

RSIT कम्प्यूटर कॉलेज करकेली

शिव मंदिर के पास, रेलवे स्टेशन रोड़ करकेली
जिला- उमरिया (म.प्र.) 484661



Subject – VB.NET

Session 2018-19

- **PGDCA**
- **DCA**
- **DCPA**
- **ADCP**
- **MCIT**

एवं अन्य 8 वीं पास सर्टिफिकेट कोर्सेस

Prepared by: -

Prakash Dwivedi (BE) 8982505087

Abhilash Pathak (MCA) 8517906324, 7974259812

Website: www.rudrasoftech.in

Microsoft Visual Studio (VB.Net 2003)

Unit-I

माइक्रोसॉफ्ट विसुअल स्टूडियो इंटीग्रेटेड डेवलपमेंट एनवायरनमेंट (IDE) है | जो की माइक्रोसॉफ्ट के द्वारा डेवलपमेंट की गई है | Languages के लिए डेवलपमेंट एनवायरनमेंट प्रोवाइड करता है |

वर्क प्रोग्राम डेवलपमेंट करने के लिए सभी आवश्यक टूल प्रोवाइड करता है | इसका प्रयोग कंसोल और ग्राफिकल यूजर इंटरफ़ेस (GUI) एप्लीकेशन बनाने के लिए किया जाता है | इसके द्वारा विंडो फॉर्म एप्लीकेशन , वेब एप्लीकेशन , websites , कंसोल एप्लीकेशन etc. create किये जा सकते हैं |

यह .NET के सभी लैंग्वेज के लिए एक कॉमन IDE प्रोवाइड करता है | जिसमें इसकी सभी प्रकार के प्लेटफॉर्म जैसे :- माइक्रोसॉफ्ट विंडो मोबाइल, विंडो कॉम्पोनेन्ट फ्रेमवर्क, .NET फ्रेमवर्क, या माइक्रोसॉफ्ट सिल्वर लाइट के एप्लीकेशन develop करने में सहायता करता है | यह .NET की महत्वपूर्ण लैंग्वेज जैसे VB.NET C# (CSharp), ASP.Net(Active server pages), J# etc. के लिए प्रोग्राम बनाने के लिए सभी सुविधाएं उपलब्ध करता है |

MSVS Visual Basic के पुराने प्रोग्राम के लिए नए फीचर प्रोवाइड करता है | और साथ ही यह सभी फीचर other लैंग्वेज के लिए भी प्रोवाइड करता है | इसमें VB.NET प्रोग्राम्स के लिए नए फीचर जैसे नए कम्पाइलर लैंग्वेज , न्यू डिबगिंग एक्सपीरियंस etc. फीचर सपोर्ट करता है | इसमें intellisense filtering form updates debugging data types etc. feature जोड़े गए हैं |

Visual Studio का code editor intellisense, Auto Complete, Auto List of Member , Auto Syntax Check, Auto quick information, जैसे कई फीचर सपोर्ट करता है |

इसका IDE Designer अलग अलग GUI Application के according design प्रोवाइड करता है जैसे window form designer, website के लिए web designer , data based के लिए data base designer etc.

Introduction Of Visual Basic.Net:

Microsoft .NET framework एक software framework है जो Microsoft Windows operating System पर use किया जाता है | यह एक बड़ी library है जो की common programming problems को program execution से .net framework के through जोड़ती है | .NET framework इन programs के execution को control करता है | यह नये applications जो की Microsoft में पहले से use किए जा रहे हैं को बनाने और manage करने में help provide करता है | .NET framework की Base Class Library (BCL) user Interface, data access , connectivity , web application development , numeric algorithms, network communication etc. facilities provide करता है |

.NET program के execution Environment और runtime requirement को manage करता है। जिसे की Common Language Runtime (CLR) कहते हैं। CLR program के execution को manage कर run करने में help करता है। इसके अलावा यह security, memory management, exception handling को भी manage करता है।

Feature of .NET:

- 1. Interpretability: computer में नये और पुराने applications के बीच interaction की आवश्यकता होती है। .NET framework इस interaction की facility को solve करने की facility प्रोवाइड करता है।

- 2. Common Runtime Engine (CRE): Common Runtime Engine CLR (common language Runtime) की सहायता से programs को run करने की facility provide करता है। .NET में बनाए गए सभी programs CLR के under में run होते हैं। CLR की मदद से program के execution की सभी requirements को पूरा किया जाता है। जैसे - memory management, security, exception handling etc.। .NET में सभी programs compile होने के बाद MSIL (Microsoft Intermediate language) में convert हो जाते हैं।

- 3. Language Independence: .NET framework में एक common type system (CTS) होता है जो की सभी possible data types, CLR और other instructions related work को support करता है। ये data types के बीच conversions, languages के बीच conversion etc. के support करता है।

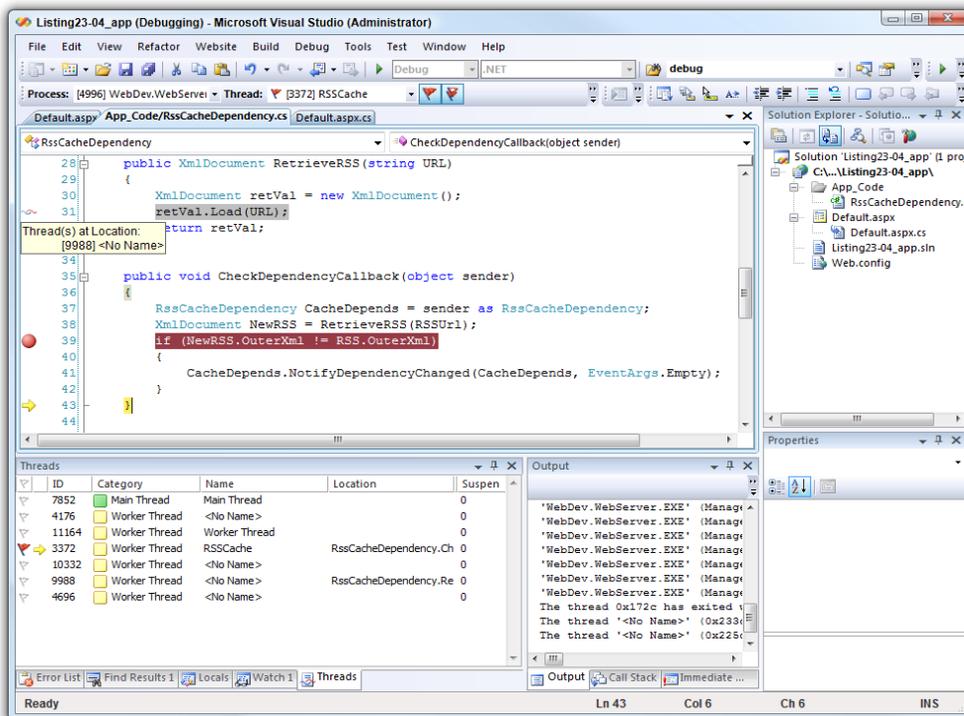
- 4. Base Class Library: BCL .Net framework का एक महत्वपूर्ण हिस्सा है। जिसका use .Net framework support करने वाली सभी language करती है। BCL में बहुत प्रकार की classes, functions, sub procedures provide किए गए हैं। जिनका use file को read, write और कई प्रकार के operations perform करने, Graphical User interface, data base conversion, XML Documentation etc. के लिए किया जाता है।

- 5. Simplified Deployment: .Net framework ऐसे features को support करता है जिसकी help से किसी भी computer software को distribute करना आसान हो जाता है। .Net में किसी भी software application के setup program को आसानी से design और distribute किया जा सकता है।

- 6. Security: .Net framework एक common security model को support करता है जिससे सभी applications secure रहें। यहाँ data को secure रखने में मदद करता है।

- 7. Portability: .Net में बनाए गए programs किसी भी windows operating system में run किए जा सकते हैं।

Integrated Development Environment (IDE)



IDE: IDE एक development environment है जो की किसी भी language में programs develop करने में सहायता करता है। यह program को बनाने और run करने और उससे related सभी features provide करता है। जैसे C या C++ के window उनके programs develop करने और उन्हें run करने में सहायता करती है। जिस तरह Photoshop की window photo editing tool provide करता है, MS word की window Documents related tools provide करता है उसी प्रकार Language की IDE program से जुड़े

सभी tools provide करती है ।

Visual Basic IDE (Visual Studio):

Visual Studio .Net के सभी language के लिए program बनाने में help करता है। Visual Studio में visual basic और अन्य कम्प्युटर भाषाओं के लिए programs लिखे जा सकते हैं। Visual Basic का प्रयोग windows Operating system के लिए applications बनाने में किया जाती है इसीलिए इसके IDE भी windows के सभी tools प्रयोग करती है जैसे menu, toolbars, windows etc. Visual Basic की IDE निम्नलिखित components provide करती

1. Menu Bar
2. Toolbar
3. Solution Explorer
4. Toolbox
5. Form Designer
6. Properties Window
7. Code Window
8. Object Browser
9. Output Window

Menu Bar:

Visual Basic IDE निम्नलिखित menu provide करता है।

I. File: File menu नया project बनाने , project को save और open करने , project add करने etc. के लिए सुविधा प्रदान करता है।

II. Edit: यह basic editing tools provide करता है। जैसे cut, copy, paste, find, replace, undo, redo, delete etc.

III. View: यह menu IDE की windows को display करने की commands contain करता है। जैसे solution explorer, designer, code window etc.

Guided by: Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)

- IV. Project: यह menu project में नए items add करने की command देता है।
- V. Build: इसमें project को build करने के लिए command होती हैं।
- VI. Debug: इसमें debugging के लिए command होती है।
- VII. Data: यह menu database से जुड़ी command contain करता है।
- VIII. Tool: यह .Net के extra tools contain करता है। जैसे connect to server, connect to device etc.
- IX. Window: यह window arrangement command contain करता है जैसे tile, cascade etc.
- X. Help: यह help से जुड़ी कमांड contain करता है। इसमें online और offline help की सुविधा होती है।

- **Toolbar:** toolbars menus में प्रोवाइड की commands के shortcut को दिखाते हैं। Visual Studio IDE में कई प्रकार के toolbars पाये जाते हैं। जो की अलग अलग प्रकार के कार्यों को करने के लिए प्रयोग किए जाते हैं। इसमें standard toolbar project create करने, save, open, editing commands (cut, copy, paste etc.), program execute करने के लिए कमांड रखता है। इसके अलावा इसमें special toolbars भी होते हैं जैसे debug, formatting, build etc.

- **Solution Explorer:** यह IDE का एक महत्वपूर्ण हिस्सा है। यह .Net में create किए गए किसी भी project में प्रयोग की गई सभी files की list को दिखाता है। Visual Studio में बनाए गए project एक solution का हिस्सा होते हैं। जिस प्रकार प्रोजेक्ट program में प्रयोग की गई सभी files का collection है उसी प्रकार से solution सभी projects का collection होता है। solution explorer में solution में प्रयोग की गई सभी files की list को explorer करता है इसीलिए इसे solution explorer कहते हैं। यह windows explorer की तरह ही होता है। यह project की सभी files को आसानी से access करने की सुविधा देता है।

Solution Explorer में सबसे ऊपर एक toolbar होता है जिसमें कुछ buttons display होती हैं इसके बाद इसमें solution, project और उसकी files hierarchical रूप में दिखाई देती हैं। इसके toolbar में मुख्यतया 6 buttons होती हैं - properties, show all files, refresh, class diagram, view designer और view code. यह buttons selected object के अनुसार बदल जाती है। इसमें इसके बाद solution का नाम दिखाई देता है। फिर project और फिर उससे related files दिखाई देती है। solution Explorer इन सभी files को arrange और access करने में मदद करता है। Solution Explorer में दिखने वाली किसी भी फ़ाइल को select कर उसे open किया जा सकता है। इसमें दिखने वाली file पर right click करने पर दिखने वाले context menu में file से जुड़े कई options दिखाई देते हैं। जैसे - View design, view code, cut, copy, paste, remove, delete, load, unload, set as startup, properties, refresh etc.

इसे view menu में जाकर Solution Explorer command या Ctrl + w + s key press कर open किया जा सकता है।

- **Toolbox:** Toolbox Visual Studio (VB) IDE का एक महत्वपूर्ण हिस्सा है। यह user interface design करने में मदद करता है। यह window Tools जिन्हे windows control कहते हैं उन्हें दिखाता है। यह drag & drop method को support कर windows controls को windows form में प्रयोग करने की सुविधा प्रदान करता है। Toolbox में windows में प्रयोग होने वाले सभी प्रकार के controls उपस्थित होते हैं।

Toolbox को दिखने के लिए View menu में Toolbox या Ctrl + w + x key के द्वारा भी दिखाया जा सकता है। Toolbox में controls की संख्या ज्यादा होने के कारण इसे category में बांटा गया है। सभी categories अलग अलग प्रकार के controls को contain करती है। user इन categories पर click कर उन्हें show या hide कर सकता है। Toolbox में display होने वाले controls पर double click या drag कर form को design किया जा सकता है। इसमें controls के नाम और उनका icon दोनों दिखाई देते हैं। कुछ windows controls निम्न लिखित हैं। -

- Common Windows Controls: Button, Textbox, Label, checkbox, combo box, list box, link label, picture box, radio button etc.
- Data: dataset, data grid, Table adapter etc.
- Containers: group box, panel etc.
- Menu & toolbars: Menu, Toolbar, Status Bar etc.

• Form Designer: Visual Studio में Designers का महत्वपूर्ण स्थान है। यह designers GUI बनाने में मदद करते हैं। Visual Studio में कई प्रकार के designers पाये जाते हैं जैसे windows form designer, web form designer, database designer etc. Windows form Designer Windows Form को design करने में मदद करता है। इसका प्रयोग windows के लिए User Interface बनाने के लिए किया जाता है। इसमें एक form दिखाई देता है जिसमें के form के सभी हिस्से दिखाई देते हैं जैसे title bar, control box, icon etc. इससे form को resize करके उसे बदला जा सकता है। यह drag & drop की सहायता से controls के add किया जा सकता है और उन्हें move कर arrange किया जा सकता है। इसमें controls को बिना प्रयोग किए उनके लिए code नहीं लिखे जा सकते हैं। windows form को दिखने के लिए view menu , solution explorer या Shift + F7 key का प्रयोग करते हैं।

• Properties Window: Visual Studio में Properties window का अधिकतर प्रयोग किया जाता है। यह VS IDE का एक main part है। इसका प्रयोग Visual Studio और projects में प्रयोग किए गए किसी भी object के properties को दिखने और उन्हें बदलने के लिए किया जाता है। इसमें windows form , उसमें प्रयोग किए गए controls, project files, database files और सभी प्रकार की files की property दिखाई देती है। properties window को view menu या Ctrl + w + p key प्रैस कर display किया जा सकता है। यह विभिन्न properties के लिए अलग अलग प्रकार की editing fields display करता है।

इसमें सबसे ऊपर एक Combo Box होता है जिसमें की किसी भी select किए गए object का नाम और उसका Type दिखाई देता है। इसके बाद इसके toolbar में button होती है जो निम्न है।

- Categorized: यह button properties को categories में दिखने के लिए प्रयोग होती है।
- Alphabetical: यह properties के alphabetical order में दिखने के लिए प्रयोग की जाती है।
- Properties: यह properties की list दिखने के लिए प्रयोग की जाती है।
- Event: यह event की list दिखने के लिए प्रयोग की जाती है।
- Properties Page: यह properties के page को दिखने के लिए प्रयोग होती है।

इन buttons के बाद properties window को 3 भागों में बांटा गया है। जो निम्न हैं।

- Property Pan: इसमें properties की list दिखाई देती है।
- Value Pan: इसमें properties की value दिखाई देती है जिसे की बदला जा सकता है।

□ Description Pan: इसमें select की गई property का short description होता है।

• **Code Window:** किसी भी programming language में coding सबसे महत्वपूर्ण हिस्सा है। किसी भी language में बिना codes लिखे programs नहीं बनाए जा सकते हैं। Visual Studio में यह सुविधा code window उपलब्ध करवाती है। Visual Studio code window में .Net के लिए सभी languages के लिए codes लिखे जा सकते हैं। इसकी code window program के अनुसार बदलती रहती है। Visual Basic की code window सभी प्रकार के Visual Basic features को support करती है। code window को display करने के लिए view menu में code, solution explorer की view code या context menu, form के context menu में view code select करते हैं।

Visual Basic की code window में 2 combo box (Drop Down list) और एक text editor होता है। इसमें पहला combo box form और उसमें प्रयोग किए गए controls के नाम को दिखाता है जिसमें controls को select किया जा सकता है। दूसरे combo box में select किए गए control की events दिखाई देती हैं जिन्हें select कर text editor में codes लिखे जा सकते हैं। Visual Basic का code editor Visual Basic के सभी coding features को support करता है। जैसे - auto complete, auto list of members, auto syntax check, auto quick information, code refactoring, rapid application development (less coding) etc.

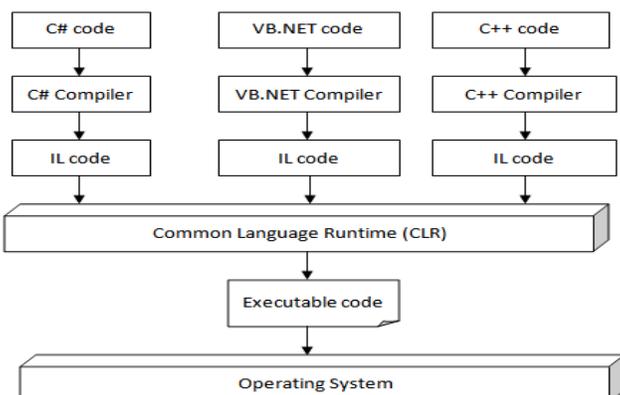
• **Object Browser:** Visual Studio में object browser का महत्वपूर्ण स्थान है। यह .Net में प्रयोग होने वाले components जैसे classes, namespaces, enumerators, interfaces etc. को दिखाता है। object browser को .Net classes की properties, events और methods को देखने के लिए इस्तेमाल किया जाता है। इसे IDE में दिखाने के लिए View menu में Object Browser या Ctrl + w + j shortcut key का प्रयोग किया जाता है।

इसमें सबसे ऊपर objects के search करने की सुविधा होती है। इसे मुख्यतः तीन भागों में बांटा गया है।

□ □ Object Pan: इस पैन में .Net के namespaces, interfaces और classes की hierarchical list होती है जिसे explorer किया जा सकता है। इसे explorer करने पर classes की list प्राप्त होती है।

□ Member Pan: इस pan में Object pan में चुनी गई class की members की list दिखाई देती है। जिसमें मुख्यतः properties, events और methods होती हैं जिन्हें select किया जा सकता है।

□ Description Pan: इस pan में Member और Object pan में select किए गए object या उसके member का description होता है।



□ **Output Window:** output window भी VS IDE की एक window है जो कि programs को compile और run करने पर generate होने वाले messages को display करती है। इसमें program को build और run होने पर उत्पन्न होने वाले message होते हैं। इसे IDE में show करने के लिए view menu में Output window या ctrl+ w+ o key का प्रयोग करते हैं। इसमें एक

toolbar होता है जिसमें message के देखने के लिए tools होते हैं जैसे search, next, previous etc. इसके बाद एक box में सभी message display होते हैं।

Architecture of .Net:

- Common Language Infrastructure (CLI) : CLI का main purpose एक language neutral platform, application development के लिए provide करना है। इसके अलावा यह exception handling , garbage collection , security और interpretability के feature को भी include करता है। इन features का प्रयोग करके किसी भी language में programs को develop किया जा सकता है। . Net framework CLI का use करके इस सभी feature के provide करता है। CLI में MSIL , CLR और Machine code सभी include होते हैं।

- Common Language Runtime: CLR .Net में लिखे गए प्रोग्राम के execution environment को define करता है। यह .Net framework का important part है। जो के .Net में लिखे गए program को execute करने में use होता है। CLR program के compile होने के बाद बनाने वाले Byte Code जिसे की Microsoft Intermediate language भी कहते हैं उसे run करता है। Programmer Visual basic , C# या किसी भी अन्य language के use कर program लिखता है जिसे ही उस language का compiler Byte code में Convert कर देता है। यह byte codes सभी languages के लिए एक समान ही होते हैं। इसके बाद .Net CLR इस byte code के machine code (Native Code) में बदल देता है। जो की execute हो जाते हैं।

इसके अलावा CLR run होते वाले प्रोग्राम के execution requirements के भी पूरा करता है। साथ ही यह user को program run करने के लिए CPU specification से free कर देता है। यह निम्नलिखित facilities provide करता है।

- Memory management
- Thread management
- Exception handling
- Garbage collection
- Security
- Common Type System:

.Net Framework में Common Type System Computer memory में represent होने वाले type System और उनकी values को specify करता है। यह अलग अलग language में लिखे गए programs के बीच information को share करने में मदद करता है। यह cross language platform के support करता है जिसके कारण यह different language में लिखे गए programmes के बीच information share करने में मदद करता है। यह एक object oriented model को support कर इन्हे पूरी तरह से implement करता है। इसके प्रयोग विभिन्न languages के objects को एक ही platform में share कर देता है। जैसे C# के programs के Visual Basic के programs के साथ प्रयोग किया जा सकता है। यह objects के बीच information share करने के नियमों को define करता है। यह विभिन्न language के data types के बीच interaction provide करता है। इस प्रकार CTS .Net में important role निभाता है।

- Microsoft Intermediate Language (MSIL), Common Intermediate Language (CIL), Byte Code:MSIL को Common Intermediate Language भी कहते हैं। यह एक low level human readable

Programming Language है। जो के .NET के Common Language Infrastructure में available होती है। यह language .NET support करने वाली language के compile होने पर generate होती है। इस language को CLR के द्वारा run किया जाता है। इसे byte codes भी कहते हैं। यह object oriented assembly language होती है जो की CLR के द्वारा virtual machine पर execute हो जाते हैं।

.Net में use होने वाली सभी language compile होने पर language specific compiler के द्वारा MSIL (CIL) में convert हो जाते हैं। यह direct machine code में नहीं बदलती हैं। CIL एक CPU और platform independent code होते हैं। इसके बाद यह byte codes .Net CLR के द्वारा machine code में convert होकर program को execute कर देते हैं। MSIL codes किसी भी environment जो की .Net framework को support करता है उस पर execute किए जा सकते हैं।

- Executing MSIL:

- Source code Language specific compiler के द्वारा Common Intermediate language में convert हो जाते हैं।

- CIL के according .Net assembly बना देता है।
- यह byte codes CLR में execute होने के लिए चले जाते हैं।
- CLR इन्हें machine code में convert कर execute कर देता है।
- Assemblies and Class Libraries:

- Assemblies: assemblies .Net framework के महत्वपूर्ण हिस्सा हैं। इस program के run होने के लिए कई प्रकार की information contain करता है। यह निम्न works के perform करता है।

यह common language runtime के द्वारा execute होने वाले कोड्स को contain करता है। Microsoft Intermediate language के code बिना assembly के execute नहीं किए जा सकते हैं। यह एक associated assembly बना देता है। यह program की security limitations के define करता है। यह प्रोग्राम को execute करने के लिए request और permission को process करता है। यह type limitations के बनाता है। सभी identify किए गए Types assembly में store हो जाते हैं। यह version limitation को define करता है। यह deployment unit को create करता है। यह program के side by side execution को support करता है।

Class Library: Microsoft base class library .Net framework का सबसे important part है। यह एक library है जो के .Net की सभी कम्प्यूटर languages के द्वारा use की जा सकती है। यह सभी के लिए common library provide करता है। इस library में बहुत से common functions, methods और classes होती हैं जो सभी .Net languages के लिए common होती हैं। यह functions और methods विभिन्न कार्यों के लिए use की जाती हैं जैसे: file reading and writing, database interaction, XML documentation etc.

कुछ important .Net Library classes और namespaces निम्नलिखित हैं।

System: यह programming के basic आवश्यकताओं को पूरा करने के लिए use होती है। यह data types और conversion functions contain करती है जैसे: string, Boolean, datetime etc.

System. Collection: यह सभी common containers और collections के define करती है। जैसे: stack, list, queue etc.

System. IO: यह programmer को Read और writing functions provide करती है। जो सभी प्रकार के Input और Output के लिए प्रयोग की जाती है।

System. Configuration: यह configuration data को handle करने के लिए functions और methods provide करती है।

System. Data: यह database related functions और methods प्रोवाइड करती है। जिसे के ADO.NET भी कहते हैं।

System. Drawing: यह graphical designing functions और methods के लिए प्रयोग की जाती है। यह विभिन्न methods जैसे DrawRectangle , DrawCircle , Pen , Brush etc. provide करता है।

System. Windows: यह Class सभी windows operating System related functions और methods के लिए प्रयोग की जाती है।

System. Windows. Forms: यह windows form के सभी controls , methods और functions के लिए प्रयोग की जाती है।

Event Driven Programming:-

VB.net में किसी भी प्रोग्राम या एप्लीकेशन को कम समय में develop किया जा सकता है | vb.net में एप्लीकेशन , form or active X Controls basic element होते हैं | जो यूजर और एप्लीकेशन के बीच interfacing का कार्य करते हैं। इन सभी एलिमेंट के Method और Properties होती है | और इस object के ऊपर events को भी Apply किया जाता है जिससे वह react करता है और किसी न किसी code को execute करते हैं।

जैसे किसी form व active X Controls के ऊपर Click करके codes को Execute किया जाता है यहाँ पर क्लिक एक event है | और इस प्रकार की facility प्रोवाइड करने वाली लैंग्वेज Event Driven Programming लैंग्वेज कहलाती है।

उदा: यदि एक कार को एक object मान लिया जाए तो color उसकी प्रॉपर्टी होगी इस प्रॉपर्टी को कार की design करते समय Change किया जाता है | किसी प्रकार के vb.net में controls की प्रॉपर्टी design time में change किया जा सकता है जिस प्रॉपर्टी को design करते समय change किया जाता है

उसे design time प्रॉपर्टी और जिन प्रॉपर्टी को run time में change किया जाता है | उसे run time प्रॉपर्टी कहते हैं | इन् सभी प्रॉपर्टी को vb.net में event के द्वारा ही execute किया जा सकता है इसलिए vb.net को Event Driven Programming कहते हैं |

Method:-

जिस प्रकार का प्रत्येक object control एवं form की प्रॉपर्टी होती है उसी प्रकार से उनकी कुछ method भी होती है जिन्हें हम commands कहते हैं method के द्वारा form या control के ऊपर स्पेशल commands apply किया जाता है जैसे form को show या hide करना, Listbox में item को add या delete करना command dialog के control को open करना etc.

उदा:-show,hide,listbox.item.add,timer.stop/start.

Event:- जब हम किसी object के ऊपर external activitie करते हैं। तो उस object के द्वारा उसमें लिखा code execute किये जाते हैं। इसी प्रकार के reaction को event कहते हैं। इस event में programming code का use करके किसी भी प्रकार का वर्क कर सकते हैं।

Event दो प्रकार के होते हैं।

1. Mouse 2. Keyboard

Mouse event :- System में जो event mouse के द्वारा activate किये जाते हैं। वो mouse event कहलाते हैं।

Keyboard event:- ऐसे event जो keyboard event के द्वारा activate किये जाते हैं उसे keyboard event कहते हैं

जैसे :- alt, Ctrl, delete, Enter Space, Tab.

उदा :- Private Sub Button1_Click()

End Sub

Private Sub Form1_Load ()

End Sub

यह एक बटन का Click event है

यह एक form का load event है

Environment:-

Visual Basic .net में हमें programming करने के लिए प्लेटफार्म (GUI Based) या (Window Based) मिलता है। अर्थात इसमें mouse का प्रयोग करते हुए कार्य कर सकते हैं और हमें set of command को use करने के लिए help प्राप्त होती है इसके अलावा हम Toolbox में मौजूद अलग अलग tool का use करके भी अपना कार्य कर सकते हैं

Visual Basic .net में Intellisense नाम की एक सुविधा प्रदान की गई है इसके द्वारा कोई भी method या keyword लिखते समय Automatic उससे related सारा detail show होता है। जैसे आपका syntax क्या है। उसमें कौन कौन से argument pass करने हैं।

Types of Environments

1. Editor Tab:- editor tab code window तथा project window की setting का विवरण करता है।

2. जब option dialog बॉक्स में प्रदर्शित होता है तो by default editor tab के option show होते हैं। editor tab के अन्दर editor tab code setting तथा project window के अन्दर निम्नलिखित tab option होते हैं

(a) Code Setting के tab option :-

(i) Auto syntax Check :- Visual Basic .net में जब हम कोडिंग करते हैं तो यदि हमने कोडिंग की एक लाइन को भी टाइप किया है। और इसमें गलती हुई है। तो यह स्वतः ही सही हो जाता है।

(ii) Require variable declaration:- यदि module में explicit variable को परिभाषित करता है। इसे चुनने के बाद option explicit statement नए module में general declare होता है।

- (iii) Auto List Member:- यह option एक बॉक्स को प्रदर्शित करता है जो सूचनाओं को दिखता है। जो current insertion point पर statement को logically पूरा करता है
- (iv) Auto quick info :- यह option फंक्शन तथा उसके parameter के बारे में सूचना प्रदर्शित करता है ।
- (v) Auto data type:- :- यह option break mode में code विंडो में variable की value या object प्रॉपर्टी show करता है । जिस स्थान पर cursor होता है।
- (b) Window Setting के tab option :-
- (i) Drag and Drop text:- यह option code विंडो में current code के element को drag & drop करने की facility प्रोवाइड करता है।
- (ii) Default to full module view:- यह option नए module के लिए default state को allow करता है ।
- (iii) Procedure Separator :- यह option Separator Bars को प्रदर्शित करने या छुपाने के लिए allow करता है । जो code विंडो में प्रत्येक Procedure के अंत में प्राप्त होता है ।
- (c) Format tab :- यह tab vb.net code के experience को specify करता है । इनके विभिन्न tab option नि० लि० है ।
- (i) Text List :- यह option उन आइटम्स की लिस्ट की show करता है । जो customized color को रखते है ।
- (ii) Fore Ground:- यह option text के लिए fore ground को specify करता है। जिस text को color text के लिए चुना जाता है।
- (iii) Size:- यह option code में use किये जाने वाले code के size को specify करता है ।
- (iv) Sample:- यह option code में use किये जाने वाले font size तथा color के setting के sample को show करता है।
- (d) General Tab:- General Tab current में प्रयोग किये जाने वाले project के लिए setting को specify करता है। तथा इसके साथ error handling तथा compile की setting को भी specify करता है।
- (i) Form grid setting :- यह option form के design के समय form के grid की Representation को show करता है।
- (ii) Show tool tips:- यह option टूलबार तथा आइटम्स के लिए tooltip को show करता है
- (iii