++, VB, Java, आदि। प्रत्येक भाषा में प्रोग्राम बनाने के नियम अलग अलग हैं। यदि एक भाषा में वेरिएबल बनाने के लिए dim शब्द का प्रयोग होता है, तब दूसरे प्रोग्रामिंग भाषा में let शब्द का प्रयोग होता है। एवं सभी प्रोग्रामर को सभी भाषा का विस्तृत ज्ञान होना संभव नहीं था। जैसे प्रत्येक प्रोग्रामिंग भाषा के सिन्टेक्स अलग अलग हैं, वैसे ही उनके कार्यक्षेत्र भी अलग अलग हैं। इसलिए एक प्रोग्रामर के लिए सभी क्षेत्र के प्रोग्राम बनाने में भी कठिनाई होती थी। इस समस्या के समाधान के लिए Microsoft कंपनी ने ऐसे प्लेटफॉर्म बनाने की सोची, जिसमें विभिन्न प्रोग्रामिंग भाषा में बनाये प्रोग्राम को एक समान कोड में बदल कर उसे क्रियान्वित किया जा

सके। इस प्लेटफॉर्म को .net framework कहते हैं। इसके कार्य को समझने के लिए हम एक उदाहरण लेते हैं, उदाहरण के लिए आप किसी ऐसे प्रोजेक्ट पर कार्य कर रहे हैं जहां बहुतसे अलग-अलग जगहों के व्यक्तियों की आवश्यकता है, इसलिए दुनिया के विभिन्न कोनों से लोगों को इस प्रोजेक्ट पर कार्य करने के लिये आमंत्रित करते हैं। प्रत्येक देश के व्यक्ति की भाषा अलग अलग है, जिससे सबको एक दूसरे के कार्य को समझने एवं समझाने में मुश्किल हो रही है। लेकिन आपको उन प्रत्येक के क्षमता के अनुसार कार्य करवाना है। तब आप प्रत्येक भाषा समझकर उसे एक सर्वमान कोड में परिवर्तित करेंगे, जिससे अंत में कार्य पूर्ण हो सके। डॉट नेट फ्रेमवर्क ऑपरेटिंग सिस्टम तथा आप के द्वारा लिखे गये प्रोग्राम के बीच एक सेतु (Intermediate level) का कार्य करता है। ऐसा करने से प्रोग्रामिंग की दो प्रमुख समस्याओं का निदान होता है। यह अलग-अलग प्रोग्रामिंग भाषाओं को एक दूसरे के साथ तालमेल बनाए रखने में सहायता करता है। डॉट नेट फ्रेमवर्क में प्रोग्रामिंग भाषाएं कम्प्यूटर ऑपरेटिंग सिस्टम के साथ डायरेक्ट एक्सेस बनाने के बजाय अर्थात् ऑपरेटिंग सिस्टम में स्वयं स्ट्रिंग तथा संख्याओं को संग्रहित करने के बजाय डॉट नेट इसे अपने तरीके से कम्प्यूटर पर संग्रहित करता है। फलस्वरूप आप निश्चित होकर दो अलग-अलग नेट भाषाओं में अपने प्रोग्राम को विकसित कर पाते हैं।

इस प्लेटफॉर्म के लिए Microsoft ने विभिन्न प्रोग्रामिंग भाषा का विकास किया, जिसमें Visual basic, C# (C sharp), ASP, C++ आदि हैं।

जो प्रोग्राम वी.बी. डॉट नेट या सी पर विकसित है, वह आप उसी पर क्रियान्वित हो सकती है, जिसमें वह प्रोग्राम कम्प्यूटर में है। वी. बी डॉट नेट पर विकसित किये हुए प्रोग्राम को इंटरनेट या लैन

(लोकल एरिया नेटवर्क) पर आसानी से वितरित किया जा सकता है। आप दुनिया के किसी कम्प्यूटर पर संग्रहित कर उसे उस नेटवर्क से जुड़े किसी भी कम्प्यूटर से एक्सेस कर सकते हैं बशर्ते उस पर डॉट नेट फ्रेमवर्क उपलब्ध हो।

## डॉट नेट फ्रेमवर्क के विशेषण (Features of Dot net Framework)

डॉट नेट फ्रेमवर्क से प्रोग्रामिंग एक नयी शुरुवात हुई है। इस संरचना या प्रणाली के निम्न गुण हैं।

**interpretability** :- अधिकांश कंपनी या संस्थाओं में पहले से सॉफ्टवेयर बने होते हैं, उन प्रोग्राम या सॉफ्टवेयर में उस संस्था का डाटा स्टोर रहता है। जब नया प्रोग्राम बनाते हैं, तब पुराने प्रोग्राम के साथ नया प्रोग्राम सही तरीके से जुड़ना चाहिए। .net framework में यह कार्य बहुत अच्छी तरीके से होता है।

**Secure, Multi Language Development Platform** :- डॉट नेट का सबसे मुख्य गुण यह है, कि उसमें विभिन्न प्रोग्रामिंग भाषा सम्मिलित कर प्रोग्राम या सॉफ्टवेयर का निर्माण किया जा सकता है। यह प्रणाली सुरक्षित होने के साथ ही विकास के लिए एक से अधिक भाषा पर कार्य करने हेतु प्लेटफॉर्म देती है। इसमें डाटा एवं प्रोग्राम सुरक्षा के विभिन्न टूल दिये हैं, जिनके सहायता से प्रोग्रामर अपने प्रोग्राम को अधिक सुरक्षित रख सकता है। तथा विभिन्न प्रबंधक टूल भी उपलब्ध हैं। .net यह प्रोग्रामर को इच्छित भाषा में प्रोग्रामिंग करने के स्वतंत्रता प्रदान करता है, साथ ही उसे अन्य भाषा की सुविधा भी प्रदान करता है। कॉमन लैंग्वेज रनटाइम शक्तिशाली स्टैटिक भाषाओं यथा वी.बी. डॉट नेट तथा विजुअल सी++ को सपोर्ट करता है।

**Common runtime engine** :- .net framework में CRE की सहायता से प्रोग्राम को क्रियान्वित करना आसान हो गया है। .net में बनाये गये सभी प्रोग्राम CLR के अतर्गत कार्यान्वित होते हैं। CLR प्रोग्राम के कार्यान्वयन की सभी आवश्यकताओं को पूर्ण करता है, जैसे मेमोरी प्रबंधन, सुरक्षा आदि। .net में सभी प्रोग्राम कम्पाइल होने के बाद MSIL (MicroSoft Intermediate Language) में परिवर्तित हो जाते हैं।

**Rapid Model Driven Programming Paradigm**:- इस प्लेटफॉर्म में अधिक तेजी से सॉफ्टवेयर का निर्माण किया जा सकता है। उदाहरण के लिए नया ADO.net नेट एन्टिटी फ्रेमवर्क आधारित पैराडाइम तथा स्टैंडर्ड्स आधारित फ्रेमवर्क प्रस्तुत करता है, जो डेटाबेस प्रोग्रामिंग की एक नयी अवधारणा प्रस्तुत करता है। ए.डी.ओ.डॉट नेट का प्रयोग करके विकासकर्ता बिल्कुल स्पष्ट रूप से अपने बिजनेस लॉजिक डाटा तथा यूजर इंटरफेस को अलग कर पाते हैं।

**Simplified development** :- .net framework में ऐसे बहुत से फीचर हैं, जिनके सहायता से सॉफ्टवेयर का निर्माण करना एवं उसे distribute करना आसान हो गया है। .net framework के सेटअप प्रोग्राम से अप्लिकेशन प्रोग्राम को आसानी से डिजाइन एवं वितरण किया जा सकता है।

**Portability** :- .net में बनाये प्रोग्राम किसी भी विंडो ऑपरेटिंग सिस्टम में कार्यान्वित किये जा सकते हैं।

**Next Generation User Experiences** :- .net framework विंडो के संपूर्ण क्षमता का उपयोग कर, अप्लिकेशन यूजर इंटरफेस, डाक्यूमेंट, तथा मिडिया घटकों को एकत्रीकृत करता है। इस ग्राफिक्स एवं विभिन्न मल्टीमीडिया घटकों का प्रयोग इच्छित एप्लिकेशन का निर्माण किया जा सकता है। विंडोज प्रेजेंटेशन

फॉउंडेशन एक्सटेन्सिबल एप्लीकेशन मार्कआप भाषा, कन्ट्रोलस डाटा बाइन्डिंग, लेआउट, द्वि-विमीय तथा त्रि-विमीय ग्राफिक्स, एनिमेशन, स्टाइल्स, टेम्पलेट्स, डॉक्यूमेंट्स, मीडिया, टेक्स्ट तथा टाईपोग्राफी जैसे फीचरों को उपलब्ध कराता है। आप डॉट नेट फ्रेमवर्क क्लास लाईब्रैरी के अन्य अवयवों को मिलाकर भी एप्लीकेशनों का निर्माण कर सकते हैं।

**Web Application Development :-** .net framework का मुख्य कार्यक्षेत्र वेब अप्लिकेशन का विकास करना है, इसमें ASP.net या .net का एक घटक है, जो निशुल्क उपलब्ध है। इसकी सहायता से वेब साईट का निर्माण करना आसान हो गया है। वर्तमान में छोटे वेबसाईट निर्माता से बड़े इंटरएक्टिव वेबसाईट के निर्माण में इस का प्रयोग हो रहा है।

**सुरक्षित, भरोसेमंद वेब सेवाएँ (Secure, Reliable Web Services) :-** .net framework यह एक common security model का समर्थन करता है, जिससे सभी अप्लिकेशन सुरक्षित रहते हैं, तथा यह डाटा को सुरक्षित रखने में मदद करता है।

**डिवाइसों तथा प्लेटफॉर्मों तक बेहतर पहुंच (Superior Reach Across Devices and Platform) :-** डॉट नेट फ्रेमवर्क पर प्रोग्रामर विभिन्न इलेक्ट्रॉनिक डिवाइस जैसे कम्प्यूटर, सर्वर, मोबाइल फोन तथा एम्बेडेड उपकरणों के लिए प्रोग्राम का निर्माण कर सकता है। इस में बहुतसे टूल उपलब्ध है, जो प्रोग्रामर को एक से अधिक ब्राउजर, प्लेटाफाम या डिवाइस पर अप्लिकेशन बनाने की सुविधा प्रदान करता है।



## डॉट नेट फ्रेमवर्क की कमियाँ (Limitations of Dot net Framework)

जैसे .net के कुछ लाभ हैं, वैसे इस प्लेटफॉर्म की कुछ कमियाँ भी हैं। इस प्लेटफॉर्म की निम्न कमियाँ हैं

- डॉट नेट फ्रेमवर्क केवल नये विंडोज संस्करणों में उपलब्ध है। अतः पुराने विंडोज सिस्टमों पर डॉट नेट प्रोग्राम को चलाना सम्भव नहीं है।
- डॉट नेट प्रोग्रामों को सिर्फ विंडोज ऑपरेटिंग सिस्टम पर ही प्रयोग किया जा सकता है, इसे लाइनक्स, android, या mac ऑपरेटिंग सिस्टम पर नहीं चलाया जा सकता।
- यद्यपि इस प्लेटफॉर्म पर बहुतसे भाषाओं का प्रयोग किया जा सकता है, लेकिन कुछ प्रोग्रामिंग भाषाएँ डॉट नेट फ्रेमवर्क के साथ काम नहीं करती हैं। उदाहरण के लिए Java प्रोग्राम को डॉटनेट फ्रेमवर्क पर नहीं चला सकते।
- यह एक बड़ा एवं जटिल प्लेटफॉर्म है, जिस पर कार्य करने से पहले उसका सही प्रशिक्षण लेना आवश्यक है।

फिर भी, उपरोक्त सभी सीमाओं के बावजूद वर्तमान में यह प्लेटफॉर्म विभिन्न अप्लिकेशन के निर्माण के लिए अधिक प्रयोग हो रहा है।

## डॉट नेट फ्रेमवर्क आर्किटेक्चर (Dot Net Framework Architecture)

डॉट नेट फ्रेमवर्क (Dot Net Framework)

फ्रेमवर्क का शाब्दिक अर्थ होता है आकार तथा सपोर्ट देने वाला ढांचा। यह कहा जा सकता है कि डॉट नेट फ्रेमवर्क एक ढांचा है जो प्रोग्रामिंग को एक आकार, रूप तथा सपोर्ट प्रदान करता है। इसी के आधार पर प्रोग्रामों को विकसित किया जाता है। इसे

प्रोग्राम को ऐसी व्यवस्था भी कहा जा सकता है जो कुछ विशेष प्रोग्रामिंग द्वारा अनुसरण किया जाता है।

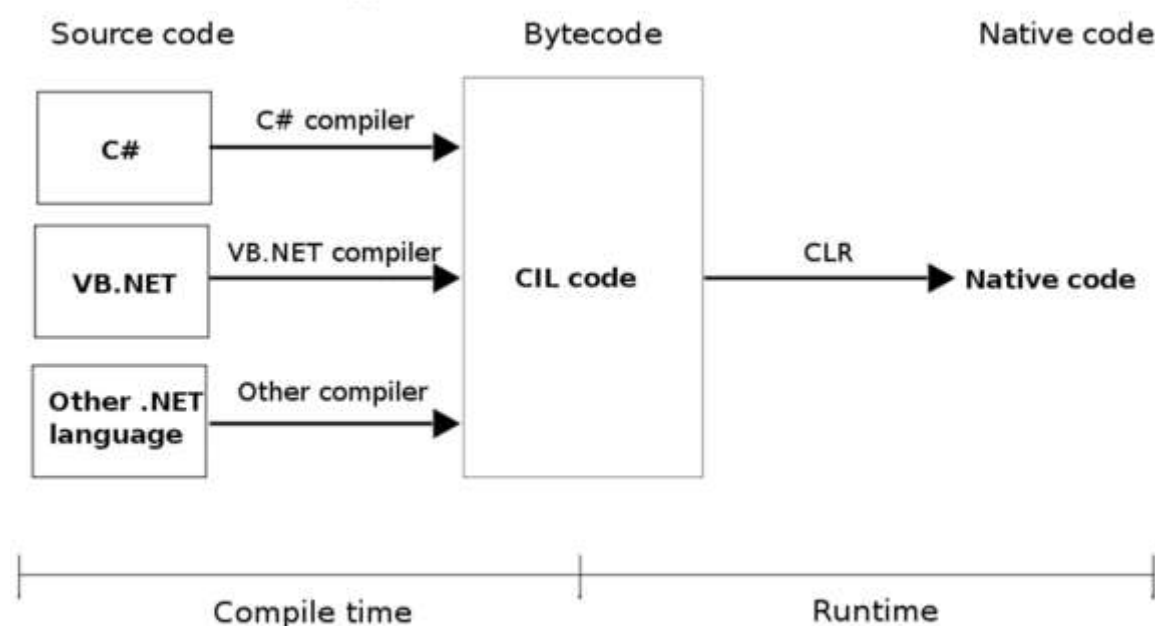
.net framework में मुख्यतः दो घटक है

1. Common language runtime (CLR)
2. .NET Framework class library

इसमे CLR यह नेट फ्रेमवर्क का मुख्य आधार है, यह प्रोग्राम क्रियान्वयन के समय एक प्रबंधक के रूप में कार्य करता है जो प्रोग्राम क्रियान्वयन के समय मेमोरी प्रबंधन, सुरक्षा, कोड शुद्धता आदि का ध्यान रखता है। असल में कोड प्रबंधन ही कोड के क्रियान्वयन का मुख्य तत्व है। इसके अतिरिक्त class library भी नेट फ्रेमवर्क का दूसरा आधार है, इसमे विभिन्न ऑब्जेक्ट का विस्तृत संग्रह है। इस ऑब्जेक्ट लायब्ररी के उपयोग से विभिन्न अप्लिकेशन के निर्माण में बहुत सहायता होती है।

### Common Language runtime

नेट फ्रेमवर्क में बनाये कोड को क्रियान्वित करने की जिम्मेदारी CLR की है। या दूसरे शब्दों में हम कह सकते है, कि .net में



बनाये हुए सभी प्रोग्राम को क्रियान्वयन के लिए CLR उपयोगी है। जो प्रोग्राम CLR में क्रियान्वीत होते हैं, उनमें मेमोरी प्रबंधन कार्य CLR स्वयं ही कर लेता है।

यह घटक प्रोग्रामिंग भाषा के complier से कोड लेकर, उन्हें मशीन भाषा में परिवर्तित करता है, जो कम्प्यूटर के सीपीयू को समझने योग्य होती है। इसे हम उपरोक्त आकृति से समझते हैं, जब आप vb.net पर कोई प्रोग्राम लिखते हैं, तब सर्वप्रथम वह कोड CIL (CLR's Intermediate Language) में परिवर्तित होता है। तथा जब कोड का क्रियान्वयन किया जाता है, तब CIL कोड यह एक विशेष कम्पाइलर की सहायता से बाइनरी कोड में परिवर्तित होते हैं। इस कोड को कम्प्यूटर हार्डवेयर सीधे समझता है।

जिस प्रोग्राम या अप्लिकेशन में विविध प्रोग्रामिंग भाषा के ऑब्जेक्ट प्रयोग हो रहे हैं, वैसे प्रोग्राम का निर्माण CLR की सहायता से सरल हो गया है। CLR की सहायता से विभिन्न भाषा में बने ऑब्जेक्ट एक दूसरे से जुड़ कर कार्य कर सकते हैं, तथा एकत्रीकृत रूप में कार्य करते हैं। कॉमन लैंग्वेज रनटाइम मेमोरी थ्रेड एक्जिक्युशन, कोड एक्जिक्युशन, कोड सेफ्टी वेरिफिकेशन, कम्पाइलेशन तथा अन्य सिस्टम सेवाओं को व्यवस्थित करता है। CLR यह प्रणाली सोर्स कोड को एक इण्टरमीडिएट लैंग्वेज में कम्पाइल करता है। इस इण्टरमीडिएट लैंग्वेज को माइक्रोसॉफ्ट इण्टरमीडिएट लैंग्वेज (MSIL) कहा जाता है।

जब प्रोग्राम को क्रियान्वित करते हैं तब MSIL भाषा को मूल कम्पाइलर के माध्यम से स्थानीय कोड या मशीनी कोड में बदलते हैं। प्रोग्राम कोड के कम्पाइल किये जाने पर परिणामस्वरूप एक पोर्टेबल एक्जीक्यूटेबल फाइल बनती है, जिसमें माइक्रोसॉफ्ट इण्टरमीडिएट लैंग्वेज तथा कुछ अतिरिक्त सूचना होती है। इस अतिरिक्त सूचना को मेटाडेटा कहा जाता है।

मेटाडेटा यह MSIL को विवरण देता है, इसमें नाम, मेथड, हस्ताक्षर और कई अन्य सूचना (dependency information) आदि ब्यौरे मौजूद होते हैं। प्रोग्राम रन होने के दौरान यह मेमोरी को व्यवस्थित करने का कार्य स्वयं ही गार्बेज कलेक्शन नामक प्रोसेस के माध्यम से करता है। इसके कारण प्रोग्रामर का मेमोरी प्रबधन का कार्य आसान हो गया है।

नरेन्द्र पब्लिकेशन

## VB.net

विजुअल बेसिक डॉट नेट या VB.net यह विजुअल बेसिक का माइक्रोसॉफ्ट डॉटनेट फ्रेमवर्क संस्करण है। विजुअल बेसिक डॉट नेट ऑब्जेक्ट ओरिएण्टेड प्रोग्रामिंग भाषा है। वी.बी. डॉट नेट की सहायता से आप बड़े स्तर पर भी एप्लिकेशन विकसित कर सकते हैं। यह अत्यन्त user friendly भाषा है। वी.बी. डॉट नेट प्रोग्रामर अपने विकास के आवश्यकताओं की पूर्ति, विजुअल स्टूडियो की सहायता से करते हैं। इसके अतिरिक्त माइक्रोसॉफ्ट डॉट नेट फ्रेमवर्क पहले से प्रोग्राम किये गये कोड से निर्मित होता है जिसे प्रयोक्ता किसी भी समय एक्सेस कर सकते हैं। इस पहले से ही प्रोग्राम किये गये कोड है, जिन्हे क्लास लाइब्रेरी कहते हैं।

### vb.net के लाभ

1. वी.बी. डॉट नेट प्रोग्रामिंग भाषा के प्रयोग से प्रोग्राम को तेजी के साथ विकसित कर सकते हैं।
2. वी.बी. डॉट नेट क्लास लाइब्रेरी में बहुतसे संसाधन हैं जिनका लाभ आप प्राप्त कर सकते हैं।
3. वी.बी. डॉट नेट को डॉट नेट फ्रेमवर्क का पूरा समर्थन प्राप्त होता है।
4. वी.बी. डॉट नेट जैसे ऑब्जेक्ट ओरिएण्टेड प्रोग्रामिंग भाषा में अपने विचारों को व्यवस्थित करना आसान होता है।
5. वी.बी. डॉट नेट अन्य माइक्रोसॉफ्ट एप्लीकेशन के साथ तालमेल बना पाता है।

### vb.net की कमियाँ

1. वी.बी. डॉट नेट के लम्बे कोडिंग में गलती ढूँढना तथा उन्हें ठीक करना कठिन होता है।

2. प्रोग्राम करने तथा कम्पाइल करने के बाद भी इसकी कोडिंग या प्रोग्रामिंग पैटर्न को समझना आसान होता है। परिणामस्वरूप कोडिंग को कॉपी करना आसान होता है तथा टैलेण्ट का दूरूपयोग हो सकता है।
3. वी.बी. डॉट नेट का लाइसेंस संस्करण महंगा होता है।

### वी.बी. डॉट नेट का विकास (Evolution of VB.Net)

वी.बी. डॉट नेट से पहले विजुअल बेसिक (VB) तथा वी.बी से पहले बेसिक था। इस का Basic यह सम्बन्ध BASIC (beginner all purpose symbolic instruction code) से है। Basic एक डॉस पर आधारित पुरानी भाषा थी, विजुअल बेसिक यह इसी भाषा से विकसित की गई है। 1975 में जब मीट्स (MITS) एल्टेयर 8800 माइक्रो कम्प्यूटर रिलिज हुआ, तब बेसिक एल्टेयर बेसिक हुआ जिसे बिल गेट्स तथा पॉल एलेन ने विकसित किया। समय के साथ बेसिक प्रोग्रामिंग भाषा का प्रयोग कम होने लगा क्योंकि basic भाषा में अधिक जटिल प्रोग्रामिंग संभव नहीं थी। 1991 में बेसिक को इसके विजुअल कम्पोनेण्ट के साथ इनफयुज किया गया तथा विजुअल बेसिक बना। विंडो में अनुप्रयोग(application) बनाने के लिए विजुअल बेसिक सबसे तेज एवं प्रभावशाली भाषा है। विजुअल बेसिक में तेजी से एवं सरलता से अनुप्रयोग विकसित करने के लिए संपूर्ण टूलस् का समूह है, इसे RAD (rapid application development tools) कहा जाता है। विजुअल बेसिक में अनुप्रयोग बनाने के लिए ग्राफिकल पद्धति प्रयोग की जाती है। इसमें फॉर्म (form) में सीधे इच्छित कंट्रोल को लाकर उनकी प्रापर्टी निश्चित (set) की जाती है, तथा कुछ कोड लगाये जाते हैं। इस प्रोग्रामिंग भाषा का उपयोग कर आप विंडो एवं वेब आधारित एप्लिकेशन का निर्माण कर सकते हैं। .net framework का प्रयोग mobile

अप्लिकेशन का निर्माण में किया जा सकता है, जिसे कम्प्यूटर के अतिरिक्त मोबाइल फोन, पामटॉप, टेबलेट आदि पर प्रयोग किया जा सकता है। इसके अतिरिक्त web services अप्लिकेशन को भी इस भाषा के द्वारा विकसित किया जा सकता है। वेब सर्वर के प्रोग्राम को चलाने के लिए वेब सर्विस प्रोग्राम का प्रयोग किया जाता है। vb.net का प्रयोग कर आप multithread application का निर्माण कर सकते हैं। multithread का अर्थ, एक साथ एक से अधिक कार्य को पूर्ण करने की क्षमता है।

विजुअल बेसिक में विजुअल (visual) का अर्थ होता है, कि वह पद्धति जिससे Graphical user interface (GUI) बनाया जा सकता है। इसमें प्रोग्राम बनाने के लिए बहुत बड़े कोड (code) लिखने की आवश्यकता नहीं होती है। आपको सिर्फ पूर्वनिर्धारित आब्जेक्टों को स्क्रीन पर इच्छित जगह रखना पड़ता है। लेकिन इसमें बहुतसे फंक्शन, स्टेटमेंट, कीवर्ड (keyword) डाले गये हैं, जिससे GUI प्रकार की प्रोग्रामिंग की जा सकती है। इसके बाद .net framework के साथ जुड़ कर नयी visual basic.net भाषा का विकास किया गया है।

विजुअल बेसिक जरूरत के अनुसार विभिन्न संस्करणों में उपलब्ध है।

विजुअल बेसिक निम्न संस्करणों में आया है।

1. 1991 में सर्वप्रथम dos पर आधारित विजुअल बेसिक बनाया गया था।
  2. 1993 में दूसरा संस्करण visual basic 3.0 बनाया गया जो पहले संस्करण से बहुत सुधारित आवृत्ती थी।
- उसके बाद visual basic 6.0 प्रयोग हो रहा था। visual basic 6.0 में विभिन्न संस्करणों में प्रयोग हो रहा था।  
visual studio एवं .net के संस्करण

संस्करण	साल	
.net 1.0	2002	visual studio.net
.net 1.1	2003	visual studio.net 2003, windows server 2003
net 2.0	2005	visual studio 2005, windows server 2003 R2
net 3.0	2006	Windows vista, Windows server
.net 3.5	2007	visual studio 2008,
.net 4.0	2010	visual studio 2010

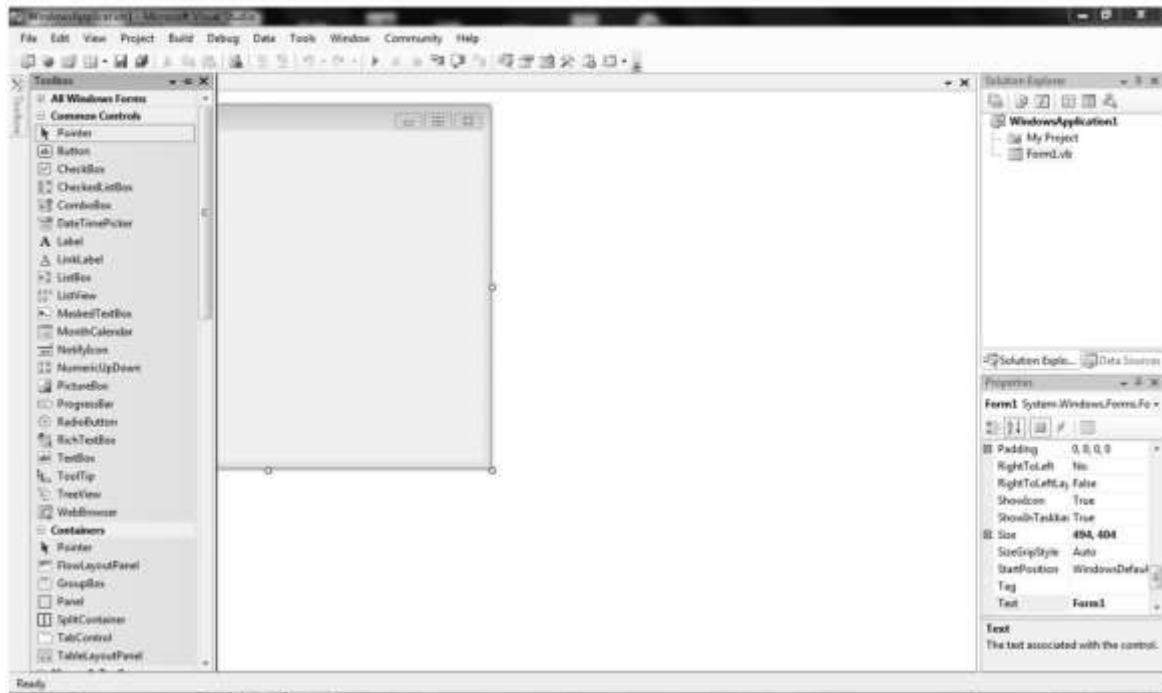
## **Integrated Development Environment** (एकत्रिकृत विकासीत वातावरण)

वी.बी. डॉट नेट में कार्य करने के क्षेत्र को integrated development environment (IDE) कहा जाता है, क्योंकि इसमें सभी काम जैसे प्रोग्राम बनाना, उसमें सुधार करना, प्रोग्राम को चलाना, डाटाबेस को जोड़ना, प्रोजेक्ट बनाना आदि काम एक ही जगह से किये जा सकते हैं। बहुतसी पारंपारिक भाषा में यह सभी टूल अलग अलग प्रोग्राम में होते हैं। IDE में निम्नलिखित घटक है।



## मेन्यू बार

Vb.net में काम करने के लिए सभी कमांड इस बार में उपलब्ध है।



इसमे File, Edit, View, Format, Windows, Help के अतिरिक्त Project, build, Data, Tools, Debug, Community आदि मेन्यू है, जिन को आप प्रोग्राम बनाते एवं चलाते समय प्रयोग कर सकते है। यह टाइटल बार के नीचे होता है। प्रत्येक मेन्यू को क्लिक करने से उससे सम्बन्धित विकल्प दिखाई देते है। प्रत्येक विकल्प किसी कार्य को सम्पादित करते है। इसमे निम्न मेन्यू है

- a. File menu : इस मेन्यू में नई फाइल, project, बनाना, प्रोजेक्ट को सेव करना, पुरानी फाइल को खोलना आदि कार्य कीये जा सकते है।
- b. Edit menu :- इस मेन्यू का प्रयोग मुख्यतः कोड विंडो में होता है। इसमे cut, copy, paste आदि विकल्प होते है। find, replace विकल्प इस मेन्यू है।

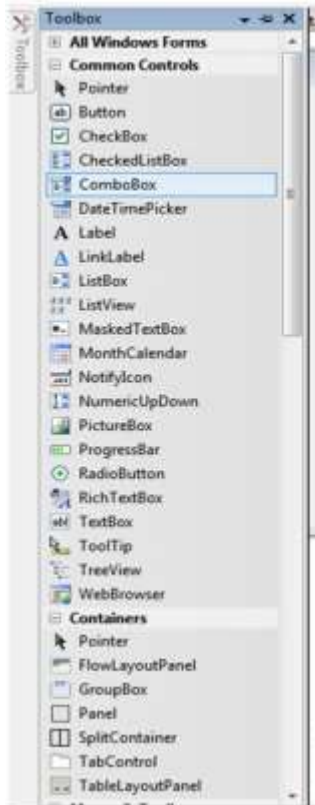
- c. **View menu** : इस मेन्यू में IDE के विभिन्न हिस्सों को चालू या बंद किया जा सकता है।
- d. **Project** :- इस मेन्यू के विकल्पों द्वारा प्रोजेक्ट के विभिन्न घटकों को जोड़ा या निकाला जा सकता है। इसके विकल्पों में प्रोजेक्ट में नया फॉर्म जोड़ना, नया मॉड्यूल जोड़ना, user control जोड़ना, नई फाइल जोड़ना, डाटाबेस जोड़ना आदि काम किये जाते हैं। इसमें चालू प्रोजेक्ट की प्रापर्टी सेट की जा सकती है।
- e. **Build** : इस मेन्यू में अप्लिकेशन निर्माण के विकल्प दर्शाये हैं। इसमें अप्लिकेशन का निर्माण, पुनःनिर्माण, आदि विकल्प हैं।
- f. **Debug** : इस मेन्यू में प्रोग्राम को जाँचने के लिए विकल्प है। इस मेन्यू का प्रयोग प्रोग्राम में गलतियों को ढुंढने के लिए होता है।
- g. **Data** :- इस मेन्यू में अप्लिकेशन के साथ डाटा बेस जोड़ने के विकल्प उपलब्ध हैं। इसमें क्युरी विकल्प भी है, जब हम डाटाबेस डिजाइन करते हैं, तब इसके विकल्प का प्रयोग कर इच्छित डाटा से query बना सकते हैं। इसमें sql के सिन्टेक्स जाँचने के विकल्प है। इस मेन्यू में डाटा संरचना सेट कर सकते हैं। filter विकल्प से इच्छित डाटा चुन सकते हैं।
- h. **Format** :- इस मेन्यू के विकल्प द्वारा फॉर्म के विभिन्न घटकों का संयोजन कर सकते हैं। इसके विकल्पों द्वारा आब्जेक्ट का आकार, alignment, आदि सेट कर सकते हैं। इसके lock object विकल्प से किसी आब्जेक्ट को स्थिर किया जाता है।
- i. **Tools** :- इस मेन्यू वी.बी. डॉट नेट के विभिन्न टूल के विकल्प है। इसमें menu editor विकल्प से आप प्रोजेक्ट में मेन्यू संरचना बना सकते हैं। इसमें activex कन्ट्रोल को सेट कर

सकते हैं। इसमें नये प्रोसीजर को जोड़ना, कार्यान्वित करना आदि कार्य किये जाते हैं।

- j. window :- इस मेन्यू के विकल्पों का प्रयोग विंडो को split करने के लिए होता है।
- k. Community :- इस मेन्यू का उपयोग इच्छित कार्य के बारे में इंटरनेट पर विचार विमर्श करने के लिए होता है।

### Toolbox

वी.बी. डॉट नेट में एक टूलबाक्स है, जिसे फॉर्म डिजाइन के समय कंट्रोल डालने के लिए प्रयोग किया जाता है। इसमें उपलब्ध टूल के अतिरिक्त आप अपने काम के अनुसार टूल चालू या बंद कर सकते हैं। प्रत्येक टूल के बारे में हम अगले अध्याय में विस्तार से देखेंगे। इसमें फॉर्म डिजाइन के विभिन्न घटक एवं कंट्रोल हैं। यह टूलबाक्स जब फॉर्म डिजाइन का कार्य करते हैं, तब ही दृश्य होता है। स्क्रीन के बायें ओर टूलबाक्स का टैब दिखाई देता है, उसे क्लिक करने पर यह सामने आता है। इसमें दिये scroll bar की सहायता से, नीचे के टूल प्राप्त कर सकते हैं। जिस टूल से कार्य करना है, उसे क्लिक करे, तथा उसे फॉर्म पर इच्छित जगह रखे। साधारणतः प्रत्येक टूल के अनुसार पाईटर का आकार दर्शाया जाता है। कुछ टूल सीधे फॉर्म पर आ जाते हैं, तथा कुछ टूल को क्लिक करने के बाद डायलॉग बाक्स दिखाई देता है, उस डायलॉग बाक्स में इच्छित मान सेट करने के बाद, वह टूल फॉर्म पर आता है। एक बार फॉर्म पर टूल लाने के बाद आप फॉर्म पर उसे आसानी से एक जगह से दूसरे जगह ले जा सकते हैं। टूल बाक्स के टूल को उनके कार्य के अनुसार विभिन्न हिस्सों में बांटा गया है। टूलबाक्स में निम्न हिस्से होते हैं, जिन्हें आप कार्य के अनुसार दृश्य या अदृश्य कर सकते हैं।



i. common control : इस हिस्से में फॉर्म में बहुतायत प्रयोग होने वाले टूल की सूची है। जैसे selection tool, command button आदि। इसे साधारणतः दृश्य ही रखा जाता है।

ii. Container :- इस भाग में एक से अधिक टूल के साथ कार्य करने वाले टूल को रखा है।

iii. Menu & toolbar :- टूलबाक्स के इस हिस्से में मेन्यू एवं टूलबार बनाने एवं उससे संबंधित टूल को रखा है।

iv. Data:- फॉर्म में डाटा बेस एवं रिकार्ड के लिए उपयुक्त टूल इस हिस्से में दर्शाये जाते हैं।

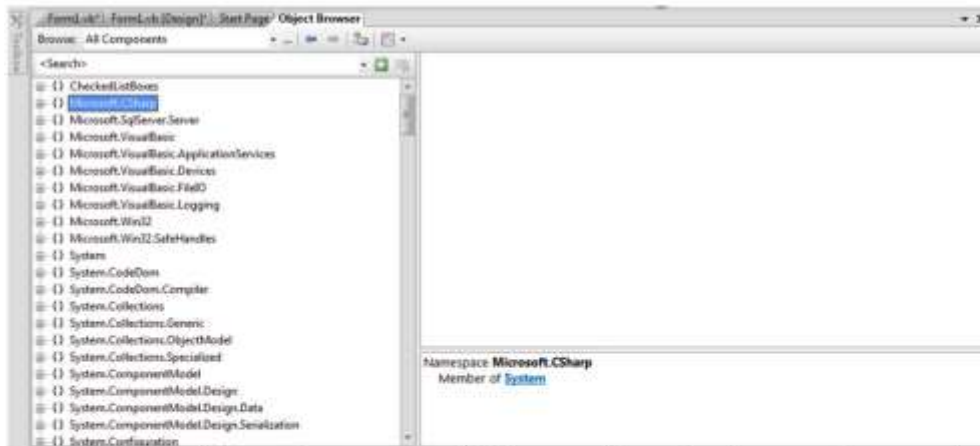
v. Component :- विंडो आधारित अप्लिकेशन के लिये आवश्यक टूल इस हिस्से में दर्शाये जाते हैं।

vi. Printing :- प्रिंटिंग से संबंधित टूल इस हिस्से में दर्शाये जाते हैं।

vii Dialog : विभिन्न डायलॉग बॉक्स बनाने के लिए आवश्यक टूल इस हिस्से में दर्शाये जाते हैं।

इच्छित टूल समूह को दृश्य करने के लिए “+” चिन्ह को क्लिक करें, तथा अदृश्य करने के लिए “-” चिन्ह को क्लिक करें।

## Object browser



इस विंडो में बायीं तरफ प्रोजेक्ट के अंतर्गत सभी आब्जेक्ट, क्लास (class), माड्युल (module) की सूची दर्शाई जाती है, तथा दायीं तरफ चुने हुए आब्जेक्ट की प्रापर्टी दर्शाई जाती है। इस विंडो को चालू करने के लिए view मेन्यू को क्लिक करे, तथा “object browser” विकल्प को क्लिक करे या “F2” बटन दबाये। इस विंडो में हम किसी इच्छित आब्जेक्ट, क्लास आदि की प्रापर्टी को परिवर्तित कर सकते है। इसके अतिरिक्त ऊपर की ओर कुछ बटन होते है। जिसकी सहायता से हम इच्छित आब्जेक्ट को आसानी से देख सकते है।

### vb.net के tab

vb.net में कार्य करने के environment के विभिन्न सेट करने के लिए विभिन्न टैब का प्रयोग किया जाता है। इसे चालू करने के लिए निम्न पद है

- Tools मेन्यू को क्लिक करे।
  - Option विकल्प को क्लिक करे। आपको विभिन्न टैब के साथ डायलाग बाक्स दिखाई देता है, इसमे निम्न टैब है
1. Editor tab

2. Editor format tab
3. General tab
4. Environment tab
5. Docking tab
6. Advanced tab

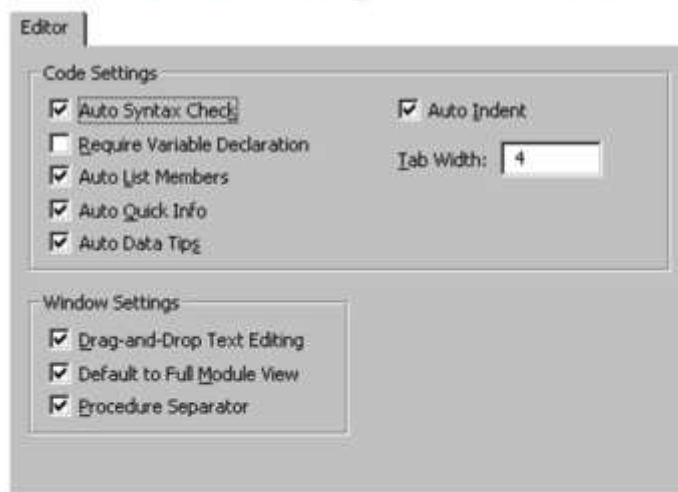
## Editor tab

यह टैब में कोड विंडो तथा प्रोजेक्ट के विभिन्न घटकों के मान सेट किये जा सकते हैं। इसमें निम्न विकल्प हैं

**Auto Syntax check :-** vb.net में कोड को लिखने का एक तरीका होता है, यदि कोड निश्चित तरीके से नहीं लिखा हो, तब प्रोग्राम कार्य नहीं करता है। यदि इस विकल्प को चालू किया जाये तब vb.net स्वयं ही कोड का सिन्टेक्स जांच लेता है। यदि संभव है, तब स्वयं ही उसे सुधार भी लेता है।

**Request variable declaration :-** यदि इस विकल्प को चालू किया जाता है, तब कोई मॉड्यूल में **explicit variable** की आवश्यकता है, या नहीं इसकी जांच करता है। इसे चालू करने से vb.net "option explicit" स्टेटमेंट को जोड़ देता है।

**auto list member :-** यह एक महत्वपूर्ण सुविधा है, जब आप कोई स्टेटमेंट टाईप करते हैं, तब उसे स्टेटमेंट को vb.net स्वयं ही पूर्ण करने का प्रयास करता है,। यदि विकल्प को चालू करते हैं, तब vb.net आपको उस स्टेटमेंट को पूर्ण करने के



आवश्यक keyword की सूची दर्शाता है, आप उनमें से इच्छित keyword सिलेक्ट कर सकते हैं। उदाहरण यदि आपने

Dim ab as

टाईप किया है, तब vb.net आपको सूची दर्शाता जिसमें विभिन्न घटक होते हैं, यदि आपको integer सेट करना है, तब int टाईप करने के बाद "integer" दिखाई देता है, उस शब्द को स्टेटमेंट में डालने के लिए enter बटन दबायें। इस विकल्प को साधारणतः चालू ही रखा जाता है।

Auto quick info :- यदि इस विकल्प को चालू रखते हैं, तब vb.net टाईप किये फंक्शन की जानकारी एवं उसके विभिन्न परिमाण (parameter) दर्शाता है।

Auto Data tips :- यदि इस विकल्प को चालू रखते हैं, तब break mode में vb.net जिस वेरिएबल के उपर आप पाईंटर रखते हैं, उसका मान दर्शाया जाता है।

auto indent :- इस विकल्प से vb.net स्वयं ही स्टेटमेंट को टैब स्पेस देकर एक सीध में रखता है। यह कंडीशनल स्टेटमेंट या loop स्टेटमेंट अधिक उपयोगी है।

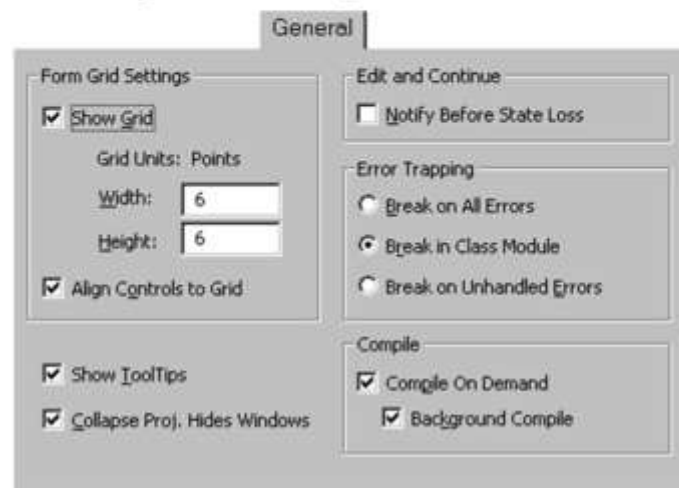
drag and drop text edition :- टेक्स्ट को कॉपी पेस्ट करने के लिए यदि "drag and drop" पद्धति का भी उपयोग करना है, तब इस विकल्प को चालू रखें। लेकिन यह सुविधा सिर्फ immediate एवं watch विंडो में ही उपलब्ध रहती है।

default to full module view :- नये मॉड्यूल का default स्टेट सेट करता है। कोड विंडो में या तो एक ही प्रोसीजर के कोड दिखते हैं, या सभी प्रोसीजर के कोड दिखते हैं। यदि इस विकल्प को चालू रखा जाता है, तब सभी प्रोसीजर के कोड दिखते हैं।

**Procedure separator :-** इस विकल्प से प्रत्येक प्रोसीजर के बाये ओर “+” या “-” चिन्ह दर्शाया जाता हैं। जिससे इच्छित प्रोसीजर के कोड को दृश्य या अदृश्य किया जा सकता है।

## General Tab

### Form grid settings



जब कोई फार्म डिजाइन सेट करते हैं, तब उसमें बहुत से कंट्रोल, टेक्स्ट आदि डाले जाते हैं। उन सभी घटकों को सही तरीके से सेट करने के लिए फार्म डिजाइन के समय एक ग्रीड दर्शाई जाती है। वह ग्रीड सिर्फ

डिजाइन के समय ही दिखाई देती है, जब प्रोग्राम को चलाया जाता है, तब वह ग्रीड नहीं दर्शाई जाती है। इस ग्रीड को इस में सेट किया जाता है

**show grid :-** फार्म डिजाइन के समय ग्रीड को दृश्य या अदृश्य करने के लिए इस चेक बाक्स का प्रयोग किया जाता है।

**grid unit :-** ग्रीड के दो लाइन या बिंदुओं के बीच की दूरी इस विकल्प में सेट की जाती है।

**align controls to grid :-** जो कंट्रोल हम फार्म में डालते हैं, वह स्वयं ग्रीड के अनुसार सेट करना है, तब इस विकल्प को चालू करें।

**show tool tips :-** जग किसी टूल पर पाईटर ले जाते हैं, तब उससे संबंधित संक्षिप्त जानकारी दर्शाने के लिए इस विकल्प को चालू किया जाता है



**collapse porj. hide windows** :- जब कोई प्रोजेक्ट बंद किया जाता है, तब उससे संबंधित सभी विंडो स्वयं ही बंद या अदृश्य हो जाती है।

**notify before state loss** :- प्रोजेक्ट चलते समय, किसी भी कारण से, कोई माडूल लेवल के वेरिएबल का मान **reset** होता है, तब उसका चेतावनी संदेश दर्शाना या नहीं दर्शाना इसके लिए इस विकल्प का प्रयोग किया जाता है।

**Break in all error** :- प्रोजेक्ट चलते समय किसी भी प्रकार की गलती प्राप्त होने पर प्रोजेक्ट रुक जाता है।

**break on class module** :- क्लास माडूल में किसी भी गलती पर प्रोजेक्ट का क्रियान्वयन रुक जाता है।

**Break on unhandled errors** : जब कोई error होती है, लेकिन उसका संदेश नहीं दर्शाया जाता है, उसे **unhandled error** कहा जाता है। इस प्रकार के error में प्रोग्राम का क्रियान्वयन रोकना है, तब इस विकल्प को चालू किया जाता है।

**compile on demand** :- यदि इस विकल्प को बंद किया जाता है, तब प्रोजेक्ट के सभी कोड का क्रियान्वयन होने के पहले उसे **compile** किया जाता है। यदि इस विकल्प को चालू रखा जाता है, तब सिर्फ वोही कोड कम्पाइल होता है, जिसके क्रियान्वित करना है,। इस विकल्प को चालू रखने से अप्लिकेशन तेजी से चालू हो सकती है।

**background compile** :- यदि इस विकल्प को चालू रखा जाता है, तब प्रोग्राम के क्रियान्वयन के समय, **vb.net** को खाली समय मिलता है, उसमे वह आगे के कोड को कम्पाइल कर लेता है।

## **Debugging window**

कभी कभी आपको प्रोग्राम के किसी एक हिस्से के क्रियान्वयन में समस्या आती है। तब उस हिस्से को अलग कर उसमे वेरिएबल के

मान या प्रापर्टी जाँच कर उस समस्या हो हल किया जा सकता है। प्रोग्राम के एक हिस्से को क्रियान्वित करने के लिए **debugging window** का प्रयोग किया जाता है। **debugging window** में आप किसी वेरिएबल या **expression** के मान जाँच सकते हैं, वी.बी. डॉट नेट में तीन प्रकार की **debugging window** होती है।

- i. **Command Window** :- यह विंडो स्क्रीन के सबसे नीचे की ओर दिखाई देती है। यह विंडो प्रोग्राम को जाँचने के लिए प्रयोग होती है। इसमें प्रोग्राम एक के बाद एक क्रियान्वित होता है, जिससे यह पता चल सकता है, की गलती कहाँ हो रही है। इस विंडो में प्रोग्राम क्रियान्वित होते समय **variable** के मान बदल सकते हैं। यह विंडो जब आप कोई एप्लिकेशन IDE में चला रहे हैं, तब ही चालू होती है।
- m. **output window** :- इस विंडो में प्रोग्राम के क्रियान्वयन के समय आप जिस **expression** को देखना चाहते हैं, उस **expression** के मान दर्शाये जाते हैं। इस विंडो में चार कॉलम होते हैं। पहले कॉलम में **expression** का मान, दूसरे कॉलम आप की दि हुई कंडीशन सही क्रियान्वित हो रही है, तब **true** दर्शाता है, अन्यथा संदेश दर्शाता है। तीसरे कॉलम में वेरिएबल का टाईप दर्शाया जाता है। तथा चौथे कॉलम में वह **expression** किस प्रोसीजर या मॉड्यूल में है, वह दर्शाया जाता है।
- n. **local window** :- चालू प्रोसीजर के अंदर के वेरिएबल का मान दर्शाया जाता है। जब चाजु प्रोसीजर के अंदर के वेरिएबल का मान बदलता है, उस प्रत्येक बदलाव को इस विंडो में दर्शाया जाता है।

## Concept of Project

प्रोजेक्ट यह प्रोग्राम या सॉफ्टवेयर बनाने को कहते हैं। कोई भी सॉफ्टवेयर बनाने के लिए निम्न पदों का प्रयोग किया जाता है।

1. प्रोजेक्ट कि outline बनाना, जो प्रयोगकर्ता के लिए काम करने में आसान हो।
2. प्रोग्राम का कोड लिखना, जो उस प्रोग्राम के विभिन्न कार्य करने के लिए है।
3. संबंधित प्रोसीजर को बनाना एवं उसे प्रोग्राम में वांछित जगह call करना।
4. प्रोग्राम को लिखने के बाद उसे विभिन्न स्थिति में जाँचना।
5. प्रोग्राम को चलाना
6. उसकी executable फाइल बनाकर उसे इच्छित जगह लोड करना

## Element of project

**Forms** :- यह एक प्रकार की विंडो होती है, जिसमें विभिन्न GUI का प्रयोग किया जाता है, प्रयोगकर्ता को प्रोजेक्ट में सिर्फ यही हिस्सा दिखाई देता है, तथा वह सिर्फ इसी में काम करता है।

**Modules** :- माड्युल् यह कोड एवं डाटा का समूह होता है, जो एक विशिष्ट कार्य करने के लिए होता है। एक माड्युल में प्रोसीजर एवं डाटा को इसी तरह सम्मिलित किया जाता है, कि वह प्रोग्राम चलते समय दिखाई नहीं देते हैं।

**Global Item**: - global item यह एक प्रोजेक्ट के सभी फॉर्म एवं माड्युल में उपलब्ध रहते हैं। आप उन्हें public कीवर्ड से declare कर सकते हैं।

## Interactive development

पारंपारीक एप्लिकेशन बनाने में मुख्यतः तीन पदों का प्रयोग होता है, 1. प्रोग्राम लिखना 2. कम्पायल करना 3. कोड को जाँच कर ठीक करना। लेकिन वी.बी. डॉट नेट भाषा में इन तीनों क्रियाओं की पारस्परिक क्रिया का प्रयोग होता है। बहुतसी प्रोग्रामिंग भाषा में यदि प्रोग्राम में कोई गलती है, तब वह गलती या त्रुटि कम्पायलेशन (compilation) के समय ही ज्ञात होती है। आपको उसे सुधार कर प्रोग्राम को फिरसे चला कर कम्पाइल करना पड़ता है। यह कार्य जब तक सभी त्रुटियों निकालते नहीं तब तक चलता है। लेकिन वी. बी. डॉट नेट में जब आप कोड डालते हैं, तब ही वह उसे इंटरप्रीट करता है, तथा उसमें अधिकांश गलतियों को दर्शाता है। गलतियों को दर्शाते समय वी.बी. डॉट नेट प्रोग्राम को कम्पाइल भी करता है। जब आप एप्लिकेशन चलाने की दशा में होते हैं, तब कम्पाइलर गलती हो तो हायलाइट करता है। आप उस गलती को सुधार कर आगे प्रोग्राम का कम्पायलेशन चालू रख सकते हैं, आपको पहले से प्रोग्राम चलाने की आवश्यकता नहीं होती है। इस सुविधा के कारण आप प्रोग्राम को बनाते समय भी चला कर देख सकते हैं। एवं बहुत बड़े एप्लिकेशन में यह सुविधा बहुत महत्वपूर्ण होती है।

## Method

method यह किसी class का एक member होता है, जिसमें किसी कार्य की निर्देश लिखे होते हैं। method का प्रयोग किसी वेरिएबल या ऑब्जेक्ट को प्राप्त करने के लिए भी होता है। उदाहरण के लिए हम एक employee नाम का class लेते हैं, जिसमें payment और appointment नाम के दो मेथड बनाये हैं।

```
public class employ
    public sub payment()
        MsgBox("Payment")
```

```
end sub
```

```
public sub appointment ()  
    MsgBox("Appointment")  
end sub
```

अब एक नया ऑब्जेक्ट बनाते हैं, जिसमें `employee` क्लास एवं `appointment` मेथड का प्रयोग किया है।

```
Dim production as new employee()  
production.appointment()
```

हम एक क्लास के अंदर इच्छित फंक्शन या सब प्रोसीजर को जोड़ सकते हैं।

### Event driven programming

Procedural प्रोग्रामिंग में कौन सा कोड किस क्रम में चलना चाहिए यह प्रोग्राम स्वयं नियंत्रित करता है। प्रोग्राम के क्रियान्वन यह पहले लाइन से चालू होती है, एवं दिये गये निर्देशों के चलती जाती है। लेकिन `event driven programming` प्रोग्राम के कोड सिर्फ एक दिए गये रस्ते (`path`) पर नहीं चलते, अपितु अलग अलग कोड किसी कार्य (`event`) पर क्रियान्वित होते हैं। उदाहरण के लिए हम एक `calculator` को लेते हैं, इसमें जब हम कोई बटन दबाते हैं, तब ही वह नंबर `calculator` के स्क्रीन पर आता है, तथा जो क्रिया ( `+`, `-`, `x`, आदि ) देते हैं वही क्रिया होती है। `Event` प्रयोगकर्ता के कार्य करने पर, या प्रणाली पर या स्वयं एप्लिकेशन पर निर्भर हो सकता है। `event` का क्रम यह कोड के क्रियान्वित होने के क्रम को नियंत्रित करता है। इसलिए किसी एप्लिकेशन में कोड का क्रियान्वित का क्रम अलग अलग हो सकता है। `event driven programming` में आप `event` का क्रम निश्चित नहीं कर

सकते। लेकिन एक अच्छा एप्लिकेशन बनाने के लिए आपको event के क्रम का अनुमान लगाना आवश्यक होता है। उदाहरण के लिए किसी salary का गणना तब तक नहीं होनी चाहिए, जब तक basic salary फ़िल्ड में कुछ डाटा न डाला हो। आपको कोई एप्लिकेशन बनाते समय यह ध्यान रखना जरूरी है, कि आपके अनुमान (assumption) वैध (valid) हो।

यह आवश्यक नहीं है, कि कोई event सिर्फ प्रयोगकर्ता ही कर सकता है, अपितु वह प्रोग्राम के द्वारा भी क्रियान्वित हो सकता है, जैसे Ms-word में जब आप enter की बटन दबाते हैं, तब कर्सर नीचे के लाइन पर आ जाता है। लेकिन जब आपका कर्सर पेज के अंत में होता है, तब वह दूसरे पेज में चले जाता है। कुछ event यह एक निश्चित अवधि में क्रियान्वित होते हैं, जैसे ms-word में auto-save से दस्तावेज स्वयं ही कुछ देर बाद save हो जाता है।

वी.बी. डॉट नेट में सामान्यता: दो प्रकार के event होते हैं।

1. Mouse द्वारा क्रियान्वित होने वाले event

2. keyboard द्वारा क्रियान्वित होने वाले event.

वी.बी. डॉट नेट में माउस द्वारा अधिकांश event क्रियान्वित किये जाते हैं। इसके अंतर्गत click, double click, right click, mouse up, mouse down, mouse move आदि event हो सकते हैं। इसमें जब प्रयोगकर्ता माउस की बटन दबाता है, तब mouse up event होता है, तथा जब प्रयोगकर्ता माउस की बटन छोड़ता है, तब mouse down event होता है। जब प्रयोगकर्ता माउस को किसी कंट्रोल के ऊपर से मूव करता है, तब mouse move event होता है।

keyboard में key down , key up, key press आदि event आते हैं। इसके अतिरिक्त फॉर्म के लिए load इस event का प्रयोग

होता है। इसके अलावा कुछ event यह किसी कार्य के बाद क्रियान्वित होते हैं। जैसे

**change** :- यह event जब किसी कंट्रोल के घटक को बदलते हैं, तब क्रियान्वित होते हैं। उदाहरण के लिए टेक्स्ट बाक्स में इस event को लागू किया है, यह तब ही क्रियान्वित होता है, जब प्रयोगकर्ता उस टेक्स्ट बाक्स में कोई टेक्स्ट डालता है।

**Focus** :- यह event जब किबोर्ड से काम कर रहे हैं, तब महत्वपूर्ण हो जाता है। किस कंट्रोल पर यह क्रियान्वित होना है, यह इस event से निश्चित किया जाता है। उदाहरण के लिए आपके फॉर्म पर दो टेक्स्ट बाक्स हैं, जब प्रयोगकर्ता पहले टेक्स्ट बाक्स में टेक्स्ट डाल रहा है, तब वह पहला टेक्स्ट बाक्स focus होता है। जब प्रयोगकर्ता एंटर या टैब key बटन प्रेस करता है, तब दूसरे टेक्स्ट बाक्स पर focus जाता है। जिस कंट्रोल पर कर्सर डालना है, उसे **gotFocus** इस event से लागू किया जाता है। जब किसी कंट्रोल से कर्सर निकालना है, तब **lostFocus** इस event का प्रयोग किया जाता है।

अलग अलग कंट्रोल पर अलग अलग event प्रयोग किये जाते हैं, उदाहरण command बटन पर **Click** इस event का प्रयोग अधिकतर होता है। टेक्स्ट बाक्स पर **gotfocus** इस event का प्रयोग होता है।

निम्न कोड में टेक्स्ट बाक्स के डाटा का रंग माउस के क्लिक से बदलता है।

```
Private Sub txtID_MouseEnter(sender As Object, e As  
EventArgs)_  
    Handles txtID.MouseEnter  
    'code for handling mouse enter on ID textbox  
    textBox1.BackColor = Color.CornflowerBlue
```

```

    TextBox1.ForeColor = Color.White
End Sub
    
```

Vb.net में निम्न method प्रयोग होती है

Close	फॉर्म को बंद करने के लिए
Show	फार्म को दर्शाने के लिए प्रयोग होता है
Refresh	स्क्रीन के घटकों को फिर draw कर, स्क्रीन में current घटक दर्शाने के लिए प्रयोग होता है।
addownform	प्रयोगकर्ता की ओर से नया फार्म अप्लिकेशन में जोड़ने के लिए
Findstring	लिस्ट बाक्स के सूची में से, प्रयोगकर्ता ने दिये गये string के आधार पर घटक दर्शाने के लिए प्रयोग होता है। उदाहरण के लिए city नाम की लिस्ट बाक्स विभिन्न शहरों के नाम दिये है, यदि प्रयोगकर्ता "Na" टाईप करता है, तब "Na" से चालू होने वाले सभी शहरों को सूची में दर्शाया जाता है।
GetitemHeight	लिस्ट बाक्य के प्रत्येक घटक की height सेट करने के लिए इस मेथड का प्रयोग होता है।

Event

closed	जब फॉर्म बंद होता है, तब यह event कार्यरत होता है।
keydown	जब किबोर्ड पर कोई key दबाते है
keyup	जब दबाई गई key छोडते है,
change	यह event जब किसी कंट्रोल के घटक को



	बदलते हैं, तब क्रियान्वित होते हैं।
mousedown	जब माउस की बटन छोड़ते हैं
mouseover	जब माउस का पाईंटर किसी ऑब्जेक्ट के उपर से गुजरता है।
click	जब माउस की बटन click करते हैं

### वी.बी. डॉट नेट में काम करना

वी.बी. डॉट नेट में जो एप्लिकेशन बनाये जाते हैं, उसे **project** कहा जाता है। या हम **project** को किसी एप्लिकेशन को बनाने के लिए लगने वाली फाइलों का समूह भी कह सकते हैं। जब कोई नई एप्लिकेशन बनाते हैं, तब साधारणतः आप नया फॉर्म डिजाइन करते हैं, लेकिन आप पुराने फॉर्म का भी प्रयोग नई एप्लिकेशन में कर सकते हैं। इसी तरह दूसरी फाइल का कोड, **ActiveX** कंट्रोल, आब्जेक्ट का प्रयोग अपने प्रोजेक्ट में कर सकते हैं।

एक प्रोजेक्ट फाइल सिर्फ सभी फाइलों और आब्जेक्ट की सूची होती है, जो उस प्रोजेक्ट से संबंधित है। साथ ही उसमें **environment** के विकल्प निश्चित किये जाते हैं। जब कोई प्रोजेक्ट संपूर्ण तरीके से बन जाता है, तब उसे **.exe** (executable) फाइल में परिवर्तित किया जाता है। यह **.exe** फाइल आप सीधे विंडो में चालू कर सकते हैं।

### वी.बी. डॉट नेट चालू करना

- विंडो में "Start" पर क्लिक करें
- "Program" बटन पर क्लिक करें
- "Microsoft Visual studio 2008" पर क्लिक करें।
- "Microsoft Visual Basic 2008" पर क्लिक करें।

अब आपको निम्न स्क्रीन दिखाई देगी।

यदि नई फाइल बनाना है, तब "New Project" बटन पर क्लिक करें। यदि आपको पुराने फाइल में काम करना है, तब "Existing" बटन पर क्लिक करें। तथा "Recent" बटन से अभी प्रयोग की गई फाइल की सूची आती है।

• Standard Exe बटन पर क्लिक करें

• Open बटन पर क्लिक करें

अब आपको मुख्य स्क्रीन दिखाई देगी जिसमें एक फॉर्म स्वयं ही खुल जाता है।

वी.बी. डॉट नेट में प्रोग्राम बनाने के लिए मुख्यतः दो पद हैं। 1. फॉर्म को डिजाइन करना 2. कोड लिखना। वी.बी. डॉट नेट में विभिन्न प्रकार के एप्लिकेशन को बना सकते हैं।

नरेन्द्र पब्लिकेशन

## Programming

कोई भी प्रोग्रामिंग भाषा में विभिन्न स्थितियों के अनुसार प्रोग्राम की दिशा निश्चित होती है, विभिन्न स्थितियों को दर्शाने के लिए प्रोग्रामिंग भाषा में स्वयं का एक तरीका होता है। वी.बी. डॉट नेट में कंडीशन के अनुसार प्रोग्राम में कार्य करने के लिए विभिन्न सिन्टेक्स दिये हैं। कार्य को वी.बी. डॉट नेट के तरीके से लिखने को कोड (code) कहा जाता है, एक प्रोग्राम में बहुतसे कोड हो सकते हैं। यह कोड, प्रोग्राम की जटिलता एवं प्रोग्रामर की कुशलता पर निर्भर होते हैं। विजुअल बेसिक में एक कार्य के कोड का समूह होता है, कोड का समूह विभिन्न स्टेटमेंट से बना होता है। उदाहरण के लिए हमें दो अंकों का जोड़ का प्रोग्राम बनाना है, तब हम सरल भाषा में निम्न तरीके से लिखते हैं

$A + B = C$

लेकिन इसी कार्य को विजुअल बेसिक में लिखने के लिए उसके सिन्टेक्स का प्रयोग करना होगा। वी.बी. डॉट नेट में दो अंकों के जोड़ के लिए निम्न तरीके से लिखते हैं।

```
dim a,b, c as integer
```

```
Private Sub add()
```

```
    c = a + b
```

```
end sub
```

उपरोक्त को वी.बी. डॉट नेट का कोड कहा जाता है, तथा “dim a,b,c as integer” आदि लाइन को स्टेटमेंट कहा जाता है। कोड के प्रत्येक लाइन को स्टेटमेंट कहा जाता है। एक स्टेटमेंट में बहुतसे घटक होते हैं। इसमें कुछ वी.बी. डॉट नेट के कमांड, कुछ वेरिएबल के नाम, ऑपरेटर आदि होते हैं। जो शब्द या शब्द समूह वी.बी. डॉट नेट के तरफ से दिये जाते हैं, उन्हें visual basic keyword कहा जाता है। उपरोक्त उदाहरण में dim, integer,

private, sub, end आदि वी.बी. डॉट नेट के कीवर्ड है, इन्हे इसी तरह लिखा जाता है, इनकी स्पेलिंग आदि में कोई भी बदलाव वी. बी. डॉट नेट मान्य नहीं करता है। यदि गलत स्पेलिंग लिखी है, या गलत जगह कीवर्ड का प्रयोग किया है, प्रोग्राम क्रियान्वित नहीं होता है।

इसके अतिरिक्त स्टेटमेंट में वेरिएबल का प्रयोग किया जाता है, वेरिएबल के नाम प्रयोगकर्ता स्वयं के अनुसार दे सकता है, इसके नाम, स्पेलिंग आदि प्रयोगकर्ता स्वयं तय करता है। उपरोक्त उदाहरण में "a, b, c" यह वेरिएबल के नाम है, इनके जगह हम यदि x, y, z रख सकते हैं, तब भी प्रोग्राम का परिणाम एक समान ही होगा।

स्टेटमेंट में ऑपरेटर का भी प्रयोग किया जाता है, ऑपरेटर वेरिएबल के बीच के संबंधों को दर्शाता है। उपरोक्त उदाहरण में "+" (जोड़) यह ऑपरेटर है

इसके अतिरिक्त स्टेटमेंट में constant, expression का भी प्रयोग किया जाता है।

निम्न कुछ शब्द वी.बी. डॉट नेट के कीवर्ड हैं,

char	by	call	Cdate	dim	do	each
	val					
for	frien	exit	false	if	intege	interfa
	d				r	ce
long	let	lib	mycla	namespa	option	or
			ss	ce	al	
obje	sele	set	privat	readonly	then	sub
ct	ct		e			
thro	until	writen	while	string	typeof	structu
w	ly					re

## स्टेटमेंट का सिन्टेक्स

यद्यपि प्रत्येक स्टेटमेंट अलग कार्य को पूर्ण करने के लिए होता है, इसलिए उसकी बनावट भी उसके कार्य के अनुसार होती है। सिन्टेक्स के कुछ घटक आवश्यक होते हैं, तथा कुछ घटक वैकल्पिक होते हैं, सिन्टेक्स में कीवर्ड को अलग फारमेट में दर्शाया जाता है, तथा वेरिएबल एवं वैकल्पिक परिमाण को अलग प्रारूप में दर्शाया जाता है। सधारणतः कीवर्ड को उसी टाईप में लिखा जाता है, तथा वैकल्पिक को [ ] बैकेट में दर्शाया जाता है

उदाहरण

```
[<attrlist>] [{public | protected | friend| protected friends |  
private | static}] [shared] [readonly]  
Dim [withevent] name [boundlist] [As [new] type] [=initiesper]
```

इसमें

dim यह कीवर्ड इस इसी तरह लिखा जाता है

name यह वेरिएबल का नाम है, जिसे प्रयोगकर्ता कार्य के अनुसार लिख सकता है

अन्य परिमाण यह वैकल्पिक है

## Data types in Visual Basic

प्रोग्राम में प्रायः गणना करने के लिए कोई मान (value) अस्थायी तौर पर संग्रहित करना पड़ता है। वी.बी. डॉट नेट भाषा में अन्य प्रोग्रामिंग भाषा के समान ही मान को स्टोर करने के लिए वेरिएबल (variable) का प्रयोग किया जाता है। प्रत्येक वेरिएबल का एक नाम होता है, एवं उसका data type (किस प्रकार का मान है) रहता है। एक से ज्यादा मान एक वेरिएबल में संग्रहित करने के लिए Arrays का प्रयोग किया जाता है। कोई भी वेरिएबल का मान यह अस्थायी रहता है, तथा constant का मान यह स्थायी होता है। वी.बी. डॉट नेट में विभिन्न प्रकार के डाटा टाईप प्रयोग है।

वेरिएबल यह प्रोग्राम को क्रियान्वित करते समय अस्थायी तौर पर संग्रहित रहते हैं। उदाहरण के लिए हमें दो अंको के जोड़ के लिए प्रोग्राम बनाना है। तब हम दो वेरिएबल का प्रयोग करते हैं, जैसे A और B, प्रत्येक बार जब हम प्रोग्राम चलाते हैं, तब हम दो मान (value) उन वेरिएबल में डालते हैं। उनका जोड़ हमें मिलता है। प्रत्येक बार हम A और B में अलग अलग मान डाल सकते हैं।

$C = A + B$

प्रत्येक बार प्रोग्राम चलाने पर हमें C का मान अलग अलग मिल सकता है। C का मान यह A और B के मान पर निर्भर है। जिनके वास्तविक इनपुट हमें पहले से ज्ञात नहीं होते हैं, वेरिएबल उन की गणना करने की सुविधा प्रदान करता है।

### Declaring constant

ऐसे मान जिनका मूल्य संपूर्ण प्रोजेक्ट में एक ही रहता है, उसे constant कहा जाता है, या दूसरे शब्द में हम यह कह सकते हैं, कि constant का मान कभी भी नहीं बदलता है। vb.net में किसी constant को एक नाम के साथ परिभाषित कर सकते हैं, तथा प्रोजेक्ट में विभिन्न कार्यों में उस constant की जगह उस नाम का प्रयोग कर सकते हैं। vb.net में constant घोषित करने के लिए "Const" कीवर्ड का प्रयोग किया जाता है।

उदाहरण

```
const pia = 3.1415
```

### वेरिएबल को घोषित करना (Declaring Variable)

वी.बी. डॉट नेट में वेरिएबल के नाम के निम्न नियम हैं।

1. वेरिएबल के नाम की लंबाई अधिकतम 255 कैरेक्टर होती है।
2. कोई भी वेरिएबल यह कैरेक्टर से चालू होता है, तथा उसके बाद नंबर आ सकते हैं। लेकिन कोई भी वेरिएबल यह नंबर से चालू नहीं होता है।

3. वेरिएबल के नाम के बीचमे स्पेस नही होनी चाहिए। तथा    (underscore) के अतिरिक्त दूसरा स्पेशल कैरेक्टर का प्रयोग नही किया जा सकता

वी.बी. डॉट नेट में वेरिएबल को घोषित करने के कई तरीके है। यह इस बात पर निर्भर करता है, कि वेरिएबल कहाँ और कैसे कार्य करते है।

वी.बी. डॉट नेट में वेरिएबल घोषित करने के लिए Dim स्टेटमेंट का प्रयोग किया जाता है।

सिन्टेक्स

Dim वेरिएबल का नाम As प्रकार

उदाहरण

*Dim A as integer*

इसमे A नाम का वेरिएबल घोषित किया है, जिसका डाटा टाईप integer है।

Dim यह Dimension का शार्टफॉर्म है। जो वेरिएबल Dim स्टेटमेंट से घोषित किये जाते है, वह उस procedure मे सिर्फ उसी समय तक रहते, जब तक वह प्रोसीजर चालू रहती है। जैसे हि वह क्रिया पूर्ण हो जाती है, वैसी ही वह वेरिएबल खत्म हो जाता है। इसमे मान यह Local रहते है, अर्थात इसके मान सिर्फ उसी प्रोग्राम या प्रोसीजर में क्रियान्वित रहते है, वह दूसरे प्रोग्राम में नही चलते। वेरिएबल में Type डालना यह वैकल्पिक है। आप एक dim स्टेटमेंट में एक से ज्यादा वेरिएबल घोषित कर सकते है, लेकिन उन सभी वेरिएबल का डाटा टाईप एक समान होना चाहिए, उदाहरण

*Dim A,B,C As Integer*

वेरिएबल का कार्यक्षेत्र (Scope)

वेरिएबल का कार्यक्षेत्र यह बताता है कि कोई वेरिएबल कोड के कौन से हिस्सों तक अस्तित्व में है। जब किसी प्रोसीजर

(procedure) में वेरिएबल को घोषित करते हैं, तब सिर्फ उस प्रोसीजर के अंदर के कोड ही उस वेरिएबल को प्राप्त कर सकते हैं, या उस वेरिएबल में बदलाव कर सकते हैं। बहुत बार बड़े कार्यक्रम के लिए वेरिएबल बनाना पड़ता है, जिसके मान संपूर्ण एप्लिकेशन के सभी प्रोसीजर में कार्यरत रह सके। वी.बी. डॉट नेट वेरिएबल के कार्यक्रम को तय करने की सुविधा देता है। वेरिएबल का कार्यक्रम यह उसे डिक्लेअर करने के तरीके पर निर्भर है।

कार्यक्षेत्र (scope)	Private	Public
Procedure level	सिर्फ एक ही प्रोसीजर में कार्यरत रहते हैं, जिनमें उस वेरिएबल को व्यक्त किया जाता है।	यह प्रोसीजर लेवल में लागू नहीं है, आप public वेरिएबल को सिर्फ एक प्रोसीजर तक सीमित वेरिएबल नहीं बना सकते हैं।
Module level	सिर्फ एक माड्यूल में क्रियान्वित रहता है, जिस माड्यूल में उसे व्यक्त किया है।	सभी माड्यूल में उपलब्ध रहता है।

वेरिएबल जो सिर्फ एक प्रोसीजर में प्रयोग होते हैं (procedure level variable)

इस प्रकार के वेरिएबल जिस प्रोसीजर में घोषित किये गये हैं, सिर्फ उसी प्रोसीजर में क्रियान्वित रहते हैं। इन्हें Local वेरिएबल भी कहा जाता है। इन्हें Dim या Static स्टेटमेंट से घोषित किया जाता है।

*dim a as integer*

*static val as integer*



**static** में घोषित किये गये वेरिएबल जब तक एप्लिकेशन चालू रहती है, तब तक उस प्रोसीजर में अस्तित्व में रहते हैं। तथा जो वेरिएबल **Dim** में घोषित किये जाते हैं, वह सिर्फ उस प्रोसीजर के चालू रहने तक क्रियान्वित रहते हैं। जब कोई अस्थाई या छोटे कॅल्कुलेशन करना है, तब इस प्रकार के वेरिएबल ज्यादा उपयोगी होते हैं।

माड्युल के अंदर प्रयोग होने वाले वेरिएबल (module level variable)

एक माड्युल के अंदर घोषित किये गये वेरिएबल, उस माड्युल के अंतर्गत सभी प्रोसीजर में क्रियान्वित रहते हैं। इस प्रकार के वेरिएबल घोषित करने के लिए **Private** स्टेटमेंट का प्रयोग किया जाता है, तथा इसे माड्युल के सबसे ऊपरी भाग में घोषित किया जाता है।

*Private num1 as integer*

वैसे माड्युल लेवल में **private** या **dim** में कोई अंतर नहीं है, लेकिन **private** को अधिक वरीयता दी जाती है, क्योंकि **private** स्टेटमेंट से बनाया हुआ वेरिएबल **public** वेरिएबल से बहुत जल्दी अंतर कर लेता है।

## Data Types

प्रत्येक वेरिएबल का एक नाम होता है, एवं उस वेरिएबल में कौन से प्रकार का डाटा प्राप्त होना है यह लिखा होता है। जैसे आपको कोई नाम डालना है, तब उस वेरिएबल का डाटा का प्रकार यह **String** रखना चाहिए, यदि किसी वेरिएबल में हमें मार्क डालना है तब उसके डाटा का प्रकार **Integer** रखा जाता है। प्रत्येक डाटा टाईप एक निश्चित मेमोरी **consume** करती है। यदि आपने किसी वेरिएबल को डाटा टाईप नहीं दिया है, तब वी.बी. डॉट नेट उसे

variant डाटा टाईप समझ कर काम करता है। वी.बी. डॉट नेट में निम्न डाटा के प्रकार उपलब्ध है।

1<sup>प</sup> **Byte** :- यदि किसी वेरिएबल में बायनरी डाटा डालना है, तब इस प्रकार के डाटा टाईप का प्रयोग किया जाता है। जब किसी सिस्टम फाइल को देखना है, या उसमें कुछ लिखना है, तब वी.बी. डॉट नेट ansi को Unicode में स्वयं ही परिवर्तित कर देता है। जो ऑपरेटर integer पर कार्य करते वह सभी byte डाटा टाईप पर भी कार्य करते हैं। इसमें ऋणात्मक संख्या नहीं आती है।

2. **Boolean** :- जब कोई वेरिएबल का डाटा सिर्फ दो ही विकल्पों में रहता है, तथा हम उसमें से किसी एक ही विकल्प का चयन करना पड़ता है, तब इस प्रकार के डाटा टाईप का प्रयोग किया जाता है। जैसे स्त्री या पुरुष, पास या फेल आदि

3<sup>प</sup> **Integer** :- इस प्रकार के डाटा टाईप में सिर्फ पूर्ण अंक आते हैं जैसे 12, 0, -30 आदि, तथा दशमलव नहीं होता है। इस प्रकार का डाटा टाईप सिर्फ उसी समय प्रयोग किया जाता है, जब यह निश्चित रहता है, कि उस वेरिएबल में दशमलव अंक नहीं डाले जाएंगे। उदाहरण विषय में मार्क, छात्रों की संख्या, आयु आदि। long integer का प्रयोग जब बड़ी संख्या डाली जा सकती है, तब किया जाता है जैसे किसी शहर की जनसंख्या आदि

4<sup>प</sup> **Decimal** :- इस प्रकार के डाटा टाईप में सभी के प्रकार अंक (जिनमें दशमलव बिंदु होता है) आते हैं। जैसे 45.25, -458.23, 0.0023 आदि। इसमें साधारणतः जब हमें ऐसी संख्या डालना है, जिसमें दशमलव के बाद अंक आती है, तब

इस डाटा टाईप का प्रयोग करते हैं। जैसे ब्याज, तापमान, पगार आदि।

5<sup>th</sup> **string** :- यदि किसी वेरिएबल में कोई कैरेक्टर डालना है, तब इस टाईप का प्रयोग किया जाता है। जैसे नाम, पता आदि। साधारणतः जब कोई string वेरिएबल बनाते हैं, तब वह variable length string होता है। आप fixed length string वेरिएबल बना सकते हैं। उसके लिए string की अधिकतम सीमा देना पड़ता है। उदाहरण Dim name as string \* 50 इसमें name वेरिएबल में अधिकतम 50 कैरेक्टर ही डाले जा सकते हैं।

6<sup>th</sup> **Date** :- हम तारीख एवं समय को date और variant दोनों प्रकार के डाटा टाईप में रख सकते हैं। जब कोई डेसिमल संख्या date में परिवर्तित करते हैं, तब डेसिमल पाईट के बायें तरफ के अंक यह तारीख में परिवर्तित हो जाते हैं। और दाएँ तरफ के अंक समय में परिवर्तित हो जाते हैं।

उदाहरण

*Private A As Integer* "a" वेरिएबल का डाटा टाईप यह integer है, तथा वह एक माड्युल में सभी प्रोसीजर में चलता है।

*Dim Num As Double* "num" यह Double प्रकार का डाटा टाईप है एवं वह सिर्फ उसी प्रोसीजर में क्रियान्वित रहता है।

*Static Name As String* "name" वेरिएबल यह string प्रकार का है, तथा यह सिर्फ उसी प्रोसीजर में क्रियान्वित रहता है।

*Public Totdate As Date* "totdate" यह Date प्रकार का वेरिएबल है, तथा यह संपूर्ण एप्लिकेशन में क्रियान्वित रह

सकता है।

हमेशा उचित डाटा टाईप का ही प्रयोग करे। उदाहरण के लिए हम mark\_phy में integer और long interger, decimal इन तीनों प्रकार के डाटा टाईप प्रयोग कर सकते है। लेकिन हमें यह पता है, की किसी भी मार्क यह 100 से ज्यादा नहीं है, उस दशा में यदि हम long या decimal का प्रयोग नहीं करते है, क्योकी तब अनावश्यक रूप में मेमोरी consume हो जाएगी। कोई भी वेरिएबल का डाटा टाईप तय करने के पहले यह जान लेना आवश्यक होता है, कि उसमे कौन सा डाटा डाला जाना है।

नरेन्द्र पब्लिकेशन

## Arrays ऐरे

ऐरे क्रमशः एक से अधिक मानों को स्टोर करने के लिए प्रयोग किया जाता है। प्रत्येक मान को **element** कहा जाता है। ऐरे में प्रथम एवं अंतिम सीमा होती है, तथा सभी **element** इसी सीमा के अंदर होते हैं। ऐरे के प्रत्येक **element** का एक अलग **index number** होता है। ऐरे के किसी एक **element** का बदलाव का असर दूसरे **element** पर नहीं होता है। ऐरे के सभी मानों का डाटा टाईप एक समान होता है। लेकिन जब **variant** प्रकार का डाटा टाईप रहता है, तब **element** का डाटा टाईप अलग अलग हो सकता है। जब हम डाटा के एक बड़े समूह को एकत्र कर उन पर प्रोसेसिंग प्रयोग करना चाहते हैं, तब ऐरे का प्रयोग करते हैं। ऐरे को प्रोग्राम में प्रयोग करने से पहले व्यक्त किया जाता है। ऐरे को व्यक्त करने के लिए उस ऐरे का नाम एवं उसमें कितने **element** डालना यह दिया जाता है। **vb** में मुख्यतः तीन प्रकार के ऐरे होते हैं। **fixed size array** एवं **dynamic array** एवं **multidimensional array**।

### Array को व्यक्त करना

किसी **array** को व्यक्त करने के लिए निम्न सिन्टेक्स है।

```
dim <array name> (<value>) As <Data Type>
```

उदाहरण

```
dim salary(10) as integer
```

इस स्टेटमेंट से **salary** नाम का ऐरे व्यक्त होता है, जिसमें अधिकतम 11 **element** रह सकते हैं, जो 0 से 10 तक हो सकते हैं। तथा सभी **element** का डाटा टाईप यह **integer** है। इस प्रकार के ऐरे को **one dimensional array** कहा जाता है। जो संख्या "( )" के अंदर लिखी जाती है, उसे **upper bounds** कहा जाता है। इसमें **lower bounds** यह '0' है। यदि हमें दोनो

bound निश्चित करना है, तब निम्न प्रकार का स्टेटमेंट लिखना पडता है।

dim counter(5 to 20) as integer

counter नाम के ऐरे में lower bound यह 5 है, तथा upper bound यह 20 है।

### Dynamic array

बहुत बार हमें यह पता नहीं होता है, कि एक ऐरे में कितने element आने है, तब उस स्थिति में dynamic ऐरे का प्रयोग किया जाता है। dynamic ऐरे का आकार किसी भी समय बदला जा सकता है। इस प्रकार के ऐरे से हम मेमोरी का अच्छा उपयोग कर सकते है। जैसे किसी ऐरे में बहुत ज्यादा element का प्रयोग हो जाने के बाद, उस ऐरे को फिर से प्रयोग नहीं करना है, तब उसकी मेमोरी खाली कर सकते है। निम्न प्रकार से dynamic array बनाया जा सकता है।

dim ticket( ) as integer

इसके अतिरिक्त आप redim स्टेटमेंट का प्रयोग कर सकते है। लेकिन इस स्टेटमेंट का प्रयोग सिर्फ procedure में किया जा सकता है। उदाहरण

redim num (5)

जब जब आप redim स्टेटमेंट का प्रयोग करते है, तब तब उसके अंदर के पुराने मान खाली हो जाते है, तथा नये मान सेट होते जाते है। यदि redim स्टेटमेंट के ऐरे के मान को वैसे ही रखना है, तब preserve स्टेटमेंट का प्रयोग किया जाता है। उदाहरण

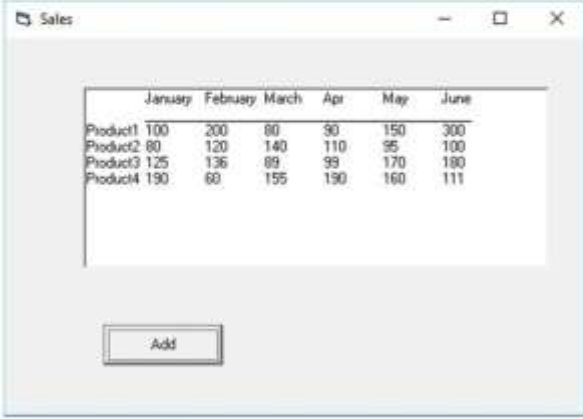
redim preserve num (30)

### Multidimensional array

इस प्रकार के ऐरे में एक से अधिक element समूह का प्रयोग किया जाता है। इनका व्यक्तीकरण निम्न प्रकार से किया जाता है

dim matrix (9,9) as double

उपरोक्त उदाहरण two dimensional ऐरे का है। इस प्रकार के ऐरे को टेबल कहते है।



	January	February	March	Apr	May	June
Product1	100	200	80	90	150	300
Product2	80	120	140	110	95	100
Product3	125	136	89	99	170	180
Product4	190	80	155	190	160	111

उदाहरण

*Private Sub*

*cmdAdd\_Click()*

*Dim prod, mth As Integer '*

*prod is product and mth is month*

*Dim saleVol(1 To 4, 1 To 6)*

*As Integer*

*Const j = 1*

```
listVolume.AddItem vbTab & "January" & vbTab &  
"February" & vbTab & "March"  
& vbTab & "Apr" & vbTab & "May" & vbTab & "June"  
listVolume.AddItem vbTab &  
" _____ "
```

```
For prod = 1 To 4
```

```
For mth = 1 To 6
```

```
saleVol(prod, mth) = InputBox("Enter the sale volume for"  
& " " & "product" & " " & prod & " " & "month" & " " &  
mth)
```

```
Next mth
```

```
Next prod
```

```
For i = 1 To 4
```

```
listVolume.AddItem "Product" & "" & i & vbTab &
saleVol(i, j) & vbTab & saleVol(i, j + 1) & vbTab &
saleVol(i, j + 2)
& vbTab & saleVol(i, j + 3) & vbTab & saleVol(i, j + 4) &
vbTab & saleVol(i, j + 5)
```

*Next i*

*End Sub*

*Output*

## Collection

ऐरे एक समान डाटा को स्टोर करने का सुविधाजनक तरीका है, लेकिन ऐरे के अलग अलग element को प्राप्त (Access) करना थोडा मुश्किल होता है। element को प्राप्त करने के लिए हमे उस element का index पता होना आवश्यक होता है। यदि आपको वह नही पता है तब, जब तक इच्छित element नही मिल जाता है, तब तक प्रत्येक element को जाँचना पडेगा है। इसके लिए वी. बी. डॉट नेट में collection नाम की सुविधा है। collection भी ऐरे के समान ही होता है, जो संबंधित डाटा को स्टोर करके रखता है। लेकिन इसमे प्रापर्टी और मेथड का प्रयोग होता है।

collection में आप एक key के द्वारा इच्छित element को ढुढ सकते है। उदाहरण के लिए student name यह studname() ऐरे की key है, तब आप किसी छात्र का संबंधित course तुरंत देख सकते है। यदी किसी student name इस collection की



किसी एंटी का index का पता है, तब भी आप तुरंत उसका course देख सकते हैं। लेकिन collection के किसी element को प्राप्त करने के लिए key यह ज्यादा अच्छा विकल्प है। collection बनाने के लिए निम्न सिन्टेक्स है।

```
dim <collection-name> as new collection
```

उदाहरण

```
dim studentn as new collection
```

Collection आब्जेक्ट की निम्न प्रापर्टी है।

add method	collection में नया घटक (element) डालने के लिए प्रयोग होता है।
remove method	collection से पुराना घटक निकालने के लिए प्रयोग होता है।
item method	किसी घटक को key या index देखने के लिए प्रयोग होता है।
count property	collection में कितने घटक है, इसकी गणना होती है।

### प्रोसीजर **Procedures**

किसी एप्लिकेशन में बहुत से प्रोग्राम, फॉर्म, माड्युल हो सकते हैं। जब कोई बड़ा प्रोजेक्ट या एप्लिकेशन बनाना हो, तब प्रथम उसकी संपूर्ण तरीके योजना बनाना चाहिए। संपूर्ण एप्लिकेशन को एक प्रोग्राम में नहीं लिखा जा सकता है, यदि उसे एक प्रोग्राम में लिखा हो तब वह बहुत ही जटिल जाता है। इसीलिए प्रोग्राम के कोड को छोटे तुकड़ों में बाँट कर लिखा जाता है। इन छोटे तुकड़ों को प्रोसीजर कहा जाता है। प्रोसीजर का प्रयोग प्रोजेक्ट में बार बार प्रयोग होने वाले कार्यों के लिए किया जाता है। प्रोसीजर का प्रयोग हम दूसरे प्रोजेक्ट में भी कर सकते हैं। प्रोसीजर के माध्यम से

बनाये गये प्रोजेक्ट में गलतीयाँ या त्रुटियाँ आसानी से खोजी जा सकती है। निम्न प्रकार की प्रोसीजर VB में प्रयोग होती है।

### Sub procedure

सबप्रोसीजर यह कोड का समूह होता है, जो किसी event पर प्रतिक्रिया करता है। कोड को तोड़ कर बनाये गये सबप्रोसीजर से हम वांछित कोड को ढुंढना या उसमें बदलाव करना आसान हो जाता है। सबप्रोसीजर को standard क्लास या फॉर्म मॉड्यूल में रखा जा सकता है। प्रत्येक बार जब प्रोसीजर को call किया जाता है, तो *sub* और *endsub* के अंदर लिखे हुए कोड क्रियान्वित होते हैं। इसका सिन्टैक्स निम्न प्रकार से है।

```
[Private/Public] [Static] sub procedure name [arguments]
```

```
body of program
```

```
end sub
```

arguments में अलग अलग argument यह “,” (कॉमा) से अलग अलग किया जाता है। प्रत्येक argument यह प्रोसीजर में एक वेरिएबल की तरह काम करता है। तथा उनके मान यह calling procedure में pass होते हैं।

सबप्रोसीजर दो प्रकार की होती है।

### General Procedure

किसी एक कार्य को करने के लिए बनाये गये कोड के समूह को जनरल प्रोसीजर कहा जाता है। जो स्टेटमेंट हमेशा प्रयोग होते हैं, उन्हें जनरल प्रोसीजर में रखा जाता है। जिससे हमें कोड को बार बार टाईप करने की आवश्यकता नहीं पड़ती है। उदाहरण

```
End Private Sub sum()
```

```
Dim a as integer, sum as integer
```

```
for a = 1 to 10
```

```
sum = sum + a
```

```
next a
```

*end sub*



*sum* प्रोसीजर को किसी भी इवेंट प्रोसीजर में **call** स्टेटमेंट का प्रयोग कर व्यक्त किया जा सकता है। प्रोसीजर को प्रयोग करने के लिए निम्न सिन्टेक्स है।

*call <procedure name>*

ऊपर दिए प्रोसीजर को **call** करने लिए निम्न कोड का प्रयोग किया जाता है।

*Private sub form\_add()*

*call sum*

*end sub*

जनरल प्रोसीजर को जोड़ने के लिए हमें निम्न बिंदुओं का ध्यान रखना आवश्यक है।

- a. जहाँ प्रोसीजर को जोड़ना है, उस मॉड्यूल की कोड विंडो खुली होनी चाहिए।
- b. जब हम कोड प्रोसीजर बनाते हैं, तब निम्न डायलॉग बॉक्स खुलता है। जिसमें निम्न विकल्प हैं
  - i. **name** टेक्स्ट बाक्स में प्रोसीजर का नाम टाईप करें।
  - ii. प्रोसीजर के प्रकार को नीचे दिए "type" रेडिओ बटन में से सिलेक्ट करें। जैसे **sub procedure** के लिए "Sub" बटन को क्लिक करें।
  - iii. जब कोई प्रोसीजर वर्तमान (current) मॉड्यूल के बाहर भी कार्यरत रखना है, तब **Public** विकल्प का चुनाव किया जाता है। और यदि वह प्रोसीजर सिर्फ उसी मॉड्यूल के अंतर्गत ही कार्यरत रहना चाहिए तब "Private" विकल्प का चुनाव किया जाता है।

इस डायलॉग बॉक्स में हम sub procedure, Function procedure एवं property procedure का निर्माण कर सकते हैं। sub प्रोसीजर मान को return नहीं करता है, अपितु function प्रोसीजर मान को return करता है।

सभी विकल्प सही तरीके डाल कर "OK" बटन को क्लिक करें, अब कोड विंडो खुल जाती है, तथा उसमें निम्न कोड दिखाई देता है।

```
Public Sub sum()
```

```
_____ body of program
```

```
End Sub
```

### Event procedure

जब प्रयोगकर्ता द्वारा कोई event चालू करना है, तब event procedure का प्रयोग किया जाता है। यह इवेन्ट, फॉर्म या कंट्रोल दोनों से संबंधित हो सकता है। जैसे किसी command button को क्लिक करना यह एक event है। इवेन्ट प्रोसीजर में जो भी वेरिएबल को व्यक्त किया जाता है, वह Public प्रकार के होने चाहिए। event procedure बनाने का सिन्टेक्स निम्न प्रकार से है।

```
private sub <control name/event name> <argument>
```

```
body of code
```

```
end sub
```

उदाहरण

```
private sub form_load()
```

```
msgbox = "this is load form event"
```

```
end sub
```

ऊपर दिए उदाहरण में प्रोसीजर फॉर्म लोड होने पर ही क्रियान्वित होता है।

## Function procedure

Sub प्रोसीजर के समान ही function प्रोसीजर एक पृथक प्रोसीजर होता है, जो किसी एक कार्य को निदेशित करता है। function एक से अधिक आर्गुमेंट लेकर सिर्फ एक मान return करता है। इसे expression के रूप में भी प्रयोग किया जा सकता है। लेकिन फंक्शन प्रोसीजर यह call किये प्रोसीजर के मान भी return करता है। फंक्शन प्रोसीजर में उसका नाम, आर्गुमेंट, एवं फंक्शन का कार्य लिखना पडता है। इस प्रकार की प्रोसीजर बनाने का सिन्टेक्स निम्न प्रकार से है।

```
[public] [private] [static] <function name> <arguments list>  
    body of code  
    name = expression  
end function
```

ऊपर दिए स्टेटमेंट का घटक के निम्न विवरण है।

**Public :-** यदि इस मान का प्रयोग करते है, तब बनाया हुआ फंक्शन यह सभी प्रोसीजर में मान्य रहता है।

**Private :-** जिस प्रोसीजर में इस फंक्शन का निर्माण किया है, सिर्फ उसी प्रोसीजर में कार्यरत रहता है।

**name :-** फंक्शन का नाम देने के लिए प्रयोग होता है।

**argument list :-** जब उस फंक्शन को call किया जाता है, तब जो आर्गुमेंट पास होना चाहिए उन वेरिएबल को डाला जाता है। यदि एक से ज्यादा वेरिएबल है, तब उन्हें “,” से अलग किया जाता है।

**body of code :-** उस फंक्शन को क्रियान्वित करने से जो कोड क्रियान्वित होना चाहिए उस कोड को लिखा जाता है।

**expression :-** फंक्शन की return value

यदि किसी फंक्शन के बनाने में public या private स्टेटमेंट नही दिया जाता है, तब VB उसे public प्रकार समझ लेता है।

सामान्यतः sub प्रोसीजर में स्वीकृत किये गये मान को प्रोसीजर में परिवर्तित किया जा सकता है, क्योंकि प्रोसीजर में वेरिएबल by reference पास किये जाते हैं। निम्न उदाहरण में किसी अंक को वर्गमूल निकालने का फंक्शन बनाया है।

```
function sqroot (num as double) as double
```

```
    if num < 0 then ' check the number
```

```
    exit function , exit to calling procedure
```

```
    else
```

```
        sqroot= sqr(num) ' retune the square root of the  
given number
```

```
    endif
```

```
end function
```

किसी फंक्शन में exit स्टेटमेंट का प्रयोग उस फंक्शन को खत्म कर बाहर आने के लिए होता है। किसी फंक्शन में एक से अधिक exit स्टेटमेंट हो सकते हैं। फंक्शन एवं प्रोसीजर का मुख्य अंतर यह है कि फंक्शन प्रोसीजर call किये प्रोसीजर को मान return करता है, जब की sub प्रोसीजर मान return नहीं करता है। VB द्वारा दिये हुए अधिकतम प्रोसीजर यह फंक्शन प्रोसीजर है।

### Passing argument by value

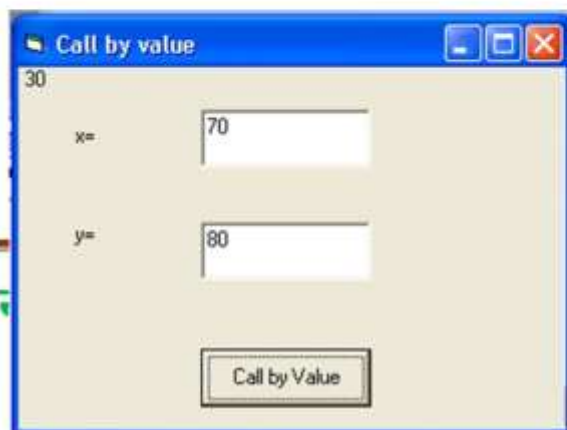
जब को आर्गुमेंट मान के आधार पर pass होता है, तब उस वेरिएबल की सिर्फ कॉपी ही पास होती है। यदि प्रोसीजर में उस वेरिएबल के मान के बदलाव सिर्फ उस कॉपी में ही होते हैं, तथा मूल वेरिएबल में कोई बदलाव नहीं होता है। इसके लिए "ByVal"

कथन का प्रयोग किया जाता है।

उदाहरण के लिए

```
sub procedure (byval  
netsalary as double)
```

```
statements
```



*end sub*

उदाहरण

*Function add(ByVal a As Integer, ByVal b As Integer)*

*Dim c As Integer*

*c = a + b*

*add = c*

*a = 70*

*b = 80*

*End Function*

*Private Sub Command1\_Click()*

*Dim x As Integer, y As Integer, z As Integer*

*y = 10*

*z = 20*

*x = add(y, z)*

*TextBox1.Text = y*

*TextBox2.Text = z*

*Print x*

*End Sub*

ऊपर दिय उदाहरण में a और b में add प्रोसीजर में दोनों के जोड़ की क्रिया हुई है, उस प्रोजिसर में a और b के मान अलग है (a=70, b=80) लेकिन वह मान calling प्रोग्राम के आर्गुमेंट x और y में स्थानांतरीत नहीं हुए है। सिर्फ add फंक्शन यह उत्तर(result) को calling प्रोग्राम में स्थानांतरीत करता है। इसलिए दोनो बाक्स में 10 और 20 यह मान दिखाई देते है।

### **Passing argument by reference**

इस प्रकार में प्रोसीजर के मूल वेरिएबल को प्राप्त किया जाता है। जिसके कारण वेरिएबल के मान में वास्तवीक बदलाव हो जाते है। यह VB का default पद्धति है। यदि आपने जो argument को

reference द्वारा pass का डाटा टाईप बताया है, तब आपको उस argument का मान भी pass करना अनिवार्य है। उदाहरण के लिए एक वेरिएबल जो प्रोसीजर में integer के रूप व्यक्त किया है, तथा उस वेरिएबल string के रूप में pass करना है। उसके लिए निम्न प्रकार से कोड लिखना होगा।

```
sub callbyref( )
    dim a as integer
    a =4
    byref(a)
end sub
sub byref (bar as string)
    msgbox bar 2 'the value in the bar is string type
end sub
```

उदाहरण

```
Function add(a As Integer, b As Integer)
Dim c As Integer
c = a + b
add = c
a = 70
b = 80
End Function
Private Sub Command1_Click()
Dim x As Integer, y As Integer, z As Integer
y = 10
z = 20
x = add(y, z)
TextBox1.Text = y
TextBox2.Text = z
Print x
```



### *End Sub*

ऊपर दिए उदाहरण में a और b के बीच में जोड़ की क्रिया की गई है। उस प्रोसीजर में a और b के मान बदल गये हैं। यह बदलाव calling program के y और z में स्थानांतरित हो गये हैं। जिससे पहले की x और z के मान जो क्रमशः 10 और 20 थे वह अब 70 और 80 हो गये हैं, जो TextBox1 और TextBox2 दिखाई देते हैं।

### **Optional argument**

किसी प्रोसीजर में option argument का प्रयोग करना है, तब argument सूची में किसी optional स्टेटमेंट प्रयोग से इसे बता सकते हैं। यदि आपने किसी एक argument को optional निश्चित किया है, तब उस सूची में उसके नीचे के सभी argument को optional निश्चित करना पड़ता है। optional argument प्रोग्राम के क्रियान्वन के समय प्रोसीजर को pass भी किया जाता है, या नहीं भी किया जाता है। आप optional argument में यह default मान भी डाल सकते हैं। उदाहरण

```
dim name as string
```

```
dim address as string
```

```
sub list (optional a as string, optional y as string)
```

```
    list1.additem a
```

```
    list2.additem b
```

```
end sub
```

```
private sub command1_click()
```

```
    name = "ABC"
```

```
    address = "XYZ"
```

```
    call listtext (name, address)
```

```
end sub
```

## Control statement

कंट्रोल स्टेटमेंट यह प्रोग्राम के क्रियान्वन (execution) की दिशा नियंत्रित करता है। किसी event के लिए जो प्रोग्राम वी.बी. डॉट नेट में लिखा जाता है, उसमें कमांड लाइन क्रम में होती है। वी.बी. डॉट नेट सबसे ऊपरी कमांड लाइन को प्रथम क्रियान्वित करता है, उसके बाद उसके नीचे, इस तरह अंत की कमांड लाइन क्रियान्वित होती है। बहुत सरल या छोटे प्रोग्राम में यह क्रमबद्धता सही काम करती है। लेकिन मध्यम या बड़े प्रोग्राम में बहुत सी तार्किक शर्तें होती हैं, जिनसे प्रोग्राम एक के बाद एक कमांड लाइन इस दिशा में नहीं चल सकता है, अपितु स्थिति के अनुसार वह कमांड लाइन क्रियान्वित करता है। किसी भी प्रोग्रामिंग भाषा की क्षमता उसके कंट्रोल स्टेटमेंट पर बहुतसी निर्भर होती है। विभिन्न तार्किक कंडीशन के लिए निम्न ऑपरेटर का प्रयोग होता है।

### Conditional operator

यह ऑपरेटर किसी विशिष्ट स्थिति में किस तरह से कार्य करना है, यह बताता है। कंडीशनल ऑपरेटर का प्रयोग गणितीय गणनाओं में नहीं किया जाता है, अपितु इसका प्रयोग दो मानों के बीच में तुलना करने में किया जाता है। इस ऑपरेटर के निष्कर्ष के अनुसार प्रोग्राम के अन्य भाग क्रियान्वित होते हैं। वी.बी. डॉट नेट में निम्न कंडीशनल ऑपरेटर हैं।

ऑपरेटर	कार्य	उदाहरण
=	बराबर	$2 + 3 = 4 + 1$
>	बड़ा	$6 > 3$
<	छोटा	$7 < 15$
>=	बड़ा या बराबर	$25 \geq 25$
<=	छोटा या बराबर	$4 \leq 4$

<>	बराबर नहीं	3 <> 2
----	------------	--------

इन ऑपरेटर का प्रयोग दो मानों, दो वेरिएबल, या समीकरणों की तुलना करने के लिए होता है। यदि दो मानों में से कोई एक मान null है, तब उसका उत्तर भी null होता है। इस तरह के समीकरणों के तीन संभव नतीजे होते हैं 1.सही, 2.गलत 3. null । string प्रकार के डाटा की तुलना करते समय वी.बी. डॉट नेट में कुछ नियम हैं।

- बड़े अल्फाबेट (capital case) यह छोटे अल्फाबेट (small case) से छोटे होते हैं। (a>A)
- अल्फाबेट क्रम की तुलना की जाती है। उदाहरण "b" यह "a" से कम माना जाता है।
- अंक यह कैरेक्टर से कम होते हैं। उदाहरण 1 यह one से कम होता है। (one >1)

वी.बी. डॉट नेट एक और विशेष प्रकार के कंडीशनल ऑपरेटर को सपोर्ट करता है।

Like :- इस ऑपरेटर में दो मानों की तुलना wildcard के आधार पर की जाती है। wildcard दो प्रकार के होते हैं "?" (सिर्फ एक कैरेक्टर) और "\*" (एक से अधिक कैरेक्टर)। इस ऑपरेटर में "#" का प्रयोग wildcard के रूप में संख्या के अंको को प्रदर्शित करने के लिए होता है।

उदाहरण

उदाहरण	हल
"abcdefgh" > "ABCDEFGH"	छोटे अल्फाबेट यह कैपीटल से बड़े माने जाते हैं।
"abcdefgh" > "abcde"	पहले मान की संख्या ज्यादा है।
"abcd" > "efgh"	क्रम में अंतर है। पहले क्रम के अक्षर बड़े माने जाते हैं।

"five" &gt; "5"

अंक छोटा होता है।

**Logical operator** तार्किक ऑपरेटर

यदि एक साथ एक से अधिक कंडीशनल ऑपरेटर का प्रयोग करना हो, तब तार्किक ऑपरेटर का प्रयोग किया जाता है। वी.बी. डॉट नेट में निम्न प्रकार के logical ऑपरेटर हैं।

ऑपरेटर	विवरण	उदाहरण	फल
AND	यदि इससे जुड़े हुए सभी कंडीशन की शर्तें true होती हैं, तब ही संपूर्ण स्टेटमेंट true माना जाता है।	(2 < 5) AND (5 = 2+3)	इन समीकरणों में दोनों कंडीशन सही हो रही हैं, तब उत्तर true में आता है।
OR ,	इससे जुड़े हुए कंडीशन में से कोई एक भी कंडीशन सही होती है, फिर भी परिणाम true आता है।	(3 < 5) OR (5=3)	इस स्टेटमेंट में एक समीकरण के मान true हैं इसलिए यह true दिखायेगा
XOR	इससे जुड़े हुए कंडीशन में से किसी एक की कंडीशन सही होती है, (लेकिन दूसरी कंडीशन false होना चाहिए) तब परिणाम true आता है।	(3 < 5) XOR (5=3)	इस स्टेटमेंट में एक समीकरण सही एवं दूसरा गलत है, तब इसका उत्तर true दर्शाता है।
NOT	यदि किसी समीकरण को गलत दर्शाना है, तब इस ऑपरेटर का	NOT( 5<7)	इसमें true दर्शाता है।

प्रयोग होता है।		
-----------------	--	--

Not ऑपरेटर का प्रयोग कम से कम हि किया जाना चाहिए। यदि कोई मान सही आ रहा है, और आपको उसे **False** करार देना है, तब इस ऑपरेटर का प्रयोग होता है। एक कंडीशन के अंदर बहुत ज्यादा conditional एवं logical ऑपरेटर का प्रयोग नहीं करना चाहिए। इसमें भ्रम पैदा होने की संभावना रहती है। यदि बहुत ज्यादा कंडीशन देने की आवश्यकता हो, तब स्टेटमेंट को तोड़ कर लिखा जाना चाहिए।

### Decision structure

वी.बी. डॉट नेट में दस प्रकार के कंट्रोल संरचनाये है, इनमें तीन प्रकार की selection structure. छह प्रकार की repetition structure एवं एक sequence structure है। प्रोग्राम में काम के अनुसार कंट्रोल संरचना का प्रयोग किया जाता है।

### Selection structure

वी.बी. डॉट नेट की प्रोसीजर किसी कंडीशन को जाँच कर उसके अनुसार कार्य निश्चित करती है। वी.बी. डॉट नेट में निम्न तीन प्रकार की selection structure उपलब्ध है।

1. *if .....then*
2. *if .....then .....else*
3. *select case*

1. *if .....then* :- यह स्टेटमेंट वेरिएबल के value या state के आधार पर कार्य करता है। वी.बी. डॉट नेट में इसे relational ऑपरेटर के साथ प्रयोग किया जाता है। इसे decision स्टेटमेंट भी कहा जाता है, क्योंकि यह समीकरणों के संबंधों को जाँच कर, निर्णय लेता है। इस स्टेटमेंट का निम्न फॉरमेट है

*if condition then*

प्रोग्राम के कोड (body of program)

*endif*

यह कंडीशन *if* से चालू होती है, तथा अंत में *endif* लगाना जरूरी है।

उदाहरण छात्रों के मार्क यदि 40 या उससे ऊपर है, तब उसे पास माना जायेगा

*if mark >= 40 then*

*result = "Pass"*

*endif*

इस उदाहरण वी.बी. डॉट नेट प्रथम mark वेरिएबल के मान को जांचता है, यदि वह 40 या उससे ज्यादा है, तब वह उसे "pass" दर्शाता है, अन्यथा कुछ भी नहीं दर्शाता है।

2. *if .....then.....else* : - वी.बी. डॉट नेट में यह सबसे अधिक प्रयोग होने वाली कंडीशन है। इसमें जब कंडीशन सही है, तब कौन सा कोड चलना चाहिए तथा कंडीशन गलत है, तब कौन सा कोड चलना चाहिए यह बताया जाता है। इस कंडीशन का निम्न फॉर्मेट है

*if condition then*

प्रोग्राम के कोड (body of program)

*else*

प्रोग्राम के कोड (body of program)

*endif*

यदि कंडीशन सही है, तब *if* के नीचे का प्रोग्राम कोड क्रियान्वित होता है, तथा यदि कंडीशन गलत है, तब *else* के नीचे का कोड क्रियान्वित होता है। *if...else* यह स्टेटमेंट *endif* से खत्म होता है। यदि हम student के mark के आधार पर result का उदाहरण लेते हैं।

*if mark >= 40 then*

```

    result = "pass"
else
    result = "fail"
endif

```

ऊपर दिए प्रोग्राम में यदि marks वेरिएबल का मान 40 या उससे अधिक है, तब result मान में "pass" दर्शाया जाएगा, और यदि marks मान 40 से कम है, तब result वेरिएबल में "fail" दर्शाया जाएगा। वी.बी. डॉट नेट में Iff फंक्शन का काम भी लगभग if—else—endif के समान है। लेकिन इस एक्शन में एक साथ तीनों ऑगुमेंट दिए जाते हैं।

iff (mark >= 40 "pass" "fail" )

इस ऑगुमेंट में पहला मान कंडीशन सही होने पर क्रियान्वित होता है, एवं दूसरा मान कंडीशन गलत होने पर आता है।

आप एक if ...else... endif के अंदर दूसरा if...else...endif स्टेटमेंट डाल सकते हैं। इसे nested स्टेटमेंट कहा जाता है। जब ऐसा कोड लिखा जाता है, तब बाहर के if स्टेटमेंट का endif यह अंत में आना चाहिए। उदाहरण

```

if salary >= 15000 then
    [ if intensive >= 4000 then
        [ deduction = 2500
        else intensive < 4000 then
            deduction = 1500
        ]
    ]
endif
endif

```

3. if .....elseif...else....endif :-यदि कोई वेरिएबल पर एक से ज्यादा कंडीशन का प्रयोग करना हो तब, if ...else...if स्टेटमेंट का प्रयोग किया जाता है। इसका फॉर्मेट

```

if condition then

```

```
    body of program
elseif condition then
    body of program
elseif condition then
    body of program
|
|
else
endif
```

उदाहरण

```
if marks >= 90 then
    grade = "A+"
elseif marks >=75 then
    grade = "A"
else if marks >=60
    grade = "B"
elseif marks >=45
    grade = "C"
elseif marks >=35
    grade = "D"
else
    grade= "fail"
endif
```

ऊपर दिए उदाहरण में marks के मानों के अनुसार grade का मान निश्चित होता है। यदि marks में ऊपर की पांचों कंडीशन पूरी नहीं होती है, तब वह else स्टेटमेंट लागू होता है, तथा grade का मान "fail" दर्शाता है।



## Loop Structure

यदि कोड को एक से ज्यादा बार क्रियान्वित करना हो, तब loop structure का प्रयोग किया जाता है। looping में कोड की पुनरावर्ती तब तक होती है, जब तक पूर्वनिर्धारित कंडीशन पूर्ण नहीं होती है। यदि कोई लूप में दि गई कंडीशन कभी असत्य नहीं होती तब उसे infinite loop कहा जाता है। loop का प्रयोग प्रोग्रामिंग में बहुत महत्वपूर्ण है। वी.बी. डॉट नेट में निम्न प्रकार के loop structure उपलब्ध है।

*do while .... loop*

*for .....next*

*for each .....next*

*while .....wend*

*do .....loop*

इस लूप का प्रयोग प्रोग्राम के कोड को एक से ज्यादा बार दोहराने के लिए होता है। इसमें एक कंडीशन दि जाती है, प्रोग्राम के कोड तब तक दोहराए जाते हैं, जब तक उसमें दि गई कंडीशन सत्य रहती है। इसके निम्न चार फॉरमेट हैं

<i>do</i> condition body of program <i>loop</i>	<i>while</i>	<i>do</i> body of program <i>loopwhile</i> condition	<i>of</i>	<i>do</i> condition body of program <i>loop</i>	<i>until</i>	<i>do</i> body of program <i>loopuntil</i> condition
---	--------------	---	-----------	--	--------------	---

इन चारों do लूप में एक कंडीशन दि जाती है। इन चारों स्टेटमेंट का नतीजे एक समान ही होते हैं। लेकिन प्रोग्रामिंग में परिस्थिती अनुसार इनका प्रयोग किया जाता है। इन के बीच का अंतर निम्न परिस्थितियों के तहत किया जाता है।

1. यदि परिस्थितिक जाँच do स्टेटमेंट के सबसे ऊपर की ओर है, और वह पहली बार ही असत्य पाई जाती है, तब लूप की

बॉडी एक बार भी क्रियान्वित नहीं होती है। यदि जाँच do स्टेटमेंट यदि नीचे दिया है, और कंडीशन असत्य है, फिर भी loop की बॉडी एक बार अवश्य क्रियान्वित होती है।

2. do लूप तब तक चलता है, जब तक while की कंडीशन सत्य होती है। और until लूप में जब तक कंडीशन का मिलान नहीं होता तब तक चलता है। यद्यपि इन दोनों के बीच के वास्तविक अंतर को नहीं देख सकते, क्योंकि कम्प्यूटर में दोहराने की क्रिया तीव्र गति से होती है।

उदाहरण

यदि हमें 1 से 10 तक अंको की सूची के प्रोग्राम बनाना है, तब चारों प्रकार के loop में निम्न कोड लिखना होगा

<i>dim A as integer</i> <i>A=0</i> <i>Do</i> <i>A&lt;=10</i> <i>A = A +1</i> <i>print A</i> <i>loop</i>	<i>dim A as integer</i> <i>A=1</i> <i>Do</i> <i>A = A +1</i> <i>print A</i> <i>loop A=10</i>	<i>dim A as integer</i> <i>A=0</i> <i>Do</i> <i>A&lt;=10</i> <i>A = A +1</i> <i>print A</i> <i>loop</i>	<i>dim A as integer</i> <i>A=1</i> <i>Do</i> <i>A = A +1</i> <i>print A</i> <i>loop until A=10</i>
---	---	---	---

*for.....next*

do... while/until लूप में कंडीशन के आधार पर कोड की पुनरावर्ती होती है। यदि आपको कोई कोड किसी निश्चित बार ही दोहराना है, तब *for....next* स्टेटमेंट का प्रयोग किया जाता है। इस लूप में कोड को कितने बार दोहराना है, यह देना पड़ता है, इसे counter कहा जाता है। इसका फॉर्मेट निम्न है।

*for counter= start to end [step increment]*

body of program

*next [counter]*

इसमें counter यह positive या negative रह सकता है। उदाहरण के लिए हम 1 से 10 के अंको की सूची का प्रोग्राम लेते हैं

```
dim A as integer
```

```
  for A = 1 to 10 step 1
```

```
    print A
```

```
  next
```

आप initial मान, अंतिम मान या स्टेप में वेरिएबल का भी प्रयोग कर सकते हैं। उदाहरण

```
Dim A as integer
```

```
dim Int, fin, stp as integer
```

```
init= 0
```

```
fin=10
```

```
stp = 1
```

```
for A= init to fin step stp
```

```
  print A
```

```
next
```

यदि लूप को असामान्य स्थितियों में बंद करना है, तब Exit स्टेटमेंट का प्रयोग किया जाता है।

```
for each .....next
```

यह कंट्रोल भी लगभग *for... next* के समान ही कार्य करता है। लेकिन *for each... next* स्टेटमेंट यह ऐसे के प्रत्येक element को दोहराने के लिए या किसी आब्जेक्ट को स्टेटमेंट के समूह को दोहराने के लिए किया जाता है। इस स्टेटमेंट का सिन्टेक्स निम्न प्रकार से है।

```
for each <element> in group
```

body of program

*next <element>*

इस स्टेटमेंट का प्रयोग करते समय निम्न बातों ध्यान रखना आवश्यक है।

1. ऐसे element का डाटा टाईप यह variant प्रकार का होना चाहिए।
2. इसमें सिर्फ variant वेरिएबल, generic आब्जेक्ट, या आब्जेक्ट ब्राउजर में सुचीबद्ध आब्जेक्ट की प्रयोग हो सकते हैं।
3. User define प्रकार के डाटा टाईप पर यह कंट्रोल प्रयोग नहीं हो सकता है।

*while ..... wend*

इस कंट्रोल में दि गई कंडीशन के अनुसार कोड की पुनरावर्ती होती है। इसका सिन्टेक्स निम्न प्रकार से है।

*while <condition>*

body of program

*wend*

यदि दि गई कंडीशन सत्य है, तब उसके अंदर के कोड क्रियान्वित होते हैं। फिरसे कमांड while स्टेटमेंट पर जाता है, तथा तब तक कोड दोहराये जाते हैं, जब तक वह कंडीशन असत्य नहीं होती है।

## Forms

फॉर्म के साथ कार्य करना

वी.बी. डॉट नेट में टूलबार के विभिन्न आब्जेक्ट को फॉर्म पर रखा जाता है। फॉर्म यह किसी भी एप्लिकेशन का मुख्य भाग होता है। इसी पर उपयोगकर्ता काम करता है। एक प्रोजेक्ट में एक या एक से अधिक फॉर्म हो सकते हैं। किसी भी एप्लिकेशन की user friendliness गुण उसके फॉर्म के डिजाइन पर बहुत मात्रा में आधारित होता है। इसकी निम्न प्रापर्टी है।

1. **Text** :- यह फॉर्म का टाइटल निश्चित करने के लिए प्रयोग होता है। जो टेक्स्ट इस प्रापर्टी में दिया जाता है, वह उस विंडो के ऊपर की ओर दर्शाया जाता है।
2. **Appearance** :- इस प्रापर्टी से फॉर्म की बाह्यआकृति निश्चित की जाती है। इसमें आप फॉर्म को तीनआयामी (3 dimensional) का रूप दे सकते हैं। इसके लिए इस प्रापर्टी में "1" मान दे अन्यथा "0" मान से फॉर्म सपाट आता है।
3. **border style** :- इस प्रापर्टी से फॉर्म की बार्डर का प्रकार निश्चित किया जा सकता है। इसमें निम्न प्रकार है。
  - i. **0-None** :- कोई बार्डर नहीं आती है। तथा run करते समय इसके आकार में परिवर्तन नहीं किया जा सकता है।
  - ii. **1-Fixed Single** :- फॉर्म के चारों ओर बार्डर आती है, लेकिन फॉर्म के आकार में परिवर्तन नहीं किया जा सकता है।
  - iii. **2-Sizeable** :- इस प्रापर्टी से फॉर्म के चारों ओर बार्डर आती है, तथा फॉर्म के आकार में बदलाव कर सकते हैं।
  - iv. **3-fixed dialog** :- फॉर्म पर डायलॉग बॉक्स प्रकार की बार्डर आती है। इस प्रकार के फॉर्म को आप सिर्फ Move या Close कर सकते हैं।

- v. **fixed tool window** :- इस प्रकार की बार्डर टूल बार के लिए किया जाता है।
4. **icon** :- फॉर्म के ऊपरी बायी ओर कोई आयकान दर्शाने के लिए इस प्रापर्टी का प्रयोग होता है। runtime में जब फॉर्म को minimize किया जाता है, तब फॉर्म के नाम के साथ आयकान दर्शाया जाता है।
5. **control box** : वी.बी. डॉट नेट में यदि फॉर्म को एक विंडो के तरह बनाना है, तब उसमें **control box** इस प्रापर्टी को मान "true" रखे। इस प्रापर्टी को लागू करने से फॉर्म के आयकन पर क्लिक करने से कंट्रोल मेन्यू दर्शाया जाता है। जिससे आप विंडो को बड़ा-छोटा या मूव कर सकते हैं। इसमें मेन्यू में "restore, move, size, minimize, maximize और close ) के विकल्प होते हैं। यदि कंट्रोल बाक्स नहीं चाहिए तब इस प्रापर्टी को "false" सेट करें।
6. **Font** :- इसमें फॉर्म पर आउटपुट से आने वाले टेक्स्ट के विभिन्न पैरामीटर सेट कर सकते हैं, जैसे फॉन्ट का आकार, रंग, स्टाइल आदि
7. **Mouse Pointer** :- इस प्रापर्टी से माउस पॉइंटर का प्रकार निश्चित कर सकते हैं। इस प्रापर्टी को क्लिक करने पर आप विभिन्न विकल्प दिखाई देते हैं। उसमें इच्छित सिलेक्ट करें। इसमें default मान यह विंडो के ऐसी माउस पॉइंटर प्रकार के लिए होता है।
8. **Picture** :- फॉर्म की पृष्ठभूमि (background) कोई पिक्चर या चित्र रखना है, तब इस प्रापर्टी का प्रयोग होता है। सिलेक्ट की गई पिक्चर पृष्ठभूमि पर आ जाती है।

9. **Window State** :- इस प्रापर्टी से जब फॉर्म चालू होता है, तब उसे कौन से आकार में रहना चाहिए (normal, minimize, maximize) यह तय किया जा सकता है।

फॉर्म प्रदर्शित करना

बनाये गये फॉर्म को प्रदर्शित करने के लिए show आरग्युमेंट का प्रयोग होता है। उदाहरण के लिए हमें Addmi नाम का फॉर्म प्रदर्शित करना है। तब निम्न स्टेटमेंट लिखेंगे

Addmi.Show

इसमें addmi यह फॉर्म का नाम है।

फॉर्म को मेमोरी में लोड करने के लिए load आरग्युमेंट का प्रयोग होता है, लेकिन इस आरग्युमेंट के बाद भी फॉर्म को प्रदर्शित करने के लिए Show आरग्युमेंट का प्रयोग करना पड़ता है। उदाहरण हमें fee नाम का फॉर्म का नाम लोड करना है, तब

Load Fee

इसमें fee यह फॉर्म का नाम है, तथा यह निर्धारित करता है कि फॉर्म यह modal यह modless है। मॉडल फॉर्म यह प्रयोगकर्ता से कमांड लेता है, जब तय यह फॉर्म चालू है, तब तब दूसरे फॉर्म या कमांड पर काम नहीं किया जा सकता है। इस प्रकार के फॉर्म को बंद करने के लिए या कार्य पूर्ण के सहमती देने के लिए एक कमांड बटन होना आवश्यक है। मॉडलेस फॉर्म यह सामान्य फॉर्म होते हैं। इस फॉर्म को चालू रखते हुए दूसरे फॉर्म पर कार्य किया जा सकता है।

फॉर्म को निम्नलिखित कारणों से लोड करना पड़ता है।

1. कुछ फॉर्म को show करने की आवश्यकता नहीं होती है, अपितु उन्हें सिर्फ load कर प्रदर्शित किया जा सकता है। इन फॉर्म में अन्य अनुप्रयोग के लिए आवश्यक प्रक्रियाएँ होती हैं या कुछ विशेष फंक्शन हो सकते हैं। उदाहरण के लिए डाटाबेस

कंट्रोल वाला फॉर्म जो दूसरे फॉर्म से जुड़ा हुआ है। इन्हे देखने के लिए यूजर इंटरफेस की आवश्यकता नहीं होती है।

2. जिन फॉर्म में अधिक ग्राफिक्स हो उस फॉर्म को लोड करने में कुछ समय लगता है, यदि फॉर्म पहले से लोड है, तब उसे show करने में कम समय लगता है।

किसी फॉर्म को Hide (अदृश्य) करना है, तब hide आरग्युमेंट का प्रयोग होता है। अदृश्य हुए फॉर्म unload नहीं होते हैं, उन्हें फिर से दर्शाने के सिर्फ show आरग्युमेंट की आवश्यकता होती है।

fee.hide

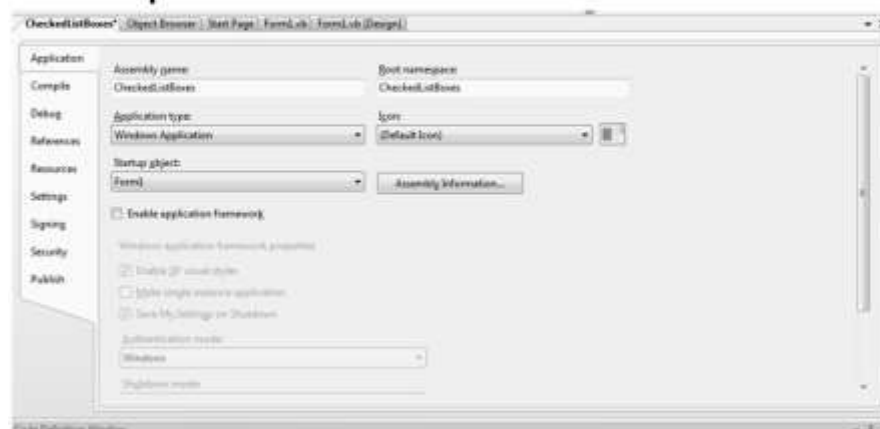
किसी फॉर्म को बंद करना है, तब "Unload" आरग्युमेंट का प्रयोग किया जाता है।

unload.addmi

हम किसी फॉर्म की प्रापर्टी, प्रापर्टी विंडो के अतिरिक्त सीधे कोड विंडो में परिवर्तित कर सकते हैं। उदाहरण के लिए हमें एक फॉर्म की caption प्रापर्टी बदलाना है, तब हम निम्न स्टेटमेंट टाईप कर सकते हैं

Form1.Caption="Fees Entry Form"

Startup form



कोई भी एप्लिकेशन में बहुतसे फॉर्म हो सकते हैं, लेकिन प्रोग्राम चालू होने के बाद जो सबसे पहले आना

चाहिए उसे startup form कहते। साधारणतः यह मुख्य फॉर्म होता



है, तथा उसके अंदर ही सभी चाइल्ड फॉर्म कार्यरत रहते हैं। इसलिए पॉरेन्ट फॉर्म को प्रोजेक्ट चालू होने के बाद सबसे पहले दर्शाया जाता है। फॉर्म को startup form बनाने के लिए

- Project मेन्यू के सबसे नीचे की ओर प्रोजेक्ट के नाम के साथ propertie विकल्प दिखाई देता है। उसमें <project name> property विकल्प को पर क्लिक करे आपको निम्न डायलॉग बॉक्स दिखाई देगा
- ऊपर की ओर application टैब को क्लिक करे।
- startup object लिस्ट बाक्स में मुख्य फॉर्म को सिलेक्ट करे।

### फॉर्म डिजाइन करना

फॉर्म डिजाइन करने के लिए वी.बी. डॉट नेट के टूलबॉक्स में विभिन्न टूल दिये हुए हैं। तथा उनकी स्थिति निश्चित करने के लिए फॉर्म पर एक ग्रीड दि होती है। हम उदाहरण एक सरल फॉर्म लेते हैं।



○ File मेन्यू पर क्लिक कर new project विकल्प को क्लिक करें

○ आपको new project का डायलॉग बॉक्स दिखाई देगा, उसमें बहुतसे icon

दिखाई देते हैं।

- उनमें से “Window Forms application” icon को क्लिक कर “OK” बटन को क्लिक करें

- अब आपको एक मुख्य पेज दिखाई देगी, उसमें फॉर्म के लिए चौकार बाक्स दिखाई देगा।



○ अब toolbox में से textbox को क्लिक करें तथा उसे फॉर्म पर ले जाकर एक टेक्स्ट बाक्स बनाये। तथा उस बाक्स के उपर के ओर ऐरो पर क्लिक करे, उसमें "multiline" का विकल्प दिखाई देता नहै। यदि जगज टेक्स्ट बाक्स में

एक से अधिक लाइन डालन है, तब इस विकल्प चालू करे। resize handle से उसका आकार निश्चित करें।

○ property window में "name" नाम की प्रापटी देखिए उसमें TextBox1 लिखा होगा, यह उस टेक्स्टबाक्स का नाम है, जो बादमें कोड

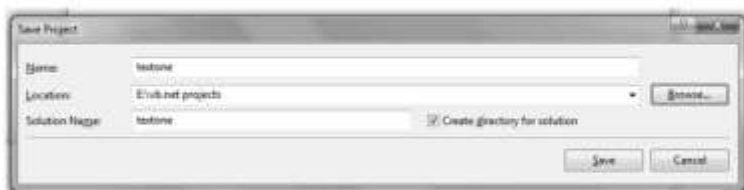
लिखते वक्त काम में आता है। वी.बी. डॉट नेट में प्रत्येक कंट्रोल को एक नाम दिया जाता है।

- Button कंट्रोल को सिलेक्ट करें तथा एक कमांड बटन अपने फॉर्म पर डालें
- Property window में उस कमांड बटन की name प्रापटी यह "Button1" रखें। तथा Text प्रापटी में "Ok" टाईप करे।
- "OK" बटन को double क्लिक करें आपको code विंडो दिखाई देगी उसमें निम्न कोड टाईप करे। वी.बी. डॉट नेट



कोड लिखने में आसान करने के लिए auto complete गुण होता है। जिससे यदि आप किसी प्रापर्टी का पहला शब्द टाईप करते हैं, तब उससे संबंधित प्रापर्टी दिखाई देती है, उस सूची में इच्छित आरग्युमेंट को सिलेक्ट करने के लिए "Tab" बटन को प्रेस करें। वी.बी. डॉट नेट स्वयं ही कोड लाइन का syntax जाँच लेता है। तथा स्वयं ही कमांड शब्द के पहले कॅरेक्टर को capital कर देता है। जो भी टेक्स्ट आप लिखते हैं, उसे " " के अंदर लिखा जाना चाहिए। ऊपर दिए उदाहरण में इस कोड का मतलब होता है, कि जब button1 नाम के कंट्रोल पर एक बार क्लिक किया जाये तब TextBox1 नाम के कंट्रोल का मान (value) यह "this is my first program" हो जाता है।

- इस प्रोग्राम का उत्तर देखने के लिए "tool bar " के Start विकल्प को क्लिक करें
- अब फॉर्म में जब आप "Ok" बटन को क्लिक करते हैं, तब टेक्स्टबाक्स में "this is my first program" यह संदेश आता है।
- इस प्रोजेक्ट को save करने के लिए "file" मेन्यू के "save project" विकल्प को क्लिक करें



फाइल का नाम टाईप करें जो एक फॉर्म फाइल के रूप में होती है, तथा इसका extension यह

“.frm” होता है। यदि आपके प्रोजेक्ट में एक से अधिक फॉर्म हैं, प्रत्येक फॉर्म को अलग नाम से सेव किया जाता है। प्रोजेक्ट को save करने के लिए

File मेन्यू के **save all** विकल्प को क्लिक करे। आपको प्राजेक्ट सेव करने को डायलाग बाक्स दिखाई देता है। उसमे इच्छित नाम टाईप करें। तथा **save** बटन को क्लिक करे।

इस प्रकार के फाइल का **extension** यह **“.vbproj”** होता है। उपरोक्त उदाहरण के फाइल का नाम **textone.vbproj** है।

नरेन्द्र पब्लिकेशन

## Object Oriented Programming

### Oops के विचार

Pascal, Basic, Fortran यह सभी क्रिया पर आधारित (Procedural) भाषायें हैं। प्रोसिजरल भाषा में बने प्रोग्राम यह निदेशों की सूची होते हैं। जब बड़ा प्रोग्राम बनाना होता है, तब एकल सूची के निदेशों पर नियंत्रण रखना मुश्किल हो जाता है। यदि प्रोग्राम को कुछ टुकड़ों में विभाजित नहीं किया गया हो, तब बहुत कम प्रोग्रामर सौ से अधिक स्टेटमेंट के प्रोग्रामों को समझने की क्षमता रखते हैं। इस कारण फंक्शन (Function) का प्रयोग किया जाता है, जिससे प्रोग्राम को आसानी से समझा जा सके। फंक्शन यह शब्द C++ या C भाषा में प्रयोग किया गया है, कुछ भाषाओं में इसे Subroutine, Subprogram या Procedure के नाम से जाना जाता है। प्रोग्राम को फंक्शन में विभाजित किया जाता है, तथा प्रत्येक फंक्शन का पूर्णतः परिभाषित उद्देश्य होता है। तथा एक फंक्शन को दूसरे फंक्शन के साथ जोड़ा जा सकता है। प्रोग्राम को फंक्शन में विभाजित कर तथा एक से अधिक फंक्शन को एकत्रित कर एक बड़ा ढाँचा जिसे मॉड्यूल (Module) कहते हैं वह बनाया जाता है। जैसे जैसे प्रोग्राम बड़ा तथा जटिल होते जाता है, वैसे वैसे पारंपारिक संरचनात्मक प्रोग्राम बनाने में मुश्किलें बढ़ती जाती हैं। प्रोसिजरलीयल भाषा में डाटा का दूसरा स्थान होता है। उदाहरण के लिये किसी इन्वेंटरी प्रोग्राम में डाटा को डिस्क फाईल से मेमोरी में लिया जाता है। जहाँ उसे एक व्यापक वेरिएबल (Global Variable) की तरह माना जाता है। इसका मतलब वह वेरिएबल फंक्शन के बाहर भी डाटा को स्थापित करता है। जिससे वह सभी फंक्शन के लिए उपलब्ध होता है। यदि नया प्रोग्रामर जो पुराने प्रोग्राम की सुक्ष्मता से ज्यादा परिचित नहीं है, वह इन्वेंटरी के

विश्लेषण के लिये जब कोई फंक्शन लिखता है, तब गलती से पुराने डाटा के खराब होने की संभावना हो जाती है। क्योंकि हर फंक्शन को डाटा प्राप्त करने का अधिकार होता है। जो डाटा किसी फंक्शन के द्वारा बदल दिया जाता है, आप उसे किसी भी तरह ठीक नहीं कर सकते। दूसरी समस्या जब डाटा का बहुत सारे फंक्शन प्रयोग करते हैं, तब डाटा को संग्रहित करना मुश्किल हो जाता है। इसके साथ पारंपारिक भाषा में नया डाटा का प्रकार बनाना बहुत ज्यादा मुश्किल कार्य होता है। इसमें कुछ पूर्वनिर्धारित डाटा के प्रकार होते हैं, जैसे Integer, Float, Charter आदि। नये प्रकार के डाटा बनाने के लिए भाषा की क्षमता को बढ़ाना पड़ता है, जो एक असामान्य बात होती है।

ऑब्जेक्ट ओरीएण्टेड भाषा को प्रोग्रामर द्वारा प्रोग्रामर के लिए विकसित की गई भाषा कह सकते हैं। ऑब्जेक्ट ओरीएण्टेड प्रणाली में हर चीज को आब्जेक्ट के रूप में लिया जाता है। हम इसे निम्न तरीके से परिभाषित कर सकते हैं।

आब्जेक्ट ओरीएण्टेड प्रणाली में आब्जेक्ट यह एक runtime entity है। आब्जेक्ट यह बैंक एकाउंट, एक जगह, डाटा को टेबल हो सकता है, या ऐसा घटक हो सकता है, जिसे प्रोग्राम में प्रयोग किया जा सकता है। इसमें प्रयोगकर्ता द्वारा बनाये घटक भी हो सकते हैं। ऑब्जेक्ट जिसकी स्वयं की एक पहचान होती है, जो उस ऑब्जेक्ट का सार होता है। जिसमें प्रत्येक ऑब्जेक्ट को स्वयं के साथ स्थापित करने की विशेषता तथा प्रभावित करने के क्षमता होती है।

हर ऑब्जेक्ट अपनी एक अलग पहचान रखता है। उस पहचान को ऑब्जेक्ट आइडी (Object Id) कहते हैं। कोई भी दो ऑब्जेक्ट की Oid एक समान नहीं होती है। जैसे ही हम प्रणाली में आब्जेक्ट बनाते हैं, वैसेही उसका Oid बन जाता है, तथा किसी भी स्थिति में

परिवर्तित नहीं होता है। यह Oid ऑब्जेक्ट के मिटाने पर ही खत्म होता है, और उस Oid को किसी और ऑब्जेक्ट के लिए प्रयोग नहीं किया जा सकता। यह Oid मेमोरी के पते के साथ संबंधित नहीं होता है।

ऑब्जेक्ट को उसके लक्षणों (Attribute) से वर्णित किया जाता है। जैसे किसी छात्र "विजय" का एट्रिब्यूट Stud\_Name हो सकता है। जिसे पारंपारिक डाटा प्रकार में डाटाबेस फिल्ड कह सकते हैं। ऑब्जेक्ट का Attribute यह एकल मान या बहु मान (Single Valued Or Multi Valued) हो सकता है।

ऑब्जेक्ट स्टेट यह मानों (Value) का समूह होता है, जो ऑब्जेक्ट के Attribute के समय उसे दिया जाता है। लेकिन ऑब्जेक्ट स्टेट यह बदल सकता है।

ऑब्जेक्ट पर क्रिन्यान्वित होने की प्रत्येक क्रिया की एक पद्धति होती है। यह पद्धतियाँ ऑब्जेक्ट के व्यवहार के बदलाव के लिये प्रयोग होती हैं। प्रत्येक पद्धति उसके नाम के साथ जानी जाती है, तथा वह उसका संपूर्ण अंग होता है। वह अंग किसी प्रोग्रामिंग भाषा में निर्देशन से बनाया जाता है। जैसे हमने Totalsal नाम की पद्धति बनाई है जिसमें सभी Salary का योग है। जो Main\_Salary तथा Sub\_Salary के ऑब्जेक्ट Attribute से बना है।

## Class

Class को भी हम कोड का समूह कह सकते हैं, जो एप्लिकेशन के बाकी आब्जेक्ट बनाने की readymade सुविधा प्रदान करता है। class module का कोड यह class द्वारा बनाये गये आब्जेक्ट का विवरण देता है। किसी भी आब्जेक्ट को बनाने के लिए तीन मुख्य बिंदु हैं

1. Encapsulation
2. Polymorphism

### 3. Inheritance

1. **Encapsulation** :- इसमें आब्जेक्ट के internal data एवं code के बारे में जानकारी होती है। data encapsulation का प्रयोग डाटा के execution के समय होता है। यह आब्जेक्ट की आंतरिक जटिलता को अन्य अप्लिकेशन से अलग रखता है। जब किसी डाटा या आब्जेक्ट का प्रयोग होता है, तब उसकी प्रापर्टि आदि को hide किया जाता है। उदाहरण के लिए जब हम किसी फार्म की caption प्रापर्टि को निश्चित करते हैं, तब हमें यह पता नहीं होता है, कि उसका string कैसे store किया है। यह एक प्रक्रिया है, जिसे डाटा एवं उसके फंक्शन को जोड़ कर उसे एक इकाई के रूप में बनाया जाता है, उस इकाई को class कहा जाता है। इस प्रकार के आब्जेक्ट को abstract data type कहा जाता है। इस प्रक्रिया से डाटा के खराब या corrupt होने की संभावना कम हो जाती है।

2. **polymorphism** :- विभिन्न आब्जेक्ट द्वारा एक समान नाम की method (उसके अंदर के घटक अलग रहते हैं) के प्रयोग करने की क्षमता को polymorphism कहते हैं। उदाहरण के लिए दो कमांड बटन में Print method का प्रयोग हो रहा है, यद्यपि मेथड का नाम एक समान है, लेकिन दोनों कमांड बटनों के अंदर कोड अलग अलग है। जैसे ही print method के कोड आब्जेक्ट में डाले जाते हैं, वैसे प्रत्येक आब्जेक्ट को यह पता है, कि जब उस method को call किया जाता है, तब क्या कार्य करना है।

**data abstraction** (डाटा का संक्षेपीकरण)

इसका मुख्य उद्देश्य डाटा की जटिलता को कम करना है। डाटाबेस डिजाइन का पहला चरण data abstraction होता है। एक बड़े



डाटाबेस प्रणाली में बहुत अधिक जटिल डाटा संरचना का प्रयोग होता है, इसलिए सर्वप्रथम उस डाटा संरचना को सरल करना आवश्यक है। जिससे संपूर्ण डाटा को संभलना आसान हो सके। **data abstraction** प्रक्रिया में डाटा के सिर्फ आवश्यक घटक ही रखे जाते हैं, तथा अन्य गुणधर्म या विशेषताओं को हटाया जाता है। तथा जब आवश्यक हो तब ही वांछित डाटा का प्रयोग किया जाता है, तथा अन्य समय उसे **hide** रखा जाता है। **data abstraction** का प्रयोग डिजाइन के समय किया जाता है, उदाहरण के लिए **bank** के डाटाबेस में किसी ग्राहक का डाटा जोड़ा गया है, तब उस ग्राहक की सिर्फ वित्तीय जानकारी ही बैंक डाटाबेस में डाली जाती है, ना की उस ग्राहक से संबंधित अन्य जानकारी है। उपरोक्त उदाहरण में हम ग्राहक का **Education qualification, experience** आदि की जानकारी नहीं डालते हैं।

### **class एवं object**

**vb.net** सभी घटक एक आब्जेक्ट के रूप में कार्य करते हैं, इस भाषा में ऑब्जेक्ट एवं क्लास एक दूसरे से जुड़े होते हैं। उदाहरण के लिए हमने "new" वेरिएबल घोषित किया है, जिसका डाटा टाईप **integer** है, अब इसमें **integer** डाटा टाईप यह क्लास है, तथा **new** वेरिएबल यह ऑब्जेक्ट है। **vb.net** में क्लास या आब्जेक्ट का निर्माण करना आसान है। क्लास के निर्माण के "Class" स्टेटमेंट का प्रयोग किया जाता है, तथा इसे खत्म करने के लिए "End Class" का प्रयोग किया जाता है। क्लास बनाने का सिन्टेक्स निम्न प्रकार से है

```
public class newdata
```

```
..  
..  
..
```

### *End Class*

क्लास की स्वयं की method, fields, properties सेट की जा सकती है, इन्हे क्लास के member कहा जाता है।

उदाहरण

```
public class newdata
    public value as integer
end class
```

### **Class एवं object member**

ऑब्जेक्ट ओरीएन्टेड प्रोग्रामिंग में class member या object member भी एक महत्वपूर्ण घटक होता है। एक क्लास पर जो मेंबर लागू होते हैं, तथा जो class name के साथ जुड़े होते हैं, उन्हें static या class member कहा जाता है। तथा जो मेंबर, क्लास द्वारा बनाये ऑब्जेक्ट से जुड़े होते हैं, उन्हें instance या object member कहा जाता है। उदाहरण

```
TextBox1.Text= "Narendra Publication"
```

TextBox1 यह एक ऑब्जेक्ट है, तथा उसके Text प्रापर्टी यह object member है

Over loading, overriding एवं shadowing

vb.net में overloading, overriding एवं shadowing यह तीन महत्वपूर्ण तकनीकें हैं, जिनके सहायता से हम एक ही नाम के एक से अधिक मेंबर बना सकते हैं।

ओवरलोडिंग Overloading

ओवरलोडिंग vb.net की वह तकनीक, जिसमें प्रत्येक मेंबर को प्रापर्टी के भिन्न version यह पद्धति प्रदान की जाती है। इसकी सहायता से एक ही मेथड के कई संस्करण बना सकते हैं। मेथड

के नाम तो एक ही होते हैं परन्तु उसके आर्ग्यूमेण्ट सूची अलग-अलग होते हैं।

प्रत्येक मेथड का एक अलग पहचान होती है उस पहचान का उपयोग कम्पाइलर स्वयं करता है। मेथड के अंदर के आर्ग्यूमेण्ट्स के बदलने से इसका पहचान भी बदल जाती है। आप मेथड के विभिन्न संस्करण रख सकते हैं तथा सभी के नाम एक ही होंगे। अर्थात् जब मेथड को ओवरलोड करते हैं तब मेथड की वास्तविक पहचान उसके अंदर के आर्ग्यूमेण्ट लिस्ट से होती है।

ओवरलोड मेथड बनाना (Creating An Overloaded Method)  
आप वी.बी. डॉट नेट में ओवरलोडेड मेथड निम्न तरीके से बना सकते हैं

पहला एक ही क्लास के अंदर दो या अधिक अलग अलग मेथड बनाएं जिसमें अलग-अलग आर्ग्यूमेण्ट हैं। उदाहरण के लिए

```
public sub Values(byval x as integer, byval y as integer)
```

```
Code goes here ....
```

```
End sub
```

```
public sub Values(byval x as long, byval y as long)
```

```
Code goes here ....
```

```
End subn
```

```
public sub values(byval x as single, byval y as single)
```

```
Code goes here ....
```

```
End sub
```

दूसरा तरीका है, कि Overloads स्टेटमेंट कर ओवरलोड मेथड बनाना, उदाहरण के लिए—

```
Public Overloads Function Get Centerinfo(By Val Stud_name  
As string)
```

```
.....
```

```
End Function
```

*Public Overloads Function Get Centerinfo(By Val fess As integer)*

.....

*End Function*

उपरोक्त कोड में Overloads की-वर्ड वैकल्पिक है। फंक्शन ओवरलोडिंग के निम्न नियम है, जिनके अनुसार ही ओवरलोड में मेथड बनाई जाती है।

ओवरलोडिंग के नियम (Rules For Overloading)

आर्ग्यूमेंट सूची में अंतर (Arguments List Must be Different) :- प्रत्येक ओवरलोडेड मेथड में आर्ग्यूमेंट सूची अलग अलग होनी चाहिए। एक नाम के मेथड में एक समान आर्ग्यूमेंट नहीं रख सकता है। तथा ऐसे आर्ग्यूमेंट भी नहीं रखे जा सकते हैं, जिनका नाम अलग हो लेकिन डाटा टाईप एक समान हो। यदि ऐसा रखा जाता है, तब वीबी डॉट नेट निम्न संदेश दर्शाता है।

*Method [methodnam] has multiple definition with identical signatures*

निम्न उदाहरण को वीबी डॉट मान्य नहीं करता है।

*Public Overloads Function Getstudinfo(By Val stud\_name As String)*

.....

*End Function*

*Public Overloads Function Get studinfo(By Val stud\_address as String)*

..

*End Function*

उपरोक्त उदाहरण में दोनो आर्ग्यूमेंट के डाटा टाईप एक समान है। किन्तु निम्नलिखित कोड स्वीकार्य है क्योंकि आर्ग्यूमेंट सूची अलग है—

```
Public Overloads Function Getstudinfo(By Val  
stud_name As String)
```

```
EndFunction
```

```
Public Overloads Function Get studinfo(By stud_address  
as String )
```

```
End Function
```

आप ओवरलोडेड मेथड में सिर्फ Optional आर्ग्यूमेंट में बदलाव कर अलग मेथड का निर्माण नहीं किया जा सकता। निम्न को को वीबी डॉट नेट अलग मेथड के रूप में स्विकृती नहीं देता है।

```
Public Overloads Function Getstudinfo(By Val  
stud_name As String)
```

```
End Function
```

```
Public Overloads Function Get studinfo(By val  
stud_address as String, optional By Val lastname as  
string)
```

```
End Function
```

जो ओवरलोड मेथड एक ही नाम को साझा कर रहे हैं, उनके सभी sub एवं procedure के एक नाम के अलग अलग डाटा टाईप को अनुमति नहीं देता है। निम्न उदाहरण को वीबी डॉट नेट स्विकृती नहीं देता है।

```
public overloads sub add(x as integer, y as ineger)
```

```
public overloads function add(x as long, y as long)
```

आप ओवरलोडेड मेथड के मध्य केवल रिटर्न टाईप्स से अंतर नहीं कर सकते हैं।। जैसे इसे उदाहरण को देखें—

```
Public Overloads Function Getstudinfo(By Val  
Stud_Name As String) As String
```

```
End Function
```

*Public Overloads Function Getstudinfor(By Val  
stud\_name As String) As Data Dataset  
End Function*

## **Inheritance**

vb.net में एक class की प्रॉपर्टि को दूसरे या उसके अंतर्गत क्लास में प्रयोग करने के कार्य को inheritance कहा जाता है। vb.net में पहले से बहुत बड़ा Class का संकलन है, उस क्लास के आधार पर आप नये क्लास बना सकते है। यह एक बहुत महत्पपूर्ण सुविधा हैं

नया क्लास बनाने के लिए "Inherits" कीवर्ड का प्रयोग किया जाता है। जो क्लास इस कीवर्ड से बनाया जाता है, उसे derived class कहा जाता है। तथा जिस क्लास से नई क्लास बनाई जाती है, उसे Base class कहा जाता हैं। derived क्लास में base क्लास की प्रॉपर्टि, मेथड, इवेंट, फील्ड आदि का प्रयोग किया जा सकता है। उदाहरण के लिए आपने computer नाम के क्लास में mouse नाम का event बनाया है, उसे आप उसके अंदर के desktop, laptop नाम के क्लास में प्रयोग कर सकते है। वैसे कोई भी क्लास यह base class के रूप में प्रयोग हो सकती है। उपरोक्त उदाहरण के अनुसार हम computer नाम का class बनाते है, तथा उसमे mouse नाम की मेथड बनाते है

```
Public class computer  
    Public MainForm as Form1  
    Public Sub New( ByVal Form1 As Form1)  
        MainForm= Form1  
    End sub  
    Public sub mouse()
```

```
        MainForm.TextBox1.Text= "Mouse is the part  
of computer "  
    end sub  
End Class
```

नया क्लास Desktop बनाने के लिए निम्न कोड है,

```
Public Class Desktop  
    Inherits Computer  
|  
End Class
```

उदाहरण

```
Module Module1  
Public Class s1  
    Public a As Integer = 5  
    Public Function val() As Integer  
        Return a  
    End Function  
End Class  
Public Class s2  
    Inherits s1  
    Public c As Integer = 20  
    Public Function add() As Integer  
        Return c + a  
    End Function  
End Class  
Sub Main()  
    Dim res As New s2  
    System.Console.WriteLine("Final Value is::")
```

```
System.Console.WriteLine(res.add())  
Console.Read()  
End Sub  
End Module
```

## एक्सेस स्पेसिफायर (Access Spcifiers)

एक्सेस स्पेसिफायर किसी ऑब्जेक्ट तथा इसके घटकों को प्राप्त करने के स्कोप का वर्णन करता है। या हम दूसरे शब्द में यह कह सकते हैं, की एक्सेस स्पेसिफायर का उपयोग कर क्लास के घटकों के स्कोप पर नियंत्रित कर सकते हैं। हम एक्सेस स्पेसिफायर का उपयोग एप्लिकेशन के डाटा के सुरक्षा के लिए होता है।

विजुअल बेसिक पाँच एक्सेस स्पेसिफायर यथा Public, Private, Protected, Friend तथा Protected Friend उपलब्ध कराता है।

### पब्लिक (Public)

यह डॉट नेट का सबसे सामान्य एवं default एक्सेस स्पेसिफायर है। इसे प्राप्त करने में कोई बंधन नहीं होता है, अर्थात् संपूर्ण प्रोजेक्ट में इसे कहीं से भी एक्सेस किया जा सकता है। यदि स्टेटमेंट में इस कीवर्ड का प्रयोग किया है, तब अर्थ होता है, कि ये घटक प्रोजेक्ट के अंदर कहीं से भी या अन्य प्रोजेक्ट से जो उन मूल प्रोजेक्ट से जुड़े हैं, उनमें बने असेम्बली में कहीं से एक्सेसिबल होता है। आप Public का उपयोग मॉड्यूल, इंटरफेस या नेमस्पेस में कर सकते हैं। इसे इंटरफेस, मॉड्यूल, क्लास या स्ट्रक्चर के भीतर घोषित किया जा सकता है परन्तु इसे किसी प्रॉसीजर के अंदर घोषित नहीं किया जा सकता है।

उदाहरण के लिए –

```
Public Class class stude
```

पब्लिक डिक्लेअरेशन को दर्शाता है।



## प्रोटेक्टेड (Protected)

Protected की-वर्ड डिक्लेअरेशन स्टेटमेंट में यह स्पष्ट करता है कि अवयव केवल उसी क्लास के अंदर या उस क्लास के क्लास से ही प्राप्त किया जा सकता है। आप Protected का उपयोग सिर्फ उसी क्लास में तथा उस क्लास के मेम्बर को डिक्लेअर करते समय कर सकते हैं। इसका अर्थ है कि आप क्लास में ही प्रोटेक्टेड अवयव को डिक्लेअर कर सकते हैं किन्तु इसे सोर्स फाइल या नेमस्पेस स्तर पर, इंटरफेस, मॉड्यूल, स्ट्रक्चर या प्रोजेक्ट के अंदर घोषित नहीं कर सकते हैं। ।

*Protected Class Institute*

## फ्रेंड (Friend)

Friend की-वर्ड का उपयोग डिक्लेअरेशन स्टेटमेंट में यह दर्शाता है, कि वह घटक एक समान घटकों के असेम्बली के अंदर ही प्राप्त किया जा सकता है। इन्हें असेम्बली के बाहर से प्राप्त नहीं किया जा सकता है। Friend विकल्प का प्रयोग मॉड्यूल, इंटरफेस या क्लास, नेमस्पेस स्तर पर कर सकते हैं। इसे किसी प्रोजेक्ट के अंदर घोषित नहीं किया जा सकता।

*Friend StringForThisProject As String*

Friend डिक्लेअरेशन को दिखाता है।

## प्रोटेक्टेड फ्रेंड (Protected Friend)

किसी डिक्लेअरेशन स्टेटमेंट में दोनों Protected तथा Friend का एक साथ प्रयोग कर सकते हैं। यदि दोनों की-वर्ड का प्रयोग किया है, तब यह अर्थ होता है, कि यह घटक या तो मूल क्लास से या फिर समान असेम्बली से प्राप्त हो सकते हैं। Protected Friend का उपयोग क्लास स्तर पर कर सकते हैं या उस क्लास के सदस्य को डिक्लेअर करते समय कर सकते हैं। लेकिन आप

इसे सोर्स फाइल या नेमस्पेस स्तर पर, इंटरफेस, मॉड्यूल, स्ट्रक्चर या प्रोसीजर के अंदर डिक्लेअर नहीं किया जा सकता है। उदाहरण

*Protected Friend StringFor Bank\_emp As String*

### प्राइवेट (Private)

स्टेटमेंट का Private की-वर्ड दर्शाता है, कि वह घटक सिर्फ मॉड्यूल, क्लास या स्ट्रक्चर से प्राप्त कर सकते हैं। इसे सोर्स फाइल या नेमस्पेस स्तर पर, इंटरफेस के अंदर या प्रोसीजर में घोषित नहीं किया जा सकता है। किन्तु Private की-वर्ड के प्रयोग से कोड को compile करना आसान हो जाता है।

यह कोड Private डिक्लेअरेशन को दिखाता है—

*Private number acc\_num As Integer*

### एब्सट्रेक्ट प्रॉपर्टीज (Abstract Properties)

क्लास के अंदर प्रॉपर्टी को Abstract की-वर्ड की सहायता से एब्सट्रेक्ट घोषित किया जा सकता है। याद रखें कि क्लास में एब्सट्रेक्ट प्रॉपर्टी में कोई कोड नहीं होता है। get/set एक्सेसर अर्द्धकॉलन (;) के साथ व्यक्त किया जाता है। व्युत्पन्न क्लास में Set तथा get एक्सेसर को लागू करना आवश्यक है।

यदि एब्सट्रेक्ट क्लास में केवल सेट एक्सेसर होता है तो हम व्युत्पन्न क्लास में केवल Set को लागू कर सकते हैं। निम्नलिखित प्रोग्राम एब्सट्रेक्ट प्रॉपर्टी को दर्शाता है—

वी.बी. डॉट नेट के अंदर प्रॉपर्टी भाषा स्तर पर जोड़े गये महत्वपूर्ण फीचर है। ये जी.यू.आई प्रोग्रामिंग में बहुत उपयोगी है। याद रखें कि कम्पाइलर वस्तुतः उपयुक्त get तथा set मेथड को बनाता है जब वी.बी. डॉट नेट प्रॉपर्टी प्रारूप को पार्स (parse) कराता है।

### एब्सट्रेक्ट क्लास (Abstract Class)

एब्सट्रैक्ट क्लास के आधार पर ऑब्जेक्ट का निर्माण नहीं होता है। अपितु इस क्लास को आधार (base) क्लास की तरह ही डिजाइन किया जाता है जिसे अन्य क्लासों द्वारा इनहेरिट किया जा सकता है। अन्य क्लासों को बनाने के लिए यह एक आधार देता है। एब्सट्रैक्ट क्लास को डिक्लेअर करने के बाद इसके स्वयं का ऑब्जेक्ट नहीं बनाया जाता बल्कि इसे उसके अंदर इनहेरिट किया जाता है। इंटरफेस की तरह एब्सट्रैक्ट क्लास मेम्बर निर्धारित कर सकता है जिसे इंहेरिट करने वाले क्लास लागू करते हैं। इंटरफेस के विपरीत एक ही क्लास एब्सट्रैक्ट क्लास को इनहेरिट कर सकता है। एब्सट्रैक्ट क्लास केवल उन्हीं सदस्यों को स्पष्ट कर सकता है जिन्हें इनहेरिट कर रहे क्लास लागू करें।

## Class module

Class module बनाने के लिए निम्न चार पदों का प्रयोग होता है।

1. Class module को जोड़ना एवं उसकी प्रापर्टी सेट करना।
2. class को बनाना एवं उसकी प्रापर्टी सेट करना
3. class की method बनाना
4. class के लिए event निश्चित करना

Class module यह वी.बी. डॉट नेट में object oriented programming का मूल तथ्य है। एक class module के अंदर उसके class की प्रापर्टी एवं मेथड परिभाषित की जाती है। प्रत्येक class की एक अलग फाइल होती है, जिसका extension “.CLS” होता है। एक class module और standard code module लगभग समान होते हैं क्योंकि दोनों को उस एप्लिकेशन के अंदर के दूसरे माड्युल, उनके फंक्शन का प्रयोग कर सकते हैं। standard module फाइल का extension “.bas” होता है। यह दो माड्युल निम्न तरीके से भिन्न हैं

डाटा स्टोर करने के तरीके दोनो प्रकार के माड्युल में अलग अलग होते है।

Standard माड्युल में डाटा की एक से अधिक कॉपी नही होती है, अर्थात जब किसी प्रोग्राम द्वारा किसी माड्युल के public वेरिएबल का मान बदलता है, तब उस प्रोग्राम का दूसरा हिस्सा उसी वेरिएबल को प्रयोग करता है, तब उसे को भी परिवर्तित मान मिलता है। लेकिन class module में प्रत्येक class के आब्जेक्ट के लिए अलग डाटा रहता है।

standard माड्युल में जब तक प्रोग्राम चालू है, तब तक डाटा रहता है। यद्यपि प्रत्येक आब्जेक्ट का डाटा उस आब्जेक्ट के lifetime तक अस्तित्व में रहता है।

Standard module के public वेरिएबल उस एप्लिकेशन में कही भी प्रयोग किये जा सकते है।लेकीन class module में public वेरिएबल सिर्फ वो ही आब्जेक्ट प्राप्त कर सकते है, जिन्हे उस माड्युल से जोडा है।

### Class module बनाना

Class module बनाने के लिए "Project" मेन्यू के "Add class module" विकल्प को क्लिक करें। एक डायलॉग बॉक्स दिखाई देगा उसमें से Class module विकल्प को क्लिक कर open बटन पर क्लिक करे। इसके बाद Class नाम की विंडो खुल जाती है।

प्रापर्टी का प्रयोग डाटा को, प्रोग्राम और class में अदान प्रदान करने के लिए होता है, class module की प्रापर्टी दो प्रकार से बनाई जा सकती है।

creating public variable

creating property procedure

वेरिएबल को "public" कीवर्ड से बनाया जाता है। उदाहरण के लिए हम result माड्युल में दो वेरिएबल बनाते है।

public grade as string  
public totalmarks as integer

Class module में प्रत्येक public वेरिएबल को उस class के आब्जेक्ट की class property माना जाता है।



Property procedure को ज्यादा वरियता दी जाती है, क्योंकि इसमें “data hiding” का गुण है। Data hiding उसे कहा जाता है, जब किसी class का बाकी डाटा प्रयोग हो रहा हो, तब आब्जेक्ट के डाटा एक हिस्सा संरक्षित रहता है। यह emcapsulation का एक जरूरी अंग है।

वी.बी. डॉट नेट में तीन प्रकार से property procedure बनाया जा सकता है।

1. Property let
2. Property set
3. Property get

हम property procedure को कोड विंडो में टाईप कर सकते हैं, या “Tools” मेन्यू के “Add procedure” विकल्प से लागू कर सकते हैं। इस विकल्प पर क्लिक करने से निम्न डायलॉग बॉक्स दिखाई देता है। डल

Name टेक्स्ट बाक्स में procedure का नाम टाईप करें। type विकल्प में किस प्रकार की प्रोसीजर बनाना है, उसे सेट करें। तथा scope में इच्छित विकल्प सिलेक्ट करें।

Ok बटन को क्लिक करें।

अब आपको और एक विंडो दिखाई देगी, जिसमें property let और property get दोनों प्रोसीजर दिखाई देती हैं। साधारणतः कोई भी प्रापर्टी यह किसी दो property procedure से मिल कर बनती है।

property procedure	विवरण
property let	जब प्रोसीजर को आब्जेक्ट द्वारा value pass किया जाता है, तब प्रयोग होती है।
property set	जो value, प्रापर्टी को pass करना है, उसे बताया जाता है,
property get	Class को value pass करने के लिए प्रयोग किया जाता है।

एक उदाहरण लेते हैं, जिसमें एक कमांड बटन और टेक्स्ट बाक्स कंट्रोल है। जब प्रयोगकर्ता उस कमांड बटन को क्लिक करता है, तब टेक्स्ट बाक्स के अंदर का टेक्स्ट capital हो जाता है।

class module के लिए कोड

```
Dim getcap As String
```

```
Public Property Get captext() As String
```

```
    captext = getcap
```

```
End Property
```

```
Public Property Let captext(ByVal newcap As String)
```

```
    getcap = UCase(newcap)
```

```
End Property
```

फॉर्म विंडो के कोड

```
Private Sub Command1_Click()
```

```
    Dim cs As Class1
```

```
    Set cs = New Class1
```

```
    cs.captext = TextBox1.Text
```

```
    TextBox1.Text = cs.captext
```

```
End Sub
```

ऊपर के उदाहरण में

वेरिएबल `getcap` यह टेक्स्ट बाक्स के अंदर के टेक्स्ट को hold करता है।

property `let` प्रोसीजर का काम टेक्स्ट बाक्स के अंदर के टेक्स्ट पर `Ucase` लागू करना एवं उस `value` को `getcap` वेरिएबल को `assign` करना है।

नरेन्द्र पब्लिकेशन

## Activex Contols

वी.बी. डॉट नेट के कंट्रोल

वी.बी. डॉट नेट में बनाये गये अनुप्रयोग (application) में परिष्कृत ग्राफिक एवं विभिन्न ऑब्जेक्ट/कंट्रोल सम्मिलित करने की सुविधा है। आप किसी टेक्स्ट को ग्राफिक के रूप में संयोजित कर सकते हैं, तथा टेक्स्ट का आकार, रंग, प्रकार आदि बदल सकते हैं। आप फॉर्म में विभिन्न कंट्रोल जैसे टेक्स्टबाक्स, कमांड बटन, लिस्ट बाक्स आदि का प्रयोग कर सकते हैं। आप इन्हे फॉर्म पर आसानी से संयोजित कर सकते हैं, इनका आकार, जगह आसानी से निश्चित कर सकते हैं। वी.बी. डॉट नेट में ऑब्जेक्ट यह पहले से परिभाषित एवं बार बार प्रयोग किये जाने वाले घटक है। इन ऑब्जेक्ट की सहायता से फॉर्म को आसानी से बनाया जा सकता है, तथा फॉर्म को बहुत user friendly किया जा सकता है। ऑब्जेक्ट यह प्रयोगकर्ता के लिए एक interface का कार्य करता है। कोई भी कंट्रोल में तीन मुख्य घटक होते हैं।

1. **Property** : आब्जेक्ट कैसे प्रदर्शित होना है, कहाँ पर दर्शाना है, आदि इस घटक में तय होता है।
2. **Method** :- आब्जेक्ट को कब कार्य करना है, वह इस घटक में तय होता है।
3. **event** :- जब ऑब्जेक्ट पर मेथड लागू होती है, तब क्या होना चाहिए यह event में तय होता है।

वी.बी डॉट नेट के कंट्रोल को आप फार्म पर एक जगह से दूसरे जगह ले जा सकते हैं, उनका आकार बदल सकते हैं। अलग अलग कार्य के लिए विभिन्न प्रॉपर्टि का प्रयोग होता है। उदाहरण के लिए किसी कंट्रोल का रंग बदलने के लिए "fore color" प्रॉपर्टि का प्रयोग होता है। आप किसी कंट्रोल की प्रॉपर्टि "properties



“window” से सेट कर सकते हैं। या कोड विंडों में प्रॉपर्टि सेट कर सकते हैं।

वि बी डॉट नेट में प्रत्येक कंट्रोल एक ऑब्जेक्ट होता है, संपूर्ण फार्म भी ऑब्जेक्ट होता है।

### Text box control

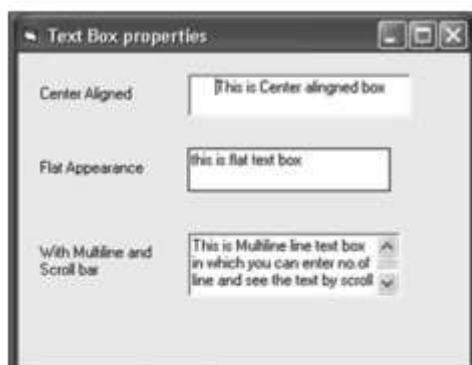


यह सबसे अधिक प्रयोग में होने वाला कंट्रोल है। इस कंट्रोल में प्रयोगकर्ता इच्छित टेक्स्ट इस बाक्स में टाईप कर सकता है। इस कंट्रोल से एक छोटा बाक्स बनता है, जिसमें ना सिर्फ टेक्स्ट को टाईप किया जा सकता है, अपितु उसे कॉपी, डिलीट, पेस्ट आदि करने की भी सुविधा होती है। यदि बड़ा टेक्स्ट डालना है, तब उसे scroll विकल्प भी दिया जा सकता है। इस टेक्स्ट बाक्स में डाला गया टेक्स्ट यह साधारण टेक्स्ट, पासवर्ड, नंबर कुछ भी हो सकता है। यह एक लाइल या एक से अधिक लाइन का भी हो सकता है। लेकिन इसमें ग्राफिक नहीं डाल सकते हैं। वि बी डॉट नेट में इसे `TextBox<n>` के नाम से सेट किया जाता है। इसका उपयोग उपयोगकर्ता से इनपुट प्राप्त करने के लिए, आउटपुट दर्शाने आदि कार्य के लिए होता है। यदि कोई default टेक्स्ट दर्शाना है, तब उसे “Text” प्रॉपर्टि में सेट किया जाता है। यदि खाली बाक्स दर्शाना है, तब “Text” प्रॉपर्टि में कुछ ना लिखें।

इस कंट्रोल निम्न प्रापर्टी है।

- i. **Font** :- इस प्रॉपर्टि से टेक्स्ट बाक्स के में टेक्स्ट टाईप होगा उसका फॉन्ट, फॉन्ट टाईप तथा टेक्स्ट का आकार सेट किया जा सकता है।
- ii. **Max Length** :- अधिकतम कितना टेक्स्ट टाईप हो, वह इस प्रॉपर्टि में सेट किया जा सकता है। यदि उससे अधिक टेक्स्ट टाईप

iii. **Locked:-** इस प्रापर्टी से प्रयोगकर्ता टेक्स्ट बाक्स के अंदर



टेक्स्ट बदल सकता है या नहीं यह निश्चित किया जाता है। यदि इसे "true" रखते हैं, तब प्रयोगकर्ता उस टेक्स्ट बाक्स के अंदर के घटकों को नहीं बदल सकता या उसके अंदर कोई नया टेक्स्ट नहीं डाल सकता है। यदि इसे false रखते हैं, तब

उसमें टेक्स्ट डाला जा सकता है।

iv. **Multiline :-** by default टेक्स्टबाक्स एक लाइन दर्शाता है। लेकिन बड़ा टेक्स्ट डालना है, तब उसे इस प्रापर्टी से हम एक से अधिक लाइन में दर्शा सकते हैं।

v. **Scroll bar :-** यदि टेक्स्ट बाक्स में स्क्रोल बार लगाना है, तब इस प्रापर्टी का प्रयोग किया जाता है।

vi. **Password Char :-** यदि डाला गया टेक्स्ट "\*" के रूप में दर्शाना है, तब इस "True" किया जाता है।

vii. **Text :-** टेक्स्ट बाक्स में default (पहले से ही) कोई टेक्स्ट दर्शाना हो तब इस प्रापर्टी का प्रयोग होता है। साधारणतः जब कोई टेक्स्ट बाक्स बनाते हैं, तब उसमें TextBox1 आदि लिखा होता है, यदि यह डाटा नहीं चाहिए तब इस प्रापर्टी से उसे मिटा सकते हैं।

viii. **Max Length :-** इस प्रापर्टी से टेक्स्ट बाक्स में प्रयोगकर्ता द्वारा टेक्स्ट डालने की सीमा तय की जाती है।

ix. **Password char :-** इस प्रापर्टी से टेक्स्ट बाक्स में डाला हुआ कैरेक्टर आपके दिए हुए कैरेक्टर में परिवर्तित हो जाता है।

## Frame Control



इस कंट्रोल का प्रयोग एक प्रकार के option button को एक समूह (group) में रखने के लिए प्रयोग होता है। इसके प्रयोग से आप एक ही फॉर्म पर दो अलग अलग option button का समूह बना सकते हैं। एक frame के अंदर के option button को चुनसे दूसरे frame के option button पर कोई अंतर नहीं पड़ता है। इस कंट्रोल की निम्न कुछ महत्वपूर्ण प्रापर्टी है।

1. Caption :- frame पर कौनसा टेक्स्ट दर्शाना है, यह निश्चित किया जाता है।
2. click :- जब प्रयोगकर्ता frame के किसी option को क्लिक करता है, तब क्या कार्य होना चाहिए यह इस प्रापर्टी में निश्चित किया जाता है।

## Check box



जब कोई चेकबाक्स को सिलेक्ट किया जाता है तब उसके आगे "X" दिखाई देता है। साधारणतः यह कंट्रोल जब कोई विकल्प चालू या बंद रखना है, तब बनाया जाता है। उदाहरण "underline" चालू रखना है, या बंद रखना है। आप एक से ज्यादा चेकबाक्स एक ग्रुप में रख सकते हैं, जिसमें एक से ज्यादा विकल्प चालू या बंद रखा जा सकता है। इस कंट्रोल की निम्न कुछ महत्वपूर्ण प्रापर्टी है।

- i. caption :-चेकबाक्स के सामने क्या टेक्स्ट दर्शाना है, यह निश्चित किया जाता है।
- ii. value :- यह प्रापर्टी चेक बाक्स चालू रखना है या नहीं यह निश्चित करती है।

Item	Charge	Amount
<input checked="" type="checkbox"/> Box Parking	Charge	30
<input type="checkbox"/> Box Insurance	Charge	
<input checked="" type="checkbox"/> Home Delivery	Charge	100
<input checked="" type="checkbox"/> Delicate Item	Charge	300

Get total Charge: 430

III. click :- चेकबाक्स को सिलेक्ट करने के बाद क्या action होना चाहिए यह निश्चित किया जाता है। इसे समझने के लिए हम एक फार्म लेते हैं, जिसमें चेक बाक्स को सिलेक्ट कर विभिन्न चार्ज सेट किये जा सकते हैं, तथा उनसे योग प्राप्त किया जा सकता

है।

*Public Class Form1*

*Public a, ins, homdel, delic, tot As Integer*

*Private Sub CheckBox1\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged*

*If CheckBox1.Checked Then*

*a = 30*

*TextBox1.Text = 30*

*End If*

*If CheckBox1.Checked = False Then*

*a = 0*

*TextBox1.Text = a*

*End If*

*End Sub*

*Private Sub CheckBox2\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox2.CheckedChanged*

*If CheckBox2.Checked Then*

*ins = 100*

```
    TextBox2.Text = ins
End If
If CheckBox2.Checked = False Then
    ins = 0
    TextBox2.Text = ins
End If
End Sub
```

```
Private Sub CheckBox3_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
CheckBox3.CheckedChanged
```

```
    If CheckBox3.Checked Then
        homdel = 100
        TextBox3.Text = homdel
    End If
    If CheckBox3.Checked = False Then
        homdel = 0
        TextBox3.Text = homdel
    End If
End Sub
```

```
Private Sub CheckBox4_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
CheckBox4.CheckedChanged
```

```
    If CheckBox4.Checked Then
        delic = 300
        TextBox4.Text = delic
    End If
```

```
If CheckBox4.Checked = False Then
    delic = 0
    TextBox4.Text = delic
End If
End Sub
Private Sub Button1_Click(ByVal sender As System.`,
ByVal e As System.EventArgs) Handles Button1.Click
    tot = ins + delic + homdel + a
    TextBox5.Text = tot
End Sub
```

## Button

इस कंट्रोल का कमांड बटन या पुश बटन बनाने के लिए प्रयोग होता है। किसी भी फॉर्म में यह सबसे महत्वपूर्ण कंट्रोल है। यह बटन किसी कार्य को चालू करने, संपादित करने के लिए प्रयोग होती है, जैसे Open, Print, Ok आदि। इस बटन के ऊपर लिखा हुआ टेक्स्ट उसका कार्य बताता है। इस कंट्रोल की निम्न कुछ महत्वपूर्ण प्रापर्टी है।



- i. **Caption** :- इस प्रापर्टी में बटन का नाम निश्चित किया जाता है, जो टेक्स्ट इसमें लिखते हैं, वह बटन के ऊपर दिखाई देता है।
- ii. **Image** :- यदि कमांड बटन में कोई पिक्चर को डालना है, तब इस प्रापर्टी का प्रयोग किया जाता है।
- iii. **Background** :- बटन के रंग सेट करने के लिए इस प्रॉपर्टि का प्रयोग किया जाता है।
- iv. **Font** :- बटन पर जो टेक्स्ट लिखा होता है, उसका फान्ट सेट करने के लिए इस प्रॉपर्टि का प्रयोग किया जाता है।

- v. **Cancel** :- यदि कोई कार्य खत्म या बंद करने के लिए कमांड बटन बनाना है, तब इस प्रापर्टी का प्रयोग किया जाता है।

### Label Control



इस कंट्रोल का प्रयोग फॉर्म में कोई टेक्स्ट डालने के लिए होता है, जिसे प्रयोगकर्ता नहीं बदल सकता है। जैसे कंपनी का नाम, या फॉर्म का टाइटल या फिल्ड का नाम आदि। इस कंट्रोल की निम्न महत्वपूर्ण प्रापर्टी है

- i. **Autosize** :- यह प्रापर्टी लेबल के अंदर के टेक्स्ट के आकार के अनुसार लेबल का आकार निश्चित करती है।
- ii. **Wordwrap** :- इस प्रापर्टी से लेबल की मार्जिन खत्म होने पर स्वयं ही दूसरे लाइन में आ जाता है। तथा टेक्स्ट स्वयं को लेबल के आकार में सेट कर लेता है।

### Combo box



यह कंट्रोल text box एवं list box का मिश्रण है। इसमें प्रयोगकर्ता दि गई सूची में से डाटा सिलेक्ट कर सकता है एवं प्रयोगकर्ता टाईप कर सूची से इच्छित डाटा प्राप्त कर सकता है। इस कंट्रोल की निम्न कुछ महत्वपूर्ण प्रापर्टी है।

**locked** :- इस प्रापर्टी में यह बताया जाता है, की प्रयोगकर्ता कोई मान टाईप कर सकता है या नहीं।

**style** :- इस प्रापर्टी से किस प्रकार का combo box दर्शाना है, यह निश्चित किया जाता है। इसमें मुख्यतः तीन प्रकार के combo box होते हैं। 1. simple combo box 2. Drop down combo box 3. drop down list combo box

**simple combo box** :- इस प्रकार के कॉम्बो बाक्स यह साधारण लिस्ट बाक्स के समान ही दिखाई देते हैं, लेकिन इसके साथ एक टेक्स्ट बाक्स जुड़ा रहता है। प्रयोगकर्ता इस टेक्स्ट में मान डाल कर उसे कॉम्बो बाक्स में जोड़ सकता है। इस प्रकार के

कॉम्बो बाक्स में प्रयोगकर्ता कोई नया item डाल सकता है, या दिए हुए सूची में से किसी एक को चुन सकता है।

**drop down combo box:-** इस प्रकार के कॉम्बो बाक्स भी साधारण लिस्ट बाक्स के समान ही दिखते हैं। लेकिन उनका स्क्रीन पर प्रदर्शन, डाटा डालने की पद्धति, एवं डाटा के चयन के गुण यह अलग होते हैं। तीनों प्रकार के कॉम्बो बाक्स में यह सबसे सरल कंट्रोल है। जब तक प्रयोगकर्ता सूची को नहीं खोलता है, तब तक यह कॉम्बो बाक्स एक ही लाइन का दिखता है। प्रयोगकर्ता किसी एक घटक को चुन सकता है, या सीधे टाईप कर नया घटक सूची में जोड़ सकता है।

**Drop down list combo box :-** इस प्रकार के कॉम्बो बाक्स में सभी item नहीं दर्शाये जाते हैं। अपितु जब कोई प्रयोगकर्ता उसे क्लिक करता है, तब उसमें के कुछ item दिखाई देते हैं। उदाहरण के लिए जैसे ms-word के toolbar में के font list होती है। यह साधारण लिस्ट बाक्स के समान ही कार्य करता है, तथा स्क्रीन पर कम जगह घेरता है। इस कंट्रोल की निम्न कुछ महत्वपूर्ण प्रापर्टी है।

**additem :-** सूची में नया item डालने के लिए इस प्रापर्टी का प्रयोग होता है। इसका सिन्टेक्स निम्न

`combo1.additem item.index`

इसमें item यह string प्रकार का रहता है, तथा index स्टेटमेंट से लिस्ट बाक्स के आयटम सूचीबद्ध दिखती है। index स्टेटमेंट यह वैकल्पिक है। यदि आपने यह index स्टेटमेंट नहीं दिया है, तब वह item सूची में सबसे नीचे आता है।

**Removeitem :-** लिस्ट बाक्स में से कोई item हटाने के लिए इस प्रापर्टी का प्रयोग होता है। इस का सिन्टेक्स `combo1.removeitem index`



**Clear :-** काम्बो लिस्ट बाक्स में से सभी item हटाने के लिए इस प्रापर्टी का प्रयोग होता है।



**listCount:** - लिस्ट बाक्स के item की गणना (count) करने के लिए इस प्रापर्टी का प्रयोग होता है।

*Combo Box*

*Public Class Form1*

*Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load*

*ComboBox1.Items.Add("weekdays")*

*ComboBox1.Items.Add("year")*

*End Sub*

*Private Sub ComboBox1\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged*

*ComboBox2.Items.Clear()*

*If ComboBox1.SelectedItem = "weekdays" Then*

*ComboBox2.Items.Add("Sunday")*

*ComboBox2.Items.Add("Monday")*

*ComboBox2.Items.Add("Tuesday")*

*ElseIf ComboBox1.SelectedItem = "year" Then*

*ComboBox2.Items.Add("2012")*

*ComboBox2.Items.Add("2013")*

```
ComboBox2.Items.Add("2014")
```

```
End If
```

```
End Sub
```

```
End Class
```

### **OLE (object linking and embedding)**

आप अपने एप्लिकेशन में OLE आब्जेक्ट को डाल सकते हैं। विंडो की बहुतसी एप्लिकेशन से डाटा आप अपने वी.बी. डॉट नेट के एप्लिकेशन में डाल सकते हैं। यह कंट्रोल या तो उस आब्जेक्ट के link को hold करता है। या आब्जेक्ट को embed करता है, यह उस कंट्रोल के setup पर निर्भर होता है।

**linking :-** जब आप दूसरे एप्लिकेशन से link करते हैं, तब ole कंट्रोल दूसरे एप्लिकेशन के साथ link बनाता है, जिससे यदि मूल एप्लिकेशन में उस आब्जेक्ट में कोई बदलाव होता है, तब वह बदलाव आपके एप्लिकेशन में उस आब्जेक्ट में दिखाई देता है।

**embedding :-** जब ole किसी आब्जेक्ट को embed करता है, तब ole कंट्रोल उस आब्जेक्ट की कॉपी आपके एप्लिकेशन में रखता है, लेकिन इसमें मूल एप्लिकेशन और आपके एप्लिकेशन में उस आब्जेक्ट की कोई link नहीं रहती है। यदि मूल एप्लिकेशन में उस आब्जेक्ट में कोई बदलाव किया है, वह बदलाव आपके एप्लिकेशन में नहीं होता है। यदि आपको फाइल या आब्जेक्ट को link करना है, तब Link चेक बाक्स को चालू करना पड़ता है।

## नये ActiveX Control

वी.बी. डॉट नेट में standard टूल के अतिरिक्त बहुतसे टूल है, इसको चालू करने के लिए "Ctrl+T" बटन को प्रेस करे। आपको component का डायलॉग बॉक्स दिखाई देगा, उनमेसे इच्छित विकल्प को चुन कर अलग अलग टूल आप अपने टूल बार में ला सकते है। साधारणतः निम्न तीन विकल्प का प्रयोग अधिक होता है।

1. "Microsoft ADO Data Control (6.0) ADOBD
2. Microsoft Common Dialog Box Control 6.0 (SP3)
3. Microsoft Windows common Controls 6.0 (sp4)

## Common dialog control

वी.बी. डॉट नेट में निम्न प्रकार के common dialog control है। इस प्रकार के कंट्रोल यह एक समान डायलॉग बॉक्स बनाते है। उदाहरण के लिए आपको किसी अनुप्रयोग में प्रिन्ट करने के लिए डायलॉग बॉक्स बनाना है, तब आप इस प्रकार के कंट्रोल का प्रयोग कर यह काम बहुत आसानी से कर सकते है। इस प्रकार कंट्रोल में निम्न क्रियाओं के लिए डायलॉग बॉक्स बनाये जा सकते है।

- a. फाइल को खोलने या सेव करने हेतु
- b. प्रिन्ट विकल्प में काम करने हेतु
- c. टेक्स्ट के फॉन्ट एवं रंग निश्चित करने हेतु
- d. सहायता प्रणाली बनाने हेतु

standard tool bar में निचे की ओर "Dailog" विकल्प में क्लिक करने पर इसके घटक दर्शाये जाते है। इसे चालू करने के लिए "Ctrl +T" बटन को दबाए। इसके बाद आपको component dialog box दिखाई देता है।

आप कोई भी कॉमन डायलाग कंट्रोल को फॉर्म में डाल कर तथा उसकी प्रापर्टी निश्चित कर प्रयोग कर सकते है। डिजाइन के समय कोई भी कॉमन कंट्रोल एक हि ऑयकॉन के रूप में दिखाई देता

है। लेकिन उसका वास्तविक प्रभाव run time के समय उसमें दिए गये प्रापर्टी के अनुसार होता है।

**Common dialog control** का प्रयोग करना

जिस फॉर्म पर आपको इस प्रकार के कंट्रोल को डालना है, उस फॉर्म पर इस टूल को उठा कर रखे। निम्न प्रकार के कोड का प्रयोग उस कॉमन कंट्रोल के कार्य निश्चित किया जाता है।

**showopen** :- यह कोड फाइल खोलने के लिए आवश्यक डायलॉग बाक्स को दर्शाता है।

**showsave**:- यह कोड फाइल को सेव करने के लिए आवश्यक डायलॉग बाक्स को दर्शाता है।

**show color** :- यह कोड कलर सीलेक्शन डायलॉग बाक्स को दर्शाता है।

**Show font** :- इस कोड से टेक्स्ट के फॉन्ट निश्चित करने का डायलॉग बाक्स दिखाई देता है।

**Showprinter** :- इस कोड का प्रयोग प्रिन्ट का डायलॉग बाक्स दर्शाने के लिए होता है।

**Showhelp** :- इस कोड का प्रयोग सहायता प्रणाली का डायलॉग बाक्स दर्शाने के लिए होता है।

उदाहरण के लिए आपको कॉमन डायलॉग बाक्स कंट्रोल का नाम **flopen** रखा है, तब उसे **file open dialog box** दर्शाने के लिए निम्न कोड लिखना होगा।

**flopen.showopen**

लेकिन किसी भी डायलॉग बाक्स को प्रयोग करने के पहले हमें उसकी कुछ प्रापर्टी निश्चित करना आवश्यक है। **common dialog control** से प्रदर्शित किये हुए डायलॉग बाक्स वास्तव में कोई कार्य नहीं करते, अपितु प्रयोगकर्ता के लिए उस कार्य से संबंधित विभिन्न विकल्प उपलब्ध कराते हैं।

फाइल खोलने या सेव करने के लिए डायलॉग बॉक्स बनाना फाइल खोलने एवं सेव करने के लिए डायलॉग बॉक्स लगभग समान ही होता है, सिर्फ "open" और "Save" बटन का अंतर होता है। फाइल खोलने के लिए showopen स्टेटमेंट का प्रयोग किया जाता है।

फाइल ओपन डायलॉग बॉक्स के निम्न प्रापर्टी का प्रयोग विभिन्न कार्यों के लिए किया जाता है।

Defaulttext	फाइल ओपन डायलॉग बॉक्स में default extention को सेट करने के लिए प्रयोग होता है।
filename	सिलेक्ट की गई फाइल को पूरा नाम एवं पाथ दर्शाने के लिए होता है।
Filetitle	सिलेक्ट की गई फाइल का नाम दर्शाने के लिए होता है।
Initdir	प्रारंभिक (initial) डायरेक्टरी सेट करने के लिए प्रयोग होता है।
maxfilesize	फाइल के नाम का अधिकतम आकार सेट करने के लिए प्रयोग होता है।

जब फाइल ओपन का डायलॉग बॉक्स खुलता है, तब एक filter फाइल के नाम को एक्सटेंशन के साथ लागू किया जाता है, ताकि डायलॉग बॉक्स उस एक्सटेंशन की सभी फाइलों को दर्शा सके। उदाहरण के लिए हमें सिर्फ .txt और .bmp फाइलों की सूची दर्शाना है तब उसे निम्न प्रकार से लिखा जाता है।

```
opfiles.filter="Text files(*.txt): .txt: bitmap files  
(.bmp):*.bmp"
```

यदि कोई प्रयोगकर्ता एक से अधिक फिल्टर का प्रयोग करना चाहता है, तब उसे फिल्टर इन्डेक्स (filter index) प्रापर्टी का प्रयोग करना आवश्यक होता है। इस प्रकार पहले फिल्टर का मान 1

निर्धारित होगा, दूसरे का मान 2 एवं तीसरे फिल्टर को मान 3 निर्धारित होता चला जाता है।

फाइल सेव करने करने के लिए डायलॉग बॉक्स में आप एक समय में एक ही फाइल सेव कर सकते हैं। फाइल सेव के डायलॉग बॉक्स के लिए निम्न flag प्रयोग किये जा सकते हैं।

cdlofnulallowmulti select	&H200	प्रयोगकर्ता को एक से अधिक फाइल एक साथ खोलने की अनुमति प्रदान करने के लिए प्रयोग होती है।
cdIOFNCreatePr ompt	&H2000	जब प्रयोगकर्ता फाइल का नया नाम टाईप करता है, तब नई फाइल बनाने के बारे में संदेश देता है।
cdIFNExplorer	&H8000 0	विंडो explorer का प्रयोग कर फाइल खोलने को डायलॉग बॉक्स दर्शाता है।
cdIFNExtension Different	&H400	यदि बताई गई फाइल का extension, टाईप किये हुए extension से भिन्न है, तब Flag संदेश दर्शाता है।
CdloFNFile must Exist	&H1000	यदि प्रयोगकर्ता ऐसे फाइल को खोलने की कोशीश कर रहा, जो उपलब्ध नहीं तब Flag संदेश देता है।
cdloFNPath must Exist	&H800	प्रयोगकर्ता यदि गलत पाथ टाईप करता है, तब एक चेतावनी संदेश दर्शाता है।

cdlo FN Help button	&H10	डायलॉग बॉक्स में help की बटन दर्शाता है।
cdloFNHide read only	&H4	read only चेक बाक्स नहीं दर्शाता है।
cdlo FN Long Names	&H200000	यह flag बड़े नाम के फाइलों को खोलने या सेव करने की सुविधा देता है।
cdloFNNoChangeDir	&H8	इस flag का प्रयोग डायलॉग बॉक्स में default या current directory निर्धारित करता है।
cdloFNNoReadOnlyReturn	&H8000	जो फाइल सिलेक्ट की गई है, वह read only नहीं है, अर्थात् उस फाइल में बदलाव किया जा सकता है।
cdloFNNoValidate	&H100	सिलेक्ट की हुई फाइल में invalid कॅरेक्टर की अनुमति प्रदान करता है।
cdloFNOverwritePrompt	&H2	प्रयोगकर्ता को एक फाइल के ऊपर दूसरी फाइल को overwrite करने की अनुमति देता है।
cdloFNShareaware	&H4000	इस flag के प्रयोग से विंडो sharing violation की गलती की उपेक्षा (ignore) करता है

### प्रिन्ट डायलॉग बॉक्स बनाना

प्रिन्ट डायलॉग बॉक्स में प्रयोगकर्ता को इच्छित आउटपुट किस तरह से लेना है, यह निर्देशित करता है। प्रयोगकर्ता उसमें कौन से पेज या डाटा, प्रत्येक पेज की कितनी प्रतियाँ, प्रिन्ट की गुणवत्ता आदि निश्चित कर सकता है। इसे चालू करने के लिए कोड में showprinter स्टेटमेंट लिखना पड़ता है।

```
Private Sub Form_Load()
```

```
sfont.ShowPrinter
```

```
End Sub
```

प्रिन्ट डायलॉग बॉक्स के निम्न flag हैं

cdlpdallpage		सभी पेज प्रिन्ट करने की बटन को default सेट करता है।
cdlpdcollage		collate चेक बाक्स को चालू रखता है।
cdlpddisableprinttofile		print to file चेक बाक्स को बंद (disable) रखता है।
cdlpdhideprinttofile		print to file चेक बाक्स को नहीं दर्शाता है।
cdlpdnselection		selection बटन को बंद रखता है।
cdlpdnowarning		यदि प्रिन्टर जुड़ा नहीं है, या प्रिन्टर बंद है, तब भी कोई चेतावनी संदेश नहीं दर्शाता है।

### Help डायलॉग बॉक्स बनाना

इस डायलॉग बॉक्स का उपयोग, प्रयोगकर्ता को प्रोजेक्ट या प्रोग्राम के संदर्भ में सहायता उपलब्ध कराने के लिए होता है। नई सहायता प्रणाली बनाने के लिए एक help file बनाना पड़ता है, तथा उस



फाइल को showhelp स्टेटमेंट से चालू किया जा सकता है। इसके बाद VB विंडो की help engine चालू करती है, तथा प्रयोगकर्ता को सुव्यवस्थित सहायता प्रणाली दिखती है।

### Tree view control

साधारणतः window explorer में हम इस प्रकार की संरचना देखते हैं, जिसमें किसी डिक्स के अंदर की डायरेक्टरी एवं फाइल्स एक खड़े संरचना में दिखाई देती है, जिस डायरेक्टरी को क्लिक करते हैं, उसके अंदर की फाइल या सब डायरेक्टरी दिखाई देती है। इसी तरह का कंट्रोल हम vb में बना सकते हैं। यह कंट्रोल विभिन्न जानकारियों को hierarchical (पदानुक्रम) रूप में दर्शाता है। यह कंट्रोल डाटा को दर्शाने के अतिरिक्त प्रयोगकर्ता को डाटा के view को परिवर्तित करने के सुविधा प्रदान करता है। इस कंट्रोल में प्रत्येक डाटा आब्जेक्ट को नोड (node) कहा जाता है। तथा उसके अंदर के डाटा को branch कहा जाता है। प्रत्येक नोड की कुछ प्रापर्टी होती है, जिनका प्रभाव डाटा के प्रदर्शन पर पड़ता है। tree view control बनाने के लिए प्रथम फॉर्म पर कंट्रोल का instance स्थित करते हैं। इसके बाद कंट्रोल की प्रापर्टी निश्चित किया जाता है। निम्न कुछ महत्वपूर्ण प्रापर्टी है।

- b. Label edit :- इस प्रापर्टी का प्रयोग मुख्य नोड को नाम (label) देने के लिए होता है। इसकी सहायता से प्रयोगकर्ता नोड में label जोड़ सकता है।
- c. line Style :- इस प्रापर्टी से node एवं branch के बीच में लाइन रखना या नहीं रखना है, यह निश्चित किया जाता है। विंडो एक्सप्लोरर में डायरेक्टरी एवं फाइल के बीच का संबंध यह लाइन द्वारा दर्शाया जाता है। साधारणतः इसे चालू रखा जाता है।

- d. imagelist:- इस प्रापर्टी का प्रयोग node के प्रयोग होने वाले फाइल के पिक्चर को रखा जाता है।
- e. sorted :- यह प्रापर्टी मुख्य node को किस क्रम में रखना है, यह निश्चित करती है।
- f. check box :- इस प्रापर्टी से प्रत्येक node के सामने चेक बाक्स रखना है या नहीं रखना है, यह निश्चित करता है।

नरेन्द्र पब्लिकेशन

## ग्राफिक्स के साथ काम करना

VB में ग्राफिक बनाने एवं उसमें सुधार करने के बहुतसे साधन उपलब्ध है। VB में अच्छे डिजाइन के फॉर्म, रिपोर्ट बनाना बाकी प्रोग्रामिंग भाषा के अनुपात में बहुत आसान है। VB में हम विंडो पर आधारित graphic package से विभिन्न ग्राफिक import कर सकते है। इसके टूल बार में एक shape control टूल होता है, जिसकी सहायता से आयताकार, गोलाकार आदि सरल डिजाइन बना सकते है।

## Grid control

वी.बी. डॉट नेट में बहुत से कार्य करने के लिए ऐरे का प्रयोग किया जाता है। ऐरे को किसी भी डाटा टाईप में declare किया जा सकता है। कंट्रोल ऐरे को भी declare किया जा सकता है, जो एक वेरिएबल ऐरे के तरह कार्य करती है। इसमें उस प्रोजेक्ट के सभी फॉर्म आब्जेक्ट का प्रयोग एक ही नाम एवं समान प्रापर्टी मान के द्वारा किया जा सकता है। मूलतः ऐरे दो प्रकार के होते है। 1 single dimensional array , एवं multidimensional array । ग्रीड कंट्रोल में two dimensional ऐरे का प्रयोग होता है। single dimensional ऐरे उसके नाम के अनुसार सिर्फ एक हि दिशा या दशा में कार्य करते है, उनमें डाटा या तो एक रो में रहता है, या एक कॉलम में रहता है। लेकिन multidimensional ऐरे में डाटा एक से अधिक अर्थात रो और कॉलम दोनो में होते है। उदाहरण के लिए हम अलग अलग शहरो का अलग अलग महीनो का तापमान का रिकार्ड रखते है। इसके लिए हमें multidimensional ऐरे का प्रयोग करना पडाता है।

शहर	जनवरी	फरवरी	मार्च	अप्रैल	मई
दिल्ली	21	24	32	40	44

मुंबई	22	25	32	36	38
भोपाल	18	21	30	35	38
नागपुर	21	25	33	40	45

इस टेबल में two dimensional ऐरे का प्रयोग कर ग्रीड बना सकते हैं। जिसे रो और कॉलम के माध्यम से प्रदर्शित किया जाता है। ग्रीड कंट्रोल टेबल डाटा को प्रदर्शित करने का एक सरल तरीका है। प्रयोग कर्ता scroll bar की सहायता से सभी डाटा देख सकता है, तथा उसमें बदलाव कर सकता है। ग्रीड कंट्रोल बनाने के लिए निम्न पदों का प्रयोग करें। यह standard toolbar में प्रदर्शित नहीं होता है।

- ctrl +T की बटन प्रेस करें। जिससे component dialog box दिखाई देता है।
- Microsoft flex grid control 6.0 को क्लिक करें।
- Ok को क्लिक करें

ग्रीड कंट्रोल की पहली रों एवं पहले कॉलम को fixed row एवं fixed column कहते हैं। इनका प्रयोग लेबल देने के लिए किया जाता है। इन्हें row header या column header भी कहा जाता है। इस कंट्रोल की निम्न कुछ महत्वपूर्ण प्रापर्टी है।

**cols :-** इस प्रापर्टी का प्रयोग कॉलम की संख्या को निश्चित करने के लिए होता है।

**rows :** इस प्रापर्टी का प्रयोग रो की संख्या को निश्चित करने के लिए होता है।

**fixed cols :-** ग्रीड में fix columns की संख्या को निश्चित किया जाता है।

**fixed rows :-** ग्रीड में fix rows की संख्या को निश्चित किया जाता है।

**custom** :- यह ग्रीड कंट्रोल के प्रापर्टी पेज डायलॉग बॉक्स दिखाई देता है, जिसमें प्रयोगकर्ता टेबल की अन्य प्रापर्टी तय कर सकता है।

**Allow bigselection** :- यह प्रापर्टी प्रयोगकर्ता को एक साथ सभी रो या कॉलम सिलेक्ट करने के सुविधा प्रदान करता है।

**allowuser resizing** :- इसमें चार मान होते हैं।

- a. **flexlresize none** :- इस प्रापर्टी चालू करने से प्रयोगकर्ता रो या कॉलम के आकार में परिवर्तन नहीं कर सकता है।
- b. **Elexresize Both** :- इस प्रापर्टी का चालू करने से प्रयोगकर्ता रो और कॉलम के आकार में परिपर्वन कर सकता है।
- c. **flexresize column** :- इस प्रापर्टी से प्रयोगकर्ता सिर्फ कॉलम के height कम या ज्यादा करा सकता है।
- d. **flexresize row** :- इस प्रापर्टी से प्रयोगकर्ता सिर्फ रो के width को कम या ज्यादा कर सकता है।

**fill\_style** :- इस प्रापर्टी से सेल की फॉरमेंटींग सेट कर सकते हैं।

**row\_hight\_min** :- इस प्रापर्टी से रो की न्यूनतम height सेट की जा सकती है।

**grid\_liner\_fixed** :- इस प्रापर्टी से fixed line को स्क्रीन पर दर्शाना है या नहीं यह तय कर सकते हैं।

**highlight** :- यह प्रापर्टी चयनीत सेल के प्रदर्शन को व्यक्त करती है।

इसके अतिरिक्त इस कंट्रोल की कुछ और प्रापर्टी होती है, जो run time के समय कोड में लिखी जा सकती है।

**cell alignment** :- इस प्रापर्टी से सेल के अंदर के मानों का alignment निश्चित किया जा सकता है।

**cell** :- यह प्रापर्टी सेल के उस कॉलम को निर्धारित करती है, जिसका मान प्रयोगकर्ता बदलना चाहता है।

**row** :- यह प्रापटी सेल के उस रो को निर्धारित करती है, जिसका मान प्रयोगकर्ता बदलना चाहता है।

**col width** :- यह प्रापटी कॉलम की चौड़ाई को निर्धारित करने के लिए प्रयोग होती है।

**text** :- यह प्रापटी सेल के content को व्यक्त करती है।

**col alignment** :- यह प्रापटी की निश्चित कॉलम में values के alignment को व्यक्त करती है।

नरेन्द्र पब्लिकेशन

## Menus

विंडो पर आधारित प्रोजेक्ट या सॉफ्टवेयर में मेन्यू यह बहुत आवश्यक अंग है। एक मेन्यू के अंदर उससे संबंधित सभी कमांड डाली जाती है। मेन्यू बार कार्य करने के लिए सबसे आसान तरीका है, जिसमें कमांड देने के बाद वह विकल्प अदृश्य हो जाता है, तथा स्क्रीन काम करने के लिए ज्यादा जगह मीलती है। मेन्यू बार यह विंडो के टाइटल बार के नीचे होता है। सॉफ्टवेयर के अनुसार एक से ज्यादा मेन्यू होते हैं। साधारणतः यह तीन से पंद्रह के बीच में हो सकते हैं। प्रत्येक मेन्यू में उससे संबंधित कमांड होते हैं, जो उस मेन्यू को क्लिक करने से दिखाई देते हैं। जिस मेन्यू के कमांड नीचे ओर खुलते हैं, उसे drop down मेन्यू कहते हैं। एक मेन्यू के अंदर सब मेन्यू भी हो सकता है। यदि किसी मेन्यू में चार या उससे ज्यादा कमांड हैं, तब उनके बीच में separator bar डाला जाता है। कोई भी एप्लिकेशन या प्रोजेक्ट की सरलता उसके मेन्यू बार से चालू होती है। कुछ मेन्यू में कमांड का समूह नहीं होता है, अपितु वह मेन्यू ही कमांड का काम करती है, उदाहरण "Exit"। कोई भी मेन्यू स्वतंत्र रूप से काम नहीं कर सकता है, अपितु वह किसी फॉर्म के साथ जुड़ा होना आवश्यक है। साधारणतः किसी भी प्रोजेक्ट में मुख्य फॉर्म पर ही मेन्यू बार रखा जाता है, लेकिन आवश्यकतानुसार बाकी फॉर्म पर भी मेन्यू बार बनाया जा सकता है। मेन्यू बार या माउस के क्लिक द्वारा क्रियान्वित होता है, लेकिन साथ में कि-बोर्ड का भी विकल्प दिया जा सकता है। उन कि-बोर्ड के बटनों को शार्टकट कहा जाता है।

वी.बी. डॉट नेट में मेन्यू बनाने के लिए एक अलग से कंट्रोल है, जिसके द्वारा आसानी से मेन्यू बनाया जा सकता है, तथा प्रत्येक विकल्प की प्रापर्टी सेट की जा सकती है। इस कंट्रोल को मेन्यू एडिटर (menu editor) कहा जाता है।

## Menu editor

मेन्यू एडिटर की सहायता से मेन्यू बनाया एवं उसमें बदलाव किया जा सकता है। इसे चालू करने के लिए

- tool box को चालू करे
- menuStrip टूल को क्लिक कर, उसे फॉर्म पर लायें।  
आपको निम्न डायलॉग बॉक्स दिखाई देता है।

यह डायलॉग बॉक्स मुख्यतः दो भागों में बंटा होता है। पहला ऊपर का भाग जिसमें मेन्यू का नाम एवं उसकी प्रापर्टी सेट की जाती है। दूसरा भाग नीचे होता है, जिसमें बनाए हुए मेन्यू, सबमेन्यू एवं उनकी स्थिति दिखाई देती है। इस डायलॉग बॉक्स में निम्न parameter होते हैं।

- 1<sup>st</sup> Caption :- इसमें मेन्यू या सबमेन्यू का नाम जो प्रोजेक्ट में दिखाई देगा उसे टाईप किया जाता है। आप जैसे की कोई नया नाम टाईप करते हैं, वह नाम नीचे के विंडो में दिखाई देता है।
- 2<sup>nd</sup> name :- इसमें मेन्यू या सबमेन्यू का नाम जो कोडींग के समय उपयोग होता है, उसे टाईप किया जाता है।
- 3<sup>rd</sup> index :- index प्रापर्टी सेट करने के लिए प्रयोग होता है।
- 4<sup>th</sup> shortcut :- यदि किसी कमांड को शार्टकट की देना है, तब उसे इस कंट्रोल से सेट किया जाता है, किसी भी मुख्य मेन्यू को शार्टकट की नहीं दी जा सकती है।
- 5<sup>th</sup> helpcontextID: सहायता प्रणाली (help system) के लिए प्रयोग होता है।
- 6<sup>th</sup> negotiateposition :- साधारणतः इसे "None" पर रखा जाता है



7<sup>th</sup> checked :- यदि चेक प्रापर्टी चालू है, तब उस मेन्यू को क्लिक करते ही सर्वप्रथम चेक किया हुआ विकल्प सिलेक्ट होता है।

8<sup>th</sup> enable :- यदि उस मेन्यू या विकल्प को चालू रखना है, तब इस प्रापर्टी का प्रयोग किया जाता है। साधारणतः इसे बंद नहीं रखा जाता है।

9<sup>th</sup> visible :- यदि मेन्यू अदृश्य करना है, तब इसे बंद रखा जाता है।

मेन्यू अंदर के विकल्प बनाने के लिए → बटन का प्रयोग किया जाता है।

यदि किसी मेन्यू के विकल्पों में separator bar डालना है, तब "Caption" प्रापर्टी में "-" टाईप करें तथा उसको नाम "name" प्रापर्टी में सेट करें।

उदाहरण

एक मेन्यू बार बनाना है, जिसमें तीन मुख्य मेन्यू हैं, Party, Transaction, Report तथा party मेन्यू के अंदर "New", "Edit", "Delete", "Search" यह सब मेन्यू हैं, तथा transaction मेन्यू के अंदर "Cash Sale", "Credit Sale", "Payment", "Receipt", "Purchase" सब मेन्यू हैं तथा report मेन्यू के अंदर "Date wise report", "Party wise report" सब मेन्यू हैं।

solution



सर्व प्रथम new project मेन्यू में "standard Exe" को सिलेक्ट कर open करें

- ⊙ आपको एक फॉर्म दिखाई देगा, उसकी "Toolbox" में "menu" विकल्प को क्लिक करें।
- ⊙ project Menu Strip टूल को क्लिक करें।
- ⊙ मेन्यू एडिटर स्क्रीन पर दिखाई देगा, उसे

स्क्रिन पर इच्छित जगह *drag* करें, साधारणतः इस फार्म के उपर की ओर रखा जाता है।

- पहले मेन्यू का नाम टाईप करें,
- निचे की ओर **Menustrip** कंट्रोल को क्लिक करें, नया मेन्यू दिखाई देगा, इस प्रकार से मुख्य मेन्यू बना सकते है। आप मेन्यू का नाम "text" प्रॉपर्टि में भी सेट कर सकते है।
- मेन्यू के अंदर option या command डालने के लिए, वांछित मेन्यू को क्लिक करें। आपको उसके अंदर के विकल्प टाईप करने के लिए जगह प्राप्त होती, उसमे इच्छित टेक्स्ट टाईप करे।
- तरह से सभी मेन्यू एवं उसके अंदर के विकल्प बनाये। अंत में "OK" को टाईप करें। यह निम्न तरीके से दिखाई देगा।

तथा इसे run करने के बाद यह निम्न तरीके से दिखाई देगा

मेन्यू में कोड जोडना

मेन्यू एडिटर में बनायी गई संरचना में कोड नहीं जोडा जा सकता है। इसमे लिए आपको अलग से कोड लिखना पडता है, जो उस विकल्प को क्लिक करने के बाद आवश्यक काम करें। मेन्यू और उसके अंदर के कमांड यह एक कंट्रोल की तरह होते है, जो run time पर event को प्रदान करते है। जब जब प्रयोगकर्ता किसी menu या उसके अंदर के विकल्प को क्लिक करता है, तब तब click event procedure क्रियान्वित होती है।

### Dialog box

डायलॉग बॉक्स यह एक अस्थाई विंडो होती है, जिसमे कोई जानकारी, संदेश दर्शाने के लिए या कोई इनपुट लेने के लिए खुलती है। जैसे हम open विकल्प खोलते है, तब ओपन का डायलॉग बॉक्स खुलता है, तथा जब हम कोई फाइल को सिलेक्ट कर open बटन को क्लिक करते है, तब वह विंडो बंद हो जाती

है। डायलॉग बॉक्स में मेन्यू नहीं रहते, एवं इसमें minimize एवं maximize की बटन नहीं होती है। डायलॉग बॉक्स के आकार में परिवर्तन नहीं कर सकते हैं। कोई भी डायलॉग बॉक्स यह मुख्य एप्लिकेशन के साथ ही जुड़ा होता है। डायलॉग बॉक्स मुख्यतः दो प्रकार के होते हैं।

1. Modal 2. modeless

### **Model डायलॉग बॉक्स**

इस प्रकार के डायलॉग बॉक्स में प्रयोगकर्ता किसी प्रकार का इनपुट देता है। उदाहरण save कमांड देने के बाद "Save" का डायलॉग बॉक्स आता है, उसमें आपको "Save" "cancel" दो बटनों में से किसी एक को क्लिक करना पड़ता है। या कोई डायलॉग बॉक्स कोई संदेश दर्शाता उसे भी modal डायलॉग बॉक्स कहा जाता है। जब तब modal डायलॉग बॉक्स चालू रहता है, तब तक उस एप्लिकेशन में कोई अन्य क्रिया नहीं की जा सकती है।

### **Modeless डायलॉग बॉक्स**

इस प्रकार के डायलॉग बॉक्स चालू रहते समय भी, एप्लिकेशन के दूसरे हिस्से में काम किया जा सकता है। इस प्रकार के डायलॉग बॉक्स कम ही प्रयोग होते हैं। जैसे ms-word का find and replace डायलॉग बॉक्स, जिसमें इसे बंद न करते हुए मुख्य डॉक्युमेंट पर काम कर सकते हैं।

**Common dialog box :-** जो डायलॉग बॉक्स वी.बी. डॉट नेट में पहले से ही उपलब्ध होते हैं, या जिनका काम करने का तरीका सभी एप्लिकेशन में लगभग समान होता है, उन्हें common डायलॉग बॉक्स कहा जाता है। उदाहरण open, save के डायलॉग बॉक्स ।

Custom dialog box :- यह प्रोजेक्ट या एप्लिकेशन अनुसार बनाये जाते हैं।

किसी फॉर्म को डायलॉग बाक्स बनाने के लिए उस फॉर्म की "border style" प्रॉपर्टी को "3- fixed dialog" सेट करें।

## Input Box

inputbox() फंक्शन से एक पॉपअप डायलॉग बॉक्स बनता है। यह एक तरह से मैसेज बाक्स के समान ही कार्य करता है, लेकिन इसमें प्रयोगकर्ता प्रोग्राम के अनुसार कोई value डाल सकता है। मैसेज बाक्स को प्रयोगकर्ता मूव या आकार में बदलाव कर सकता है, लेकिन इनपुट बाक्स का आकार कम ज्यादा नहीं हो सकता है। इसमें सिर्फ OK और Cancel बटनों का प्रयोग होता है, तथा इसके साथ कोई आयकान नहीं दर्शाया जाता है। इसका सिन्टेक्स निम्न प्रकार से है।

```
A=inputBox (Prompt,[Title], [default], [xpos], [ypos],  
[help file])
```

```
a=inputBox ("Enter value of A", "Example of input box",  
"0", 500,750)
```

डायलॉग बॉक्स बनाते समय निम्न बातों का ध्यान रखें

1. जहाँ तक संभव है, common डायलॉग बॉक्स का प्रयोग करें, इससे कोड सरल हो जाते हैं। तथा संपूर्ण एप्लिकेशन में समानता आती है।



2. जब सिर्फ एक ही मान का इनपुट लेना है, तब input बाक्स का प्रयोग करें।

3. कोई संदेश दर्शाने के लिए मैसेज

बाक्स का प्रयोग करे।

नरेन्द्र पब्लिकेशन

## Multiple document interface (MDI)

MDI (multiple document interface) यह एक फॉर्म के अंदर एक या अधिक फॉर्म को क्रियान्वित करने की सुविधा प्रदान करता है। कुछ अनुप्रयोग जैसे Ms-word, Excel में इस प्रकार के फॉर्म बहुतायत प्रयोग होते हैं, जिसमें एक साथ एक या अधिक डॉक्युमेंट खोल सकते हैं। तथा प्रयोगकर्ता उन सभी फॉर्म को स्क्रीन जमाकर रख सकता है। उस एप्लिकेशन के मेन्यू एवं टूल सभी खुले हुए डॉक्युमेंट में कार्य करते हैं। वी.बी. डॉट नेट में इस प्रकार के कार्यक्षेत्र (working environment) बनाने की क्षमता है। वी.बी. डॉट नेट में दो प्रकार की स्टाइल उपलब्ध हैं

1. Single Document Interface (SDI)

2. Multiple Documents Interface (MDI)

SDI विकल्प एक विंडो में प्रयोगकर्ता काम कर सकता है। यदि दूसरी विंडो खोलना है, तब पहली विंडो बंद करना पड़ता है। नोटपैड, पेंट इस प्रकार के इंटरफेस में आते हैं।

MDI विकल्प में एक मुख्य विंडो के अंदर एक से अधिक विंडो एक बार में खोल सकते हैं। इसके अंदर कि विंडो में minimize, restore कर सकते हैं।

वी.बी. डॉट नेट में मुख्य विंडो जो की बाकी सभी विंडो को contain करती है, उसे MDI form के रूप में देखा जाता है। जो मुख्य विंडो होती है, उसे Parent window कहा जाता है, तथा उसके अंदर के सभी विंडो को child window कहा जाता है। child window यह साधारण फॉर्म के रूप में होती है। जब कोई child window को minimize किया जाता है, तब मुख्य विंडो वैसे ही रहती है, लेकिन child window छोटे आयकान के रूप में दिखती है। लेकिन जब parent window को minimize किया जाता तब उसके अंदर की सभी child window स्वयं ही

minimize हो जाती है। MDI फॉर्म पर जबतक सभी कंट्रोल सही तरीके से align नहीं होते हैं तबतक कंट्रोल को सीधे नहीं रख सकते। MDI फॉर्म बनाने के लिए

- Project मेन्यू को क्लिक करें।
- Add window form विकल्प को क्लिक करें
- Add new item का डायलॉग बॉक्स दिखाई देगा। उसमें MDI parent form विकल्प क्लिक करें।
- नया फॉर्म दिखाई देगा जिसका background रंग गहरा होता है।

mdi प्रकार के फॉर्म को अलग से दर्शाया जाता है। mdi फॉर्म के आयकान में एक फॉर्म के अंदर छोटे फॉर्म को दर्शाया जाता है। इस आयकान के नीचे Mdi form लिखा होता है। mdi फॉर्म सभी कंट्रोल प्रयोग नहीं करते हैं, जो standard फॉर्म में होते हैं। इसमें सिर्फ वोही कंट्रोल प्रयोग हो सकते हैं, जिनमें align प्रापर्टी होती है, एवं जो runtime के समय प्रदर्शित नहीं होते हैं जैसे picturebox, timer आदि। लेकिन picture box को एक container के रूप में प्रयोग कर आप उसके अंदर उन सभी कंट्रोलों का प्रयोग कर सकते हैं, जो सीधे MDI फॉर्म में प्रयोग नहीं



किये जा सकते हैं।

MDI फॉर्म में मुख्य फॉर्म एवं child फॉर्म दोनों में मेन्यू बार हो सकता है। मुख्य फॉर्म का मेन्यू तब ही दर्शाया जाता है, जब उसके किसी child फॉर्म में मेन्यू नहीं है।

यदि child फॉर्म पर मेन्यू बार नहीं है, तब मुख्य फॉर्म का मेन्यू दर्शाया जाता है।

फॉर्म को **MDI** का **child form** बनाना

किसी भी साधारण फॉर्म को उसकी MDI child प्रापर्टी को True करके MDI child फॉर्म बनाया जा सकता है। जब किसी फॉर्म की MDI child प्रापर्टी चालू की जाती है, तब वह फॉर्म केवल उसी Parent MDI फॉर्म के कार्यक्षेत्र में ही क्रियान्वित होता है। कोई फॉर्म अलग फॉर्म है या MDI का चाइल्ड फॉर्म है, यह आप प्रोजेक्ट विंडो में देख सकते हैं, उसका आयकान अलग होता है। auto show children प्रापर्टी में true मान डाला जाता है, तब child फॉर्म वी.बी. डॉट नेट स्वतः ही एक्टिव कर दर्शाता है। एक्टिव child फॉर्म का मेन्यू मुख्य फॉर्म के मेन्यू बार पर प्रदर्शित होता है, child फॉर्म पर नहीं होता है।

MDI चाइल्ड फॉर्म के **run time** के समय गुण (characteristics) एक MDI फॉर्म एवं उसके अंदर के सभी चाइल्ड फॉर्म के निम्न गुण (characteristics) होते हैं।

1. एक MDI फॉर्म के अंदर के सभी चाइल्ड फॉर्म सिर्फ उसी MDI के अंदर कार्य कर सकते हैं। प्रयोगकर्ता उन चाइल्ड फॉर्म के आकार में बदलाव कर सकता है, तथा उन्हें मुख्य फॉर्म के अंदर मूव कर सकता है।
2. MDI पॅरेन्ट फॉर्म में मेन्यू होता है, लेकिन किसी भी चाइल्ड फॉर्म में मेन्यू नहीं होता है।
3. जब कोई चाइल्ड फॉर्म को maximize किया जाता है, तब उसका टाइटल यह MDI के टाइटल के साथ संयुक्त रूप से दिखता है।



4. autoshowchildren प्रापर्टी के द्वारा आप सभी चाइल्ड फॉर्म, जब MDI चालू होता है, तब देख सकते है।
5. जब चाइल्ड फॉर्म को minimize किया जाता है, तब वह parent form के अंदर की एक छोटे आयकान के रूप में दिखता है।
6. MDI चाइल्ड फॉर्म मॉडल नहीं होते है, अर्थात ये प्रयोगकर्ता को दूसरे MDI चाइल्ड फॉर्म को चुनने से नहीं रोकते है।
7. चाइल्ड फॉर्म को मुख्य पॅरेन्ट फॉर्म के अंदर विभिन्न तरीकों से समायोजित (arrange) किया जा सकता है।

नरेन्द्र पब्लिकेशन

## Data Access in VB.net

### ADO

एडू (ADO) एक माइक्रोसॉफ्ट ने विकसित किया टूल है, जो डाटाबेस प्रोग्रामिंग में प्रयोग होता है। यह ऑब्जेक्ट मॉडल का एक सेट है, जो इच्छित डाटाबेस से डाटा को प्राप्त करने के लिए प्रोग्रामिंग इंटरफेस उपलब्ध कराता है। यह डाटाबेस तथा प्रोग्रामिंग भाषा के बीच एक माध्यम के रूप के कार्य करता है। एडू की सहायता से प्रोग्रामर वीबी डॉट नेट में प्रोग्राम से साथ इच्छित डाटाबेस फाइलों को जोड़ कर, उनके डाटा पर कार्य कर सकते हैं। एडू मूलतः चार घटकों से बना है जो फ़िल्ड, प्रॉपर्टी, पैरामीटर तथा एरर (errors) है, तथा इसमें बारह ऑब्जेक्ट प्रयोग होते हैं, कनेक्शन, कमाण्ड, रिकॉर्ड सेट, इमिडिएट, बैच, ट्रांजैक्शन, रिकॉर्ड, स्ट्रिम, पैरामीटर, फ़िल्ड, प्रॉपर्टी तथा एरर इसके बारह ऑब्जेक्ट हैं, जिससे मिलकर यह बना है।

एडू की सहायता से डाटा को प्राप्त करने, उसमें सुधार करने, एवं बदलाव करने के निम्न पद हैं—

- डाटाबेस से जुड़ने के लिए कनेक्शन ऑब्जेक्ट का निर्माण
- डाटा को एक्सेस करने के लिए रिकॉर्ड सेट ऑब्जेक्ट का निर्माण
- कनेक्शन खोलना
- कनेक्शन की प्रापर्टि सेट करना
- रिकॉर्ड सेट खोलकर इसे पॉप्यूलेट करना/भरना तथा खुले हुए फंक्शन को पैरामीटर के रूप में इच्छित टेबल का नाम या एस. क्यू.एल. स्टेटमेंट पास करना।
- लाये गये डाटा पर इच्छित सर्च करना या अन्य प्रोसेसिंग करना

- बदलाव को Update अथवा UpdateBatch की सहायता से सुरक्षित करना।
- रिकॉर्ड सेट बंद करना।
- कनेक्शन बंद करना

इस पूरे क्रम में डाटाबेस कनेक्शन खुला रहता है। इसलिए इसे संयोजित डाटा आर्किटेक्चर (connected data architecture) पर आधारित प्रोग्रामिंग कहते हैं और यही कमी के कारण एडू डॉट नेट का जन्म हुआ।

एडू को कई भाषाओं यथा ए.एस.पी., पावर बिल्डर तथा वी.बी. डॉट नेट फॉर एप्लिकेशन्स का सपोर्ट प्राप्त है।

### एडू डॉट नेट (ADO.NET)

वी.बी. डॉट नेट यह ado.net का प्रयोग करता है। जब आपको अप्लिकेशन कोई डाटा बेस जोड़ना है, तब सर्व प्रथम data sources के बारे में सोचना आवश्यक है। data source वह होता है, जहाँ से आपको डाटाबेस प्राप्त करना है। Microsoft SQL server यह .net का डिफाल्ट डाटा प्रोवाइडर है। लेकिन इसके अतिरिक्त विजुअल बेसिक.नेट यह ODBC (Open Data Base Connectivity) का भी समर्थन करता है। अर्थात् आप अन्य स्रोतों से भी डाटाबेस प्राप्त कर सकते हैं। आप SQLconnection object, Oracle connection आदि का प्रयोग डाटाबेस प्राप्त करने के लिए कर सकते हैं।

एक बार डाटा सोर्स के साथ कनेक्शन जुड़ने के बाद, उस डाटा के साथ कार्य करने के लिए data adapter की आवश्यकता होती है। डाटाबेस जुड़ने के बाद वह डाटाबेस मूल अप्लिकेशन से disconnect हो जाता है, उसका सक्रिय सहभाग नहीं रहता है, इसलिए data adapter आवश्यक है। data adapter वास्तव में

sql स्टेटमेंट है, जो डाटा पर कार्य करता है। oracle डाटा के लिए `SQLDataAdapter` object एवं `OracleDataAdapter` की आवश्यकता होती है।

एडू डॉट नेट लायब्रेरी का एक संकलन है जिसका उपयोग माइक्रोसॉफ्ट SQL सर्वर, माइक्रोसॉफ्ट सर्वर, ऑरेकल, एस.क्यू.एल., माइक्रोसॉफ्ट एक्सेल इत्यादि जैसे विभिन्न संसाधनों का उपयोग करते हुए शक्तिशाली डाटाबेस के निर्माण में होता है। एडू डॉट नेट डॉट नेट फ्रेमवर्क के विभिन्न क्लास पर निर्भर करता है जो आग्रह (requests) को प्रोसेस कर डाटाबेस प्रणाली तथा उपयोगकर्ता के बीच ट्रांजीशन को सम्पन्न करता है। ये सारी क्रियाएं सामान्यतः डाटासेट क्लास के माध्यम से नियंत्रित होती है। डाटासेट क्लास, डाटाबेस इंजिन तथा विंडोज कंट्रोल (जो पूरे यूजर इंटरफेस का होता है) के बीच का एक माध्यम (agent) होता है। डाटासेट ऑब्जेक्ट किसी भी प्रकार की सूची को व्यवस्थित करने में सक्षम होता है चाहे वह डाटाबेस इन्वायरमेंट में बना हो अथवा न बना हो।

एडू डॉट नेट एप्लिकेशन में डाटाबेस के फीचरों को उपयोग करने के अलावा आप एक्स.एम.एल. के लाभ भी पूरी तरह से प्राप्त कर सकते हैं क्योंकि एक्स.एम.एल. डाटासेट क्लास के द्वारा पूरी तरह से सपोर्ट किया जाता है। एप्लीकेशन के एक्स.एम.एल. सपोर्ट देने के लिए डॉट नेट में `system.xml.dll` लायब्रेरी उपलब्ध होता है। परन्तु जब आप एक्स.एम.एल. का उपयोग करते हैं तो आपको इस लाइब्रेरी को आयात (import) नहीं करना होता है। इसका कारण यह है कि जब आप **New Project** डायलॉग बॉक्स में विंडोज फॉर्मस एप्लीकेशन का निर्माण करते हैं तो `system.Xml.dll` नेमस्पेस आपके एप्लीकेशन में स्वतः ही जुड़ जाता है। डॉट नेट

फ्रेमवर्क में जो क्लास एक्स.एम.एल. को लागू करता है वह Systems.Xml नेमस्पेस से परिभाषित होता है।

## ADO and RecordSet

प्रत्येक DBMS में डाटा एक विशिष्ट फॉरमेट में स्टोर होता है। जैसे Ms-access में .mdb फाइल के रूप या FoxPro में .dbf के रूप में। यदि आपको कोई एप्लिकेशन एक DBMS से दूसरे में परिवर्तित करना है, तब संपूर्ण एप्लिकेशन फिरसे बनानी पडती है। पुराने समय में साधारणतः कोई एप्लिकेशन एक DBMS में बनाई जाती थी। लेकिन वर्तमान कम्प्यूटर एप्लिकेशन का प्रयोग बड़े एवं बहुत जटिल व्यापारिक कामों में भी किया जाता है। जिसके लिए आवश्यकतानुसार अलग अलग DBMS एप्लिकेशन का प्रयोग करना पडता है। उदाहरण के लिए पहले किसी बैंक की एक मुख्य शाखा के लिए FoxPro में प्रोग्राम बनाया गया था, लेकिन बाद में उस बैंक की प्रत्येक शाखा को जोडने के लिए Access की आवश्यकता है, उस दशा में पुराना प्रोग्राम का पूरा डाटा access में परिवर्तित करना पडेगा, जो की बहुत बडा काम हो जाता है। इसके लिए ऐसी एप्लिकेशन बनाने की आवश्यकता है, जो अलग अलग डाटाबेस फॉरमेट का एक साथ प्रयोग कर सके। यह काम ऐसे कर सकते है, कि कोई एप्लिकेशन हमे एक ही फॉरमेट में कमांड दे। तथा यह कमांड विभिन्न DBMS के डाटा फॉरमेट को एक ही फॉरमेट में परिवर्तित करे। इसके लिए एप्लिकेशन और डाटाबेस के बीच में एक translator होता है, जो किसी भी डाटा फॉरमेट को अपने एप्लिकेशन के अनुसार translate करता है।



यह काम हम ODBC (open database connectivity) के माध्यम से कर सकते हैं। odbc यह एप्लिकेशन और डाटाबेस के बीच में कॅम्प्यूनीनेट करने की पद्धति बताता है। जो एक API (application programming interface) के साथ जुड़ी होती है। API प्रोग्रामिंग में प्रयोग होने वाला फंक्शन या कमांड का समूह है, जो किसी कार्य को करने के लिए बनाया जाता है। ODBC driver इसमें translator का कार्य करता है। यह driver एप्लिकेशन के विभिन्न कमांड को ग्रहण करता है, तथा उन्हें टारगेट डाटाबेस को समझने योग्य बनाता है। साथ ही डाटाबेस से विभिन्न परिणाम प्राप्त कर उन्हें एप्लिकेशन के फॉरमेट में परिवर्तित करता है।

### **OLEDB (Object linking and embedding database)**

यह एक डाटाबेस की संरचना (database architecture) है, जो एप्लिकेशन को एक समान डाटा के रूप स्टोर करने की सुविधा प्रदान करती है। OLE DB यह इंटरफेस की व्याख्या करता है, इसलिए इसे low level कहा जाता है, तथा OO(object oriented) में सीधे प्रयोग किया जा सकता है, C++ या वी.बी. डॉट नेट.ने में इसे सीधे प्रयोग किया जा सकता है। इसके लिए VB में activeX data Object (ADO) टूल दिया है।

कम्प्यूटर का प्रयोग बढ़ने से distributed एप्लिकेशन का चलन बहुत बढ़ा है। distributed एप्लिकेशन में दो भाग होते हैं एक front end जो client कम्प्यूटर पर चलता है, और दूसरा back end जो server पर चलता है। distributed एप्लिकेशन का मुख्य कार्य यह बड़े काम को अलग अलग कम्प्यूटर में विभाजीत कर आसान करना है। वी.बी. डॉट नेट में तीन प्रकार data access के पद्धतियाँ हैं।

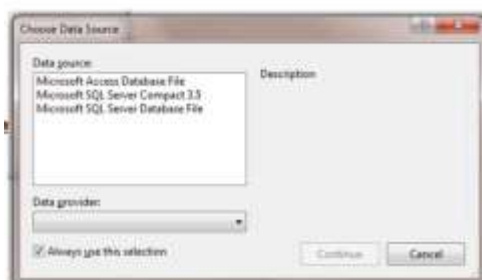
1. **Data access Objects (DAO)** :- इस तकनीक में सिर्फ लोकल (local) डाटाबेस के साथ ही कार्य किया जा सकता है। यह तकनीक दूर के डाटा को प्राप्त करने के लिए नहीं है।
2. **Remote Data Objects(RDO)**:- यह मूलतः दूर के (remote) डाटाबेस के कार्य करने के लिए प्रयोग होती है। वी. बी. डॉट नेट के 4.0 version से यह तकनीक उपलब्ध है। यह तकनीक में relation database को प्राप्त करती है, जैसे MS-SQL, Oracle आदि
3. **ActiveX Data Object(ADO)** :यह तकनीक version 6.0 से उपलब्ध हुई है। यह आधुनिकतम तकनीक है, जिसमें लगभग सभी फॉरमेट के डाटा को प्राप्त किया जा सकता है। ADO यह relational और non-relational डाटाबेस दोनों के साथ कार्य कर सकता है। इसके साथ यह दूसरे प्रकार के डाटा source के साथ भी कार्य करने की सुविधा प्रदान करता है, जैसे e-mail, file system, spreadsheets आदि।

एक बार data adapter सेट होने के बाद, उसके सहायता से dataset का निर्माण किया जाता है। कोड के द्वारा डाटाबेस से डाटा पर कार्य किया जाता है, उसे dataset कहा जाता है।

### **server Explorer** की सहायता से डाटा प्राप्त करना

vb.net में डाटा बेस प्राप्त करने के लिए सर्वप्रथम, वांछित डाटाबेस फाइल को जोड़ना आवश्यक है। vb.net में server explorer आपको विभिन्न प्रकार के data source से जुड़ने की सुविधा प्रदान करता है। इसे चालू करने के लिए

➔ view मेन्यू को क्लिक करें।



database explorer विकल्प को क्लिक करें। आपको database

explorer की विंडो दिखाई देगी, उसमें उपर की ओर connect to database विकल्प को क्लिक करें। आपको “choose Data Source” का डायलाग बाक्स दिखाई देता है।

- उसमें उपलब्ध Data Source की सूची दिखाई देती है। उसमें से इच्छित डाटा सोर्स सिलेक्ट करें।
- Continue विकल्प को क्लिक करें।



➤ Add new data source विकल्प को क्लिक करें। Data source configuration wizard का डायलॉग बाक्स खुल जाता है।

➤ उसमें database ऑयकान को क्लिक करें, तथा Next बटन को क्लिक करें।

➤ दूसरे विंडो में “new connection” बटन को क्लिक करें।

➤ आपको Add connection का डायलॉग बाक्स दिखाई देता है। उसमें data sources में “Micro Soft Access Database File (OLE DB)” दिखाई देता है। यदि दूसरा डाटा सोर्स से डाटा प्राप्त करना है, तब उसके आगे “change” बटन को क्लिक करें।



➤ नीचे की ओर डाटाबेस फाइल को नाम सिलेक्ट करने के लिए



“Browse” बटन को क्लिक करे।

- इच्छित फाइल सिलेक्ट करे।
- यदि कनेक्शन सही प्रकार से हो रहा है, या नहीं यह जाँचने के लिए नीचे की ओर “Test Connection” बटन को क्लिक करे। यदि जाँच सही पाई जाती है, तब “ok” बटन को क्लिक करे।
- अब पुराना डायलॉग बाक्स दिखाई देता है। उसमें “Next” बटन को क्लिक करे।
- अंत में उस डाटाबेस में की सभी टेबल की सूची दर्शाई जाती है। उनमें से इच्छित या सभी टेबल को सिलेक्ट करे।
- टेबल के साथ आप उसमें फिल्ड आदि भी सिलेक्ट कर सकते है।
- Finish बटन को क्लिक करे।

अब आपके solution exporent विंडो में उस डाटाबेस फाइल का नाम जुड जाता है।

server explorer विंडो में भी डाटाबसे जुड जाती है। किसी टेबल की संरचना एवं डाटा देखने के लिए, उसे टेबल पर बाया क्लिक करे, तथा

**Recordsource Property:** - कनेक्शन जुडने के बाद रिकार्ड कहां पर आना है, यह निश्चित किया जाता है। इस प्रापर्टी में टेबल का नाम, stored करने की प्रोसीजर या SQL स्टेटमेंट आदि निश्चित की जाती है। हमेशा SQL स्टेटमेंट का प्रयोग करना बेहतर होता है, इसमें संपूर्ण टेबल सिलेक्ट करने के बजाय सिर्फ उपयुक्त फिल्ड हि सिलेक्ट की जा सकती है। इस प्रापर्टी पर क्लिक करने से property page का डायलॉग बॉक्स दिखाई देता है। इसमें command type में चार विकल्प होत है।

टाईप	विवरण
adCmdUnknown	unknown कमांड यह इसका default रहता है।
adCmdText	कमांड टेक्स्ट बाक्स में SQL कमांड दिया जा सकता है।
adCmdTable	डाटाबेस में उपलब्ध टेबलों की सूची दर्शाता है।
adCmdStoredProc	डाटाबेस में उपलब्ध सभी प्रोसीजर और क्युरीज दर्शाता है।

यदि आप access के डाटाबेस के साथ काम कर रहे हैं, तब adcmdtable प्रापर्टी सिलेक्ट करें। तथा नीचे लिस्ट बाक्स में उस डाटाबेस के टेबल की सूची आती है, उन से इच्छित टेबल को सिलेक्ट करें

connection और recordsource अलग अलग सेट करने के बजाय आप custom property को सिलेक्ट कर सकते हैं। ADODC से डाटा को जोडा जा सकता है, तथा सूचनाए प्राप्त की जा सकती है, लेकिन यह कंट्रोल स्वयं कोई डाटा नहीं दर्शा सकता है। इसके लिए हमें अलग से Bind control की आवश्यकता होती है। यह कंट्रोल टेक्स्ट बाक्स, लिस्ट बाक्स आदि हो सकते हैं। datasource प्रापर्टी यह इन स्टेटमेंट में क्या दर्शाना वह निश्चित करती है। इसके बाद datafield प्रापर्टी सेट करनी पडती है, जो उस datasource की फिल्ड निश्चित करती है। जब ADODC को क्लिक करते हैं, तब उससे जुड सभी bind कंट्रोल में डाटा दर्शाया जाता है। ADODC में साधारण कंट्रोल के अतिरिक्त निम्न और कुछ कंट्रोल का प्रयोग होता है।

कंट्रोल	विवरण
datagrid	रो और कॉलम में आकार में स्क्रीन दिखाई देती

	है, जिसमें डाटा में बदलाव किया जा सकता है। इसमें रो यह रिकार्ड को दर्शाता है, एवं कॉलम यह फिल्ड को दर्शाता है।
datacombo	यह कंट्रोल जो datasource की फिल्ड इससे जुडी है, उसके सभी रिकार्ड को दर्शाता है।
datalist	यह कंट्रोल जो datasource की फिल्ड इससे जुडी है, उसकी सभी रिकार्ड लिस्ट के रूप में दर्शाता है।
hierarchicla flexGrid	यह sort, merge तथा table format जैसे क्रियायें करने के लिए प्रयोग होती है।
masked edit	इस कंट्रोल में बताई गई कंडीशन के अनुसार डाटा लेता है, तथा फॉरमेट किया हुआ डाटा दर्शाता है।
richtext box	इसमें बड़े आकार का टेक्स्ट डाला जा सकता है, तथा उसमें बदलाव किया जा सकता है। इसमें टेक्स्ट फॉरमेंटींग की कुछ सुविधाएं उपलब्ध हैं।

इनमें maskEdit और Rich text box यह कंट्रोल ADO और DAO के साथ कार्य कर सकती हैं, बाकी कंट्रोल सिर्फ ADODC के साथ कार्य करती हैं।

साधारणतः एप्लिकेशन में टेक्स्ट बाक्स, चेक बाक्स आदि का प्रयोग किया जाता है। लेकिन लिस्ट कंट्रोल के प्रयोग से प्रयोगकर्ता सीधे इच्छित रिकार्ड को क्लिक कर डाटा को सिलेक्ट कर सकता है।

इसके लिए वी.बी. डॉट नेट में list box एवं combo box का प्रयोग किया जाता है।

जब हम ADODC में list box या combo box डाटाबेस के किसी एक फिल्ड के डाटा को दर्शाते हैं। list box में list प्रापर्टी से डाटा फिल्ड को सिलेक्ट किया जाता है। combo box की text प्रापर्टी

को प्रयोग ADO DC में bound के लिए किया जाता है, तथा इसमें प्रयोगकर्ता इच्छित डाटा को क्लिक द्वारा सिलेक्ट कर सकता है या उस डाटा को टाईप कर उसे सिलेक्ट कर सकता है।

सरल एप्लिकेशन के standard list box और combo box का प्रयोग होता है। लेकिन जटिल या बड़े एप्लिकेशन में Datalist यह DataCombo का प्रयोग करना बेहतर होता है। इन दोनों कंट्रोल में डाटा को सिलेक्ट करने की एवं डाटा प्राप्त करने के ज्यादा अच्छी सुविधा है। उदाहरण के लिए एक छात्र के नामों की लिस्ट डाली है, एवं उसके साथ course की सूची डाली है। इसमें जब हम किसी छात्र को क्लिक करते हैं, तब उसक संबंधित course दूसरे सूची में सिलेक्ट हो जाता है। साधारण listbox या combo box में एक साथ एक ही फिल्ड की सूची दर्शाई जा सकती है। datalist या datacontrol यह standard toolbar में दिखाई नहीं देते हैं, इसके लिए Project मेन्यू के component विकल्प से "Microsoft datalist control 6.0(OLE DB) को सिलेक्ट करे। ADO DC में recordset प्रापर्टी से विभिन्न डाटा कंट्रोल में इच्छित टेबल का रिकार्ड डाला जाता है। इस प्रापर्टी से एक समय में एक ही रिकार्ड को प्राप्त किया जा सकता है।

## Data control

इस कंट्रोल से डाटा को प्राप्त कर सकते हैं, तथा डाटा में बदलाव कर सकते हैं। इसके विभिन्न प्रापर्टी के सेटिंग से इच्छित डाटाबेस फाइल एवं डाटा को प्राप्त कर सकते हैं। data control स्वयं कोई डाटा प्रदर्शित नहीं करता है। इसका काम सिर्फ डाटा के बहाव (flow) को नियंत्रित करना है। डाटा को फॉर्म पर प्रदर्शित करने के लिए अन्य कंट्रोल का प्रयोग करना पड़ता है। यह कंट्रोल जैसे listbox, textbox आदि है, यह data control के साथ डाटा पर कार्य करते हैं, उन्हें data aware (bound) कहा जाता है, तथा

डाटा को जोडने के क्रिया को binding कहा जाता है। data bound control निम्न कार्यो को करते है

1<sup>०</sup> बाउंड कंट्रोल, डाटाबेस के डाटा को वी.बी. डॉट नेट डाटा आब्जेक्ट में लोड करते है।

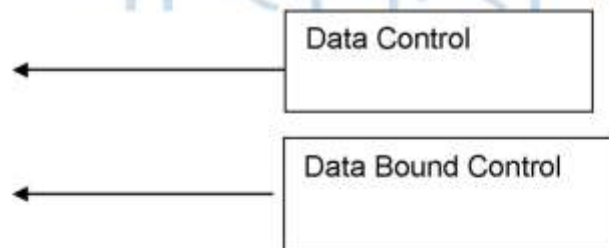
2<sup>०</sup> प्रयोगकर्ता द्वारा डाटा रिकार्ड के लिए की गई नई रिक्वेस्ट को सिलेक्ट करते है।

3<sup>०</sup> प्रयोगकर्ता द्वारा की जाने वाली गलतियों को ढुंढ कर बताते है।

4<sup>०</sup> डाटाबेस इंटीग्रिटी नियमों को लागू करते है।

प्रत्येक डाटा बाउंड कंट्रोल अपने आप रिकार्ड को data control की स्थिति के अनुसार दर्शाता है। इन डाटा बाउंड कंट्रोल के द्वारा सूचना को डाटाबेस में स्टोर किया जाता है, तथा डाटा में बदलाव किया जा सकता है।

नरेन्द्र पब्लिकेशन





डाटा बाउंड कंट्रोल निम्न प्रकार के डाटा पर कार्य कर सकते है ।

Control	Data type	Property
label	text, numeric, date	Text
text box	text, memo, numeric, date	text
check box	logical	Value
picture box	long binary	Picture
image	long binary	Picture

यह bound कंट्रोल, टेबल (table) की बताई गई फिल्ड के साथ जुड जाता है। data control यह एक रिकार्ड से दूसरे रिकार्ड पर जाने के लिए प्रयोग होता है।

डाटा कंट्रोल की निम्न प्रापर्टी अधिक प्रयोग होती है।

1. **connect** :- इस प्रापर्टी में किस एप्लिकेशन से डाटाबेस लेना है, वह तय कर सकते हैं, इसमें विभिन्न डाटाबेस एप्लिकेशन की सूची आती है, जैसे **access, FoxPro, excel** आदि
2. **databaseName** :- इस प्रापर्टी में कौनसी डाटाबेस फाइल प्रयोग करना है, उसका नाम दिया जाता है।
3. **recordsource** : यदि आप चाहते हैं, कि डाटा कंट्रोल केवल एक ही टेबल के साथ कार्य करे तब इसके लिए आप **recordsource** प्रापर्टी पर क्लिक कर उसमें उपस्थित टेबल की सूची में से टेबल को चुन सकते हैं। यदि आप कुछ चुनी हुई सूचनाओं को एक टेबल या एक से ज्यादा टेबल में से सिलेक्ट करना चाहते हैं, तब इसके लिए आपको **SQL** के स्टेटमेंट में रिकार्ड सोर्स प्रापर्टी में लिखना होगा। इसके लिए आपको **recordsource** प्रापर्टी में **Query Def** का नाम लिखना होगा, जो की डाटाबेस में **Sql** स्टेटमेंट के रूप में स्टोर है।
4. **Record set type** :- data control को किस प्रकार का record set बनाना है, यह इस प्रापर्टी से निश्चित किया जाता है। इसमें निम्न तीन मान हैं
  - a. **dynaset type** :- इस प्रकार के रिकार्डसेट में एक से अधिक टेबल के रिकार्ड को **add, change, delete** किया जा सकता है। इसके बदलाव टेबल पर भी आते हैं
  - b. **table type**:- एक ही टेबल के रिकार्ड को **add, change, delete** किया जा सकता है।
  - c. **snap shot type** :-ऐसे डाटा का समूह जिन्हे किसी रिपोर्ट के लिए रख सकते हैं, उन्हें दर्शाता है। इस प्रकार में डाटा में बदलाव नहीं कर सकते हैं।

## Data bound कंट्रोल में डाटा दर्शाना

जब आप data control में डाटाबेस का नाम निश्चित कर लेते हैं, तथा उसकी बाकि प्रापर्टी सेट कर लेते हैं। उसके बाद आप को डाटा दर्शाने या डाटा में बदलाव करने के लिए data bound कंट्रोल की आवश्यकता होती है, जैसे text box, list box आदि। data bound कंट्रोल में निम्न प्रापर्टी सेट करना पडता है।

data source :- इस प्रापर्टी में जिस data control को इस कंट्रोल को जोडना है, उसका नाम सिलेक्ट करे।

data field :- इस प्रापर्टी में कंट्रोल के अंदर डाटाबेस की कौन सी फिल्ड दर्शाना चाहिये वह निश्चित कर सकते हैं।

निम्न प्रजेक्ट में हम admission का फॉर्म लेते हैं, जिसमे stud नाम की Ms-access की फाइल को जोड कर डाटा एंन्टी का फॉर्म बनाते हैं। इस टेबल में निम्न फिल्ड है

फिल्ड का नाम	टाईप	width
name_St	text	30
address	text	50
course	text	20
daofadd	date	
addfee	single	
telphoneno	text	25

फॉर्म की डिजाइन

प्रापर्टी

object	property	value
label1	caption	Student

object	property	value
label2	caption	Student Address



		name			s
label3	caption	For Course	label4	caption	Date of Admission
label5	caption	ICT	TextBox1	text	
	borderStyle	fixed single		multiline	true
	font (size)	14		dataSource	data1
	alignment	center		datafield	name_St
data1	connect	access	TextBox2	text	
	database Name	stud.mdb		dataSource	data1
	recordSource	admission		datafield	addresses
textBox3	text		textBox4	text	
	datasource	data1		dataSource	data1
	datafield	course		datafield	daofadd
textb4	text		textBox5		
	datasource	data1		dataSource	data1

	datafield	addfee		datafiel d	telphon eno
Butto n 1	caption	add record	Button 2	caption	Modify Record
Butto n 3	caption	delete record	Button 4	caption	First record
Butto n 5	caption	last reord	Button 6	caption	Previou s
Butto n7	caption	last	form1	caption a	Admiss ion Form



### Code

```
Private Sub Command1_Click()  
Data1.Recordset.AddNew  
TextBox1.SetFocus  
End Sub
```

```
Private Sub Command2_Click()  
Data1.UpdateRecord  
End Sub
```

```
Private Sub Command3_Click()  
Data1.Recordset.Delete
```

```
Data1.Recordset.MoveNext
```

```
If Data1.Recordset.EOF = True Then
```

```
Data1.Recordset.MovePrevious
```

```
End If
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Data1.Recordset.MoveFirst
```

*End Sub*

*Private Sub Command5\_Click()*

*Data1.Recordset.MoveLast*

*End Sub*

*Private Sub Command6\_Click()*

*Data1.Recordset.MovePrevious*

*If Data1.Recordset.BOF = True Then*

*Data1.Recordset.MoveFirst*

*End If*

*End Sub*

*Private Sub Command7\_Click()*

*Data1.Recordset.MoveNext*

*If Data1.Recordset.EOF = True Then*

*Data1.Recordset.MoveLast*

*End If*

*End Sub*

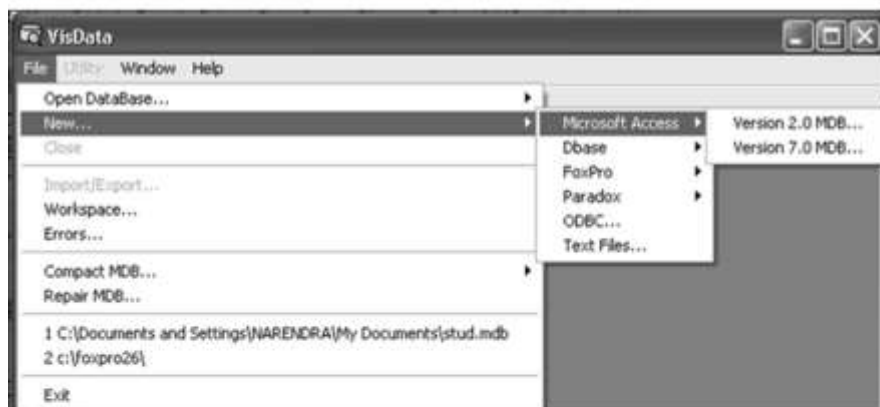
**Data Grid view**

डाटाबेस फाइल में के डाटा को सूची के रूप में दर्शाने या संपूर्ण डाटा के वर्कशीट के रूप में दर्शाने के लिए data grid view का प्रयोग किया जाता है। इसके लिए vb.net में data grid view नाम का टूल उपलब्ध है।

**Vb.net में डाटाबेस फाइल बनाना**

वी.बी. डॉट नेट स्वयं डाटाबेस फाइल नहीं बना सकता है, लेकिन इसके “data manager” की सहायता से हम विभिन्न डाटाबेस एप्लिकेशन की फाइल बना सकते हैं।

Tools मेन्यू के “add in manager” अप विकल्प को क्लिक करें।



आपको addin का डायलॉग बाक्स दिखाई देगा उस में “visual Data manager” विकल्प को क्लिक करें।

आपको एक नई विंडो दिखाई देगी। इसमें नई डाटाबेस फाइल बनाने के लिए “new” विकल्प को क्लिक करे।

इसमें विभिन्न डाटाबेस एप्लिकेशन के नाम दर्शाए जाते है। इनमें इच्छित एप्लिकेशन को क्लिक करे। यदि हम Ms-access करते है, तब उसमें दो विकल्प आते है। यदि आप windows-3.1 में काम कर रहे है, तब Version 2.0 को क्लिक करे, अथवा version 7.0 को क्लिक करे।

इसके बाद डाटाबेस फाइल का नाम दें।

अब आपको विंडो में दो स्क्रीन दिखाई देगी। इस डाटाबेस में नई टेबल जोडने के लिए “database window” में कहीं पर right click करे। तथा new table विकल्प को क्लिक करे। इसके बाद निम्न विंडो दिखाई देगी। नई फिल्ड जोडने के लिए “add field” बटन पर क्लिक करे। फिर एक नई विंडो खुलेगी, जिसमें इच्छित फिल्ड का नाम, उसका टाईप, तथा width डाले। तथा Ok के बटन को क्लिक करे। इसी तरह सभी फिल्ड का नामए टाईप आदि डालने के बाद close बटन को क्लिक करे। आपको सभी फिल्ड की सूची दिखाई देगी। यदि आप कोई फिल्ड हटाना चाहते है, तब remove field को क्लिक करे। यदि किसी फिल्ड पर index करना है। तब Add index बटन पर क्लिक करे। add index

विंडो में इंडेक्स का नाम, कौन से फिल्ड पर इंडेक्स करना है, आदि बतायें , तथा Ok के बटन को क्लिक करे।

जब टेबल अंतिम रूप में बन जाये, तब “Build Table” बटन को क्लिक करें । अब नई डाटाबेस टेबल यह “database window” में दिखाई देगी। यदि बनाई गई टेबल में कोई परिवर्तन करना है, तब उस टेबल पर right click करे। तथा “design” विकल्प पर क्लिक करे। आपको फिर से table structure विंडो दिखाई देगी। इसमे इच्छित बदलाव कर फिर से close बटन को क्लिक करे। अब visual data manger बंद कर वी.बी. डॉट नेट के मुख्य विंडो में वापस आ सकते है।

### रिकार्ड में बदलाव करना

#### रिकार्ड को जोडना (Adding record)

किसी टेबल में नया रिकार्ड जोडने के लिए Addnew स्टेटमेंट का प्रयोग होता है।

`adostud.recordset.Addnew`

किसी रिकार्ड में बदलाव करने के लिए Update स्टेटमेंट का प्रयोग होता है।

`adostud.recordset.update`

किसी रिकार्ड के बदलाव cancel करने के लिए CancelUpdate स्टेटमेंट का प्रयोग होता है।

`adostud.recordset.CancelUpadate`

किसी रिकार्ड को ढुंढने के लिए Find स्टेटमेंट का प्रयोग होता है, इसका स्टेटमेंट निम्न प्रकार से है।

`object.find(criteria, skiprows, searchdirection, start)`

उदाहरण

`adostud.recordset.Find “Stud_name=”Amit”, 0,  
adSearchBackward,`

ADO.net object से हम ज्यादा अच्छी एप्लिकेशन बना सकते है। ADO.net डाटा कंट्रोल से हम डाटा को दुबंकर बदलाव कर सकते है, लेकिन ADO.net आब्जेक्ट में डाटा प्राप्त करने तथा उसमे बदलाव की ज्यादा अच्छी सुविधा उपलब्ध है।

ADO.net object के ADO.net data control से निम्न तरीको में ज्यादा लाभप्रद है।

- i. ADO आब्जेक्ट में डाटा के ऊपर इच्छित validation लगा सकते है। जिससे गलत डाटा डालने के संभावना कम हो जाती है।
- ii. हम ऐसे डाटाबेस एप्लिकेशन बना सकते है, जिसमे प्रयोगकर्ता को interface नहीं है।
- iii. ADO आब्जेक्ट में SQL स्टेटमेंट का प्रयोग किया जा सकता है।

ADO object में डाटा को प्राप्त करने के निम्न तीन पद है।

1. connection      2. Recordset      3. Command

ADO आब्जेक्ट बनाने से पहले ADO object library चालू करना आवश्यक है, इसे project मेन्यू से references विकल्प में से "Microsoft activeX data object 2.0 Library" को क्लिक उसे चालू किया जा सकता है।

Connection :- जब किसी डाटाबेस के डाटा को प्राप्त करना है, तब सबसे प्रथम हमे उस data source से साथ connection जोडना पडता है। यह "Connection" आब्जेक्ट यह एप्लिकेशन का डाटाबेस एप्लिकेशन के साथ संबध जोडता है। प्रत्येक connection object यह सभी low level object(recordset and command object) के साथ संबध जोडता है।

**Recordset:** - यह आब्जेक्ट टेबल से संपूर्ण डाटा को प्राप्त करता है, इसका काम query का डाटा देना है। इस आब्जेक्ट का प्रयोग डाटा के बदलाव में भी होता है।

**Command :-** कमांड आब्जेक्ट का प्रयोग रिकार्ड को प्राप्त करना, बड़े operation क्रियान्वित करना तथा डाटाबेस के संरचना में बदलाव करना है। इसका प्रयोग को देखना, उस डाटा को प्राप्त करना जो डाटाबेस ने भेजे है।

ADO.net पर आधारित एप्लिकेशन बनाने के लिए निम्न पद प्रयोग होते हैं।

1<sup>st</sup> Connection आब्जेक्ट को बनाना, इसके लिए connection string की आवश्यकता होती है। connection string में data source के संबंधों की जानकारी होती है। इसमें data source का नाम, user identification और पासवर्ड रहता है।

2. Connection को चालू करना

3. Data source के query को क्रियान्वित करना।

4. जो डाटा query से प्राप्त हुआ है, उसमें इच्छित बदलाव करना।

5. Connection को बंद करना।

**Connection बनाना एवं चालू करना**

connection आब्जेक्ट बनाने के लिए निम्न पदों का प्रयोग होता है।

एक वेरिएबल को व्यक्त किया जाता है,

```
dim con as ADODB.connection
```

```
set con=new connection
```

इसमें ADODB स्टेटमेंट से नया connection आब्जेक्ट बनता है।

connection आब्जेक्ट बनाने के लिए हमें connectionString और connectionTimeout प्रापर्टी सेट करना पड़ता है। इसमें निम्न प्रापर्टी है।

प्रापर्टी	विवरण
provider	जो connection में प्रयोग होने डाटाबेस प्रोवाइडर का नाम निश्चित होता है। यह प्रापर्टी कौन से प्रकार का डाटाबेस प्रयोग हो रहा है, यह दर्शाती है।
data source	connection के प्रयोग होते हुए data source का नाम निश्चित किया जाता है। उदाहरण Ms-access
userID	connection को चालू करते समय लगने वाले प्रयोगकर्ता को ID
password	connection को चालू करते समय लगने वाला पासवर्ड

अलग OLEDB प्रोवाइडर के निम्न पैरामीटर है।

प्रोवाइडर	पैरामीटर
Micorsoft Jet	Microsoft.Jet.OLEDB 3.51
Oracle	MSDAORA
Microsoft ODBC driver	MSDASQL
SQL Server	SQLOLEDB

निम्न उदाहरण में jet database का connection जोड़ा है।

```
con.connectionstring=
```

```
"Provider=Microsoft.jet.oledb.3.51;" & _ "dataSource =  
C:\My Document\Stud.mbd"
```

इसका सिन्टेक्स निम्न प्रकार से है।

```
connection.open [connectionString], [userID],  
[Password]
```

उदाहरण



```
dim con as ADODB.connection
set con = new connection
con.connection="Provider=Microsoft.jet.oledb.3.51;" & _
"dataSource = C:\My Document\Stud.mbd"
con.open
connection को बंद करना
connection को बंद करने के लिए "close" स्टेटमेंट का प्रयोग
होता है। लेकिन कोई connection आब्जेक्ट बंद करने के बाद भी
वह मेमोरी में रहता है। उसे संपूर्ण तरीके से निकालने के लिए
वेरिएबल को मान "nothing" करना पड़ता है।
con.close
set con= nothing
```

### Data Adaptor

एक बार डाटा सोर्स के साथ कनेक्शन जुड़ने के बाद, उस डाटा के साथ कार्य करने के लिए **data adapter** की आवश्यकता होती है। डाटाबेस जुड़ने के बाद वह डाटाबेस मूल अप्लिकेशन से **disconnect** हो जाता है, उसका सक्रिय सहभाग नहीं रहता है, इसलिए **data adapter** आवश्यक है। यह **ado.net** का ही एक हिस्सा है। यह **data source** तथा **dataset** को जोड़ने का कार्य करता है। **data adapter** वास्तव में **sql** स्टेटमेंट है, जो डाटा पर कार्य करता है। **data adaptor** से आप **data source** में **select, insert, updata , delete SQL** आदि कार्य कर सकते है। **oracle** डाटा के लिए **SQLDataAdapter** object एवं **OracleDataAdapter** की आवश्यकता होती है। निम्न **data adaptor** का प्रयोग **vb.net** में होता है।

**OleDbDataAdapter** :- इस का प्रयोग OLE से प्राप्त डाटा जैसे Ms-access आदि डाटाबेस फाइल के लिए प्रयोग होता है।

**SqlDataAdapter**:- SQL server से प्राप्त डाटाबेस फाइल के लिए इसा प्रयोग होता है।

**ODBCDataAdapter** :- ODBC से प्राप्त डाटा के लिए प्रयोग होता है।

**OracleDataAdapter** :- oracle से प्राप्त डाटा के लिए प्रयोग होता है।

### Recordset बनाना एवं खोलना

data source से संबंध बनाने के बाद, हमे recordset आब्जेक्ट बनाना पडता है। यह recordset आब्जेक्ट में डाटाबेस वांछित डाटा लिया जाता है। आप इसे recordset आब्जेक्ट से या command आब्जेक्ट से यह काम कर सकते है। recordset यह एक या एक से अधिक टेबल से आवश्यक फिल्ड और रिकार्ड लेता है। recordset का मुख्य उद्देश डाटाबेस से डाटा को प्राप्त कर उसे जाँचना एवं उसमे इच्छित बदलाव करना है। recordset object में निम्न काम करते है।

1. कौन से रो (फिल्ड) चाहिए वह बताना
2. उस रो को ढुढना
3. एक से अधिक रो है, तब उन्हे सही क्रम (order) में रखना।
4. रो को जोडना, या निकालना
5. data source को update करना

recordset आब्जेक्ट में बहुतसे प्रापर्टी और मेथड है, recordset आब्जेक्ट बनाने के लिए निम्न पद्दो का प्रयोग होता है।

```
dim rd as ADO.DB.recordSet
```

**set rd = new connection**

यह आब्जेक्ट बनाने के लिए हम open मेथड का प्रयोग किया जाता है। इसका सिन्टेक्स निम्न प्रकार से है।

*recordset.open [source], [activeconnection], [cursortype], [locktype], [option]*

**Source** :- यह वैकल्पिक स्टेटमेंट है, यह टेबल के नाम या SQL स्टेटमेंट को एक वेरिएबल के रूप दर्शाने के लिए होता है।

**activeconnection** :- यदि आपने पहले से connection बना लिए है, तब उस connection का नाम इसमें दिया जाता है। यदि connection नहीं बनाया है, तब यह आरग्युमेंट connection बनाने के लिए आवश्यक सूचनायें देता है।

**cursor Type** :- cursor यह recordset की सूचना रखता है।

एक recordset यह चुन हुए रिकार्ड को प्रदर्शित करता है, जिससे रिकार्ड में बदलाव करना या देखना आसान होता है। रिकार्ड यह टेबल एक रो को दर्शाता है, लेकिन ADO में टेबल नहीं दर्शाता है, इसलिए इसमें cursor यह वांछित रिकार्ड को दर्शाता है। चार प्रकार के cursor होते हैं।

- a. **dynamic (adopendynamic)** :- इसमें नेटवर्क में दूसरा प्रयोगकर्ता recordset में परिवर्तन कर सकता है, तथा नया recordset डाल सकता है। तथा सभी प्रकार की movement कर सकता है।
- b. **keyset (adopenkeyset)** :- यह dynamic cursor के समान ही कार्य करता है, लेकिन जो दूसरा प्रयोगकर्ता रिकार्ड को बनाता है, या मीटाता है, वह आप नहीं देख सकते हैं। लेकिन रिकार्ड में किये हुए बदलाव आप देख सकते हैं। सभी प्रकार की movement कर सकता है।

- c. `static(adopenstatic)`:- सभी प्रकार बदलाव दूसरा प्रयोगकर्ता करता है, वह आप नहीं देख सकते हैं।
- d. `forward-only (adopenforwardonly)` :- यह `dynamic cursor` के समान ही कार्य करता है, लेकिन इसमें सिर्फ आगे के तरफ `movement` की जा सकती है।

नरेन्द्र पब्लिकेशन

## Error Handling

कोई भी प्रोग्राम बनाने के बाद उसे जाँचना बहुत आवश्यक काम होता है। प्रोग्राम में मुख्यतः तीन प्रकार की गलतियाँ होती हैं।

. Syntax Error

. Logical Error

. Run Time Error

**Syntax error** : प्रोग्राम के कोड लिखने में यदि कोई गलती होती है, तब उसे **syntax error** कहते हैं। यह सामान्यतः वेरिएबल का नाम गलत लिखना, डाटा अवयवों का गलत प्रयोग, विराम चिन्हों गलत जगह लगाना या नहीं लगाना आदि हो सकती है। इस प्रकार की गलतियों को वी.बी. डॉट नेट स्वयं ही बताता है, तथा जब तक उसे सुधारा नहीं जाता तब तक प्रोग्राम क्रियान्वित नहीं होता है।

**Logical error** : इस प्रकार की गलती प्रोग्राम के क्रियान्वन में कोई बाधा उत्पन्न नहीं करती, लेकिन उससे मिलने वाला उत्तर गलत होता है। उदाहरण के आपको दो अंको का जोड़ करना है, लेकिन आपने "+" के जगह "x" ऑपरेटर का प्रयोग किया है। इस प्रकार की गलतियों को वी.बी. डॉट नेट स्वयं ही ढुंढ सकता अपितु यह गलतियाँ प्रोग्रामर को स्वयं ढुंढ कर सुधारना पड़ता है। लेकिन वी. बी. डॉट नेट में इस प्रकार की गलतियों को ढुंढने के लिए कुछ tool दिए हैं। इस प्रकार की गलतियों को खोजने में अधिक समय लगता है। विशेषतः जब कोई डाटाबेस के साथ काम कर रहे, तथा उसके डाटा में परिवर्तन कर स्टोर कर रहे हैं। इस प्रकार की गलतियों को कम करने के लिए हमें प्रोग्राम को कम्प्यूटर पर बनाने के पहले उस का संपूर्ण प्रारूप पेपर पर बना लेना चाहिए, तथा उसे Flow chart के सहायता से प्रोग्राम की संरचना बनायें। कोई भी प्रोग्राम को लागू करने से पहले उसे सभी स्थितियों में जाचना आवश्यक होता है। प्रोग्राम को जाँचने के प्रक्रिया को **debugging**

कहा जाता है। प्रोग्रामे में पाये जाने वाले त्रुटियों को bug कहा जाता है।

वी.बी. डॉट नेट में जब आप कोई कोड लिखते, तब यदि उस कोड में कोई गलती है, तब वी.बी. डॉट नेट तुरंत उसे एक मैसेज बाक्स द्वारा दर्शाता है। साधारणतः उस मैसेज बाक्स के सूचना से कौन सी गलती है, यह समझा जा सकता है।

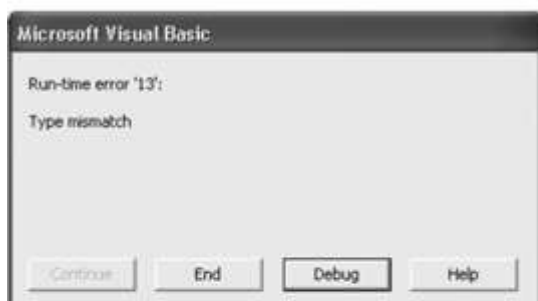
इसके अतिरिक्त वी.बी. डॉट नेट प्रोग्राम में और एक ऐरर हो सकती है जिसे run time error कहा जाता है। यह ऐरर जब कोई स्टेटमेंट उस कार्य करने की कोशिश करता है, जो उसके लिए नहीं बनाई गई है। उदाहरण के लिए किसी numeric वेरिएबल में string स्टोर कर रहे हैं, वी.बी. डॉट नेट को error handler सिर्फ run time ऐरर को ढुंढता है।

**Run time error :-** इस प्रकार की त्रुटिया ऊपर के दोनों प्रकार के त्रुटियों से अलग दूसरे कारणों से होती है।

निम्न स्थितियों run time error आ सकता है ।

1. ऐसे टेबल को सिलेक्ट करना जो बंद (close) किया है।
2. numeric वेरिएबल में string प्रकार का डाटा डाला है।
3. किसी एक रिकार्ड को एक समय में दो प्रयोगकर्ता (user) सेव करने के कोशिश कर रहे हैं।
4. किसी अंक को zero शून्य से डिवाइड करना
5. उस फाइल को खोलने की कोशीश करना हो अस्तित्व में नहीं है।

बहुत बार इस प्रकार की गलतियाँ प्रोग्रामर के नियंत्रण के बाहर होती है। जैसे की प्रयोगकर्ता ने प्रयोग होने वाली डाटाबेस फाइल मीटा दि आदि। ऐसी गलतियो सिर्फ दर्शाया जा सकता



है, तथा प्रोग्राम को रोक कर रखा जा सकता है। ऐसी गलतियों से बचाव के चेतावनी या सूचना देना चाहिए। उदाहरण के लिए हम एक प्रोग्राम लेते हैं, जिसमें एक मान को दूसरे मान भाग देना है। इस प्रोग्राम में हम एक संदेश दर्शाते हैं, कि किसी वेरिएबल का मान शून्य ना रखे। यदि किसी ने गलती से वेरिएबल का मान शून्य डाल दिया है, तब प्रोग्राम स्वयं ही रुक जाये।

```
if x=0
    msgbox " do not enter 0 zero value"
    x=0
    exit sub
else
    z=y/x
endif
```

Runtime एरर को संभालने के लिए एक error handler होता है। यदि वह error handler नहीं है, तब वी.बी. डॉट नेट उसे एरर को एक message box में सूचना देता है, तथा उसे संदेश के साथ एक अंक होता है, तथा प्रत्येक प्रकार के एरर के लिए अलग अंक होता है।

## Run time error handling

आप प्रोग्राम में ऐसा कोड जोड़ सकते हैं, जो प्रोग्राम में गलती का पता लगाता है। run time error handling को लागू करने के तीन मुख्य पद हैं।

1. गलती का पता लगाना एवं program flow को error handler को पुनः निदेशित करना
2. error को सही करना या error को highlight करना

3. वापस प्रोग्राम के क्रियान्वयन को चालू करना  
 वी.बी. डॉट नेट में प्रत्येक प्रोसीजर के लिए अलग अलग error handling कोड लगाना पडता है। error handler बनाने के लिए प्रोसीजर में On Error Go To स्टेटमेंट का प्रयोग किया जाता है। इस स्टेटमेंट के साथ एक नाम (label) दिया जाता है, यह लेबल यदि प्रोसीजर में कोई गलती है, तब प्रोगाम का पाईट किनसे स्टेटमेंट में जाना है, यह बताता है। उदाहरण

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
    On Error GoTo errmage
```

```
    num1 = TextBox1.Text
```

```
    num2 = TextBox2.Text
```

```
    TextBox3.Text = num1 * num2
```

```
errmage:
```

```
    MsgBox "Enter proper number"
```

```
End Sub
```

error object

वी.बी. डॉट नेट में किसी run time ऐरर को खोजने के लिए err आब्जेक्ट का प्रयोग किया जा सकता है। यह आब्जेक्ट कौन सी गलती हुई है, उसका विवरण तथा कहा हुई है, यह दर्शाता है। err आब्जेक्ट लेबल की निम्न प्रापर्टी है

number	ऐरर का नंबर दर्शाने के लिए प्रयोग होता है
description	ऐरर का विवरण दर्शाया जाता है ।
source	किस string या डाटा से ऐरर उत्पन्न हुआ है, वह दर्शाता है
helpfile	help फाइल का नाम एवं पाथ दर्शाया जाता है



प्रोग्राम का क्रियान्वित फिर से चालू करना कोई ऐरर आने के बाद उसे सुधार करने के लिए उस प्रोसीजर का क्रियान्वित चालू रखना आवश्यक होता है, जिससे वह प्रोसीजर खत्म हो और उसमें आवश्यक सुधार किया जा सके। इसके लिए resume स्टेटमेंट का प्रयोग किया जाता है।

resume	प्रोसीजर के जिस लाइन में ऐरर है, उसे फिर से क्रियान्वित किया जाता है
resume next	प्रोसीजर के जिस लाइन में ऐरर है, उसे के बाद के लाइन को क्रियान्वित किया जाता है
resume Line	प्रोसीजर की किसी इच्छित लाइन को क्रियान्वित किया जाता है

उदाहरण

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    On Error GoTo err1
    num1 = TextBox1.Text
    num2 = TextBox2.Text
    TextBox3.Text = num1 * num2
err1:
    MsgBox Err.Number
    MsgBox Err.Description
    MsgBox Err.Source
    Resume Next
End Sub
```

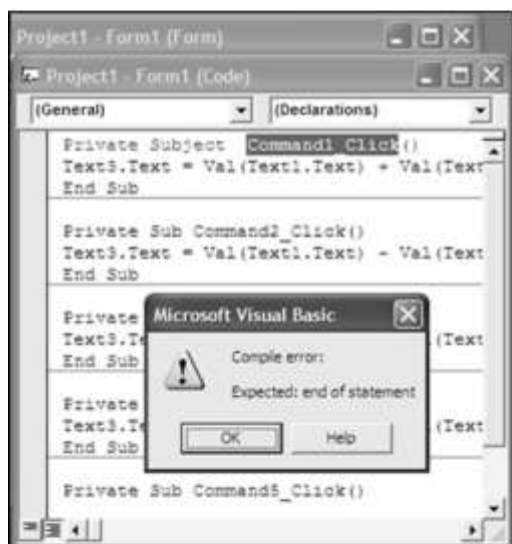
## Syntax errors

वी.बी. डॉट नेट में साधारणतः सिन्टेक्स ऐरर को पहचाना जा सकता है। गलत keyword, या कॉमा न लगाना आदि से सिन्टेक्स ऐरर

उत्पन्न हो जाती है। **syntax** ऐरर के कुछ अस्पष्ट कारण भी होते हैं, जैसे प्रोसीजर कॉल या वेरिएबल के नाम का गलत प्रयोग आदि। वी.बी. डॉट नेट त्रुटीपूर्ण लाइन में उपस्थित सिन्टेक्स ऐरर को दर्शाती है। वी.बी. डॉट नेट में कोड में की गलतीयां दर्शाने का स्तर निश्चित कर सकते हैं। इसके लिए Tool मेन्यू के Option विकल्प को क्लिक करें, तथा Editor टैब को क्लिक करें।

Auto syntax check विकल्प चालू करने से, जैसे ही आप को कोड लाइन पूर्ण करते हैं, वह उसे जाँचता है, यदि उसमें कोई सिन्टेक्स ऐरर है, तब वह तुरंत दर्शाता है।

## Scanning of Error



कोड विंडो में कोड लाइन टाईप कर **enter** की बटन दबाते हैं, तब विजुअल बेसी उस लाइन को स्कॅन कर लेता है। यदि उस लाइन में कोई गलती है, तब उसके लिए एक संदेश दर्शाया जाता है। यदि विजुअल बेसिक को गलती समझ में नहीं आ रही है, तब उस स्टेटमेंट को फिर से जाँच करता है। स्टेटमेंट के विभिन्न शब्दों में **space** लगाया

जाता है, तथा वे शब्द जिन्हें वी.बी. डॉट नेट के reserved word माने जाते हैं, उन्हें पहचाना जाता है। तथा उन शब्दों के focus कर रिकार्ड किया जाता है। जैसे इस बाक्स में वर्तनी की गलती दर्शाई गई है। कभी कभी गलत वर्तनी वाले स्टेटमेंट भी गलती नहीं दर्शाते हैं, यह इसलिए होता है क्योंकि वी.बी. डॉट नेट यह सोचता है, कि आप एकदम भिन्न उत्तर चाहते हैं। यदि आपको दिखता है, कि

कोई शब्द reformat नहीं किया है, तब शब्द की वर्तनी जाँच ले, उसके बाद स्टेटमेंट को जाँच ले।

### **Compile –time error report**

विजुलअ बेसिक प्रत्येक स्टेटमेंट को जाँचता है, लेकिन कुछ गलतीया उस लाइन में नहीं होती है, अपितु ऊपर या नीचे के स्टेटमेंट से जुड़ी होती है। जैसे if स्टेटमेंट को endif से खत्म नहीं किया है, या case की संरचना सही तरीके से नहीं लिखी गई है आदि। यद्यपि प्रोग्राम क्रियान्वयन के पहले कोड करे कम्पाइल करते समय इन गलतियों को ढुढ़ं लिया जाता है। यह एक एरर संदेश दर्शाता तथा गलती कहीं हुई है उसे highlight करता है। बड़े loop संरचना यह प्रणाली मुश्कील पैदा करती है।

### **Runtime Error Report**

कुछ गलतियाँ यह compile के समय प्रदर्शित नहीं होती है, लेकिन प्रोग्राम के क्रियान्वयन के समय दर्शाई जाती है। इस प्रकार की गलतियों का कारण सामान्यतः ऐसे वेरिएबल या फाइल का प्रयोग प्रोग्राम में करना जो अस्तित्व में नहीं है होता है। जैसे कुछ वेरिएबल पहले declare किये है, तथा बादमे उस वेरिएबल का नाम बदल दिये है, या उस वेरिएबल को हटा दिया है, इस प्रकार की गलतियाँ runtime के समय दर्शाई जाती है। यदि प्रोग्राम में प्रयोग होने वाले वेरिएबल, आब्जेक्ट, फाइल आदि की सूची है, तब इस प्रकार की गलतियों को सुधारना आसान हो जाता है।

### **Debugging Techniques**

वी.बी. डॉट नेट स्वयं गलती सुधार नहीं करता, अपितु वह ऐसे टूल उपलब्ध करता है, जो आपको यह दर्शाता है, की प्रोग्राम किस दिशा में जा रहा है, और उसके वेरिएबल और प्रापर्टी में प्रोग्राम के क्रियान्वित के समय क्या बदलाव हो रहे है। वी.बी. डॉट नेट के

debugging में breakpoints, break expression, watch expression, आदि सुविधा उपलब्ध है। यह प्रोसीजर के कोड step by step क्रियान्वित करके दर्शा सकता है, जिसमें आप उसके अंदर के वेरिएबल और प्रापर्टी के बदलाव आसानी से देख सकते हैं। इसमें edit and continue की भी सुविधा है, जिससे गलती का सुधार वही कर प्रोग्राम को आगे क्रियान्वित कर सकते हैं। debugging टूल निम्न प्रकार में आपको साहयक है।

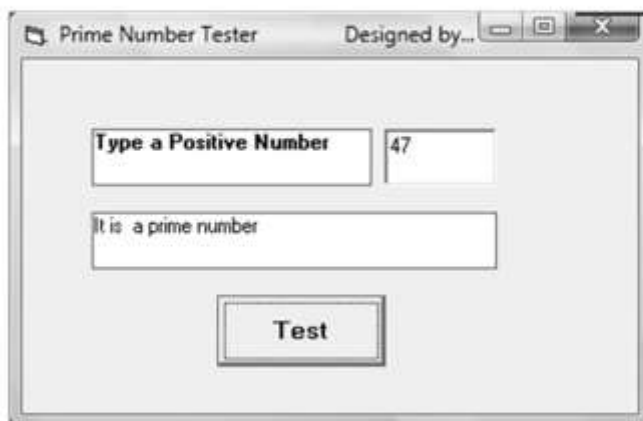
1. Logical और run time error को ढुंढना
2. कोड के संरचना को जाँचना, साधारणतः जब कोई प्रोग्राम run करते हैं, तब हमें उसका आउटपुट सबसे अंत में मिलता है, यदि वह उत्तर गलत है, तब हमें यह पता नहीं चल सकता है, कि कहाँ गलती हुई है। लेकिन debug से हम कहाँ गलत हो रहा है, यह देख सकते हैं।

डिबगिंग प्रोसेस के समय मुख्य काम यह होता है की क्यों, कोई एप्लिकेशन कुछ परिस्थितियों में सही नतीजे देती है, लेकिन दूसरी परिस्थितियों में गलत परिणाम देते हैं।

वी.बी. डॉट नेट में बहुत से ऐसे टूल हैं, जो सरल सिन्टेक्स ऐरर से लेकर जटिल ऐरर को सुलझाने के लिए प्रयोग होते हैं। डिबगिंग टूल, प्रोग्राम के क्रियान्वन को बीच में रोकने, प्रोग्राम में प्रयोग होने वाले वेरिएबल के मान (value) का परीक्षण करने, और एकजीक्युशन को प्रोसीजर के दूसरे स्थान से प्रारम्भ होने के लिए आदि कामों को निदेशित करता है। मुख्य रूप से वी.बी. डॉट नेट एप्लिकेशन की डीबगिंग में निम्न कार्य किये जा सकते हैं।

1. Execute procedure :- वी.बी. डॉट नेट में प्रोग्राम को कोई भी प्रोसीजर निर्माण के समय भी क्रियान्वित (run) किया जा सकता है। इसलिए किसी एक subroutine या फक्शन के लिए संपूर्ण एप्लिकेशन क्रियान्वित करना आवश्यक नहीं है।

2. **Suspending execution** :- break point के द्वारा प्रोग्राम का एक्जीक्युशन को बीच में रोक कर किसी भी स्थान पर एप्लिकेशन में कोड या वेरिएबल में परिवर्तन किया जा सकता है।
3. **Single stepping through code** :- इस सुविधा से कोड में परिवर्तन कर सकते हैं, तथा परिपक्व वेरिएबल का मान देख सकते हैं या वेरिएबल के मान में स्वयं परिवर्तन कर उसका प्रभाव प्रोग्राम के एक्जीक्युशन में देख सकते हैं।
4. **Examine variable** :- watches या watch point का प्रयोग कर वेरिएबल की स्थिति की जाँच की जा सकती है। इस विंडो में वेरिएबल के नाम एवं उसका current मान दिखाई देता है। जैसे आप "step into" बटन दबाकर आगे जाते हैं, वैसे वैसे प्रोग्राम के कंडीशन के अनुसार उसके मान उस watch विंडो में परिवर्तित हो कर दिखते हैं।
5. **Redirecting program flow**:- डिबर्गिंग टूल यह विकल्प भी होता है, की कोड की किसी भी विशेष लाइन को तुरन्त एक्जीक्युशन किया जा सकता है।



```
Private Sub  
Command1_Click()  
Dim N, D As Single  
Dim tag As String  
N =  
Val(TxtNumber.Text)  
Select Case N  
Case Is < 2
```

```
Lbl_Answer.Caption = "It is not a prime number"
```

```
Case Is = 2
Lbl_Answer.Caption = "It is a prime number"
Case Is > 2
D = 2
Do
If N / D = Int(N / D) Then
Lbl_Answer.Caption = "It is not a prime number"
tag = "Not Prime"
Exit Do
End If
D = D + 1
Loop While D <= N - 1
If tag <> "Not Prime" Then
Lbl_Answer.Caption = "It is a prime number"
End If
End Select
End Sub
```

ABC Transport Co.

### Extra Charge Slip

<input checked="" type="checkbox"/> Box Packing	Charge	30
<input checked="" type="checkbox"/> Box Insurance	Charge	100
<input checked="" type="checkbox"/> Home Delivery	Charge	100
<input checked="" type="checkbox"/> Delicate Item	Charge	300

Transport By

Surface (Normal Mode)  
 By Air  
 Air Conditional Vehicle

Get total Charge: 530      Flight Charges: 600

Next

Public Class Form1  
Public a, ins,  
homdel, delic, tot,  
fr As Integer

```
Private Sub CheckBox1_CheckedChanged(ByVal sender  
As System.Object, ByVal e As System.EventArgs) Handles  
CheckBox1.CheckedChanged
```

```
    If CheckBox1.Checked Then
```

```
        a = 30
```

```
        TextBox1.Text = 30
```

```
    End If
```

```
    If CheckBox1.Checked = False Then
```

```
        a = 0
```

```
        TextBox1.Text = a
```

```
    End If
```

```
End Sub
```

```
Private Sub CheckBox2_CheckedChanged(ByVal sender  
As System.Object, ByVal e As System.EventArgs) Handles  
CheckBox2.CheckedChanged
```

```
    If CheckBox2.Checked Then
```

```
        ins = 100
```

```
        TextBox2.Text = ins
```

```
    End If
```

```
    If CheckBox2.Checked = False Then
```

```
        ins = 0
```

```
        TextBox2.Text = ins
```

```
    End If
```

```
End Sub
```

```
Private Sub CheckBox3_CheckedChanged(ByVal sender  
As System.Object, ByVal e As System.EventArgs) Handles  
CheckBox3.CheckedChanged
```

```
    If CheckBox3.Checked Then
```

```
        homdel = 100
```

```
        TextBox3.Text = homdel
```

```
    End If
```

```
    If CheckBox3.Checked = False Then
```

```
        homdel = 0
```

```
        TextBox3.Text = homdel
```

```
    End If
```

```
End Sub
```

```
Private Sub CheckBox4_CheckedChanged(ByVal sender  
As System.Object, ByVal e As System.EventArgs) Handles  
CheckBox4.CheckedChanged
```

```
    If CheckBox4.Checked Then
```

```
        delic = 300
```

```
        TextBox4.Text = delic
```

```
    End If
```

```
    If CheckBox4.Checked = False Then
```

```
        delic = 0
```

```
        TextBox4.Text = delic
```

```
    End If
```

```
End Sub
```



```
Private Sub Button1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button1.Click
```

```
    tot = ins + delic + homdel + a
```

```
    TextBox5.Text = tot
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button3.Click
```

```
    fr = 300
```

```
    If RadioButton1.Checked Then
```

```
        TextBox6.Text = fr
```

```
    End If
```

```
    If RadioButton2.Checked Then
```

```
        fr = fr * 2
```

```
        TextBox6.Text = fr
```

```
    End If
```

```
    If RadioButton3.Checked Then
```

```
        fr = fr * 3
```

```
        TextBox6.Text = fr
```

```
    End If
```

```
End Sub
```

```
End Class
```

BCST

# कनक कम्प्यूटर एजुकेशन

माखनलाल चतुर्वेदी वि.वि. से संबद्ध

## DCA / PGDCA

माखनलाल चतुर्वेदी विश्वविद्यालय भारत का एक प्रमुख विश्वविद्यालय है जो की पत्रकारिता तथा कम्प्यूटर शिक्षा के क्षेत्र में कार्य करता है। इसकी स्थापना मध्यप्रदेश शासन के द्वारा 1990 में की गयी।

हमारी संस्था का उद्देश्य बेहतर तकनीकी ज्ञान रखने वाले विद्यार्थियों को तैयार करना है, जिससे की वे अपने भविष्य को अच्छा बनाने के साथ साथ सकारात्मक सोच रखते हुए देश के विकास में भी भरपूर सहयोग कर सके।



एक कदम उत्कृष्टता की ओर....



श्री फ्लेक्स मुलताई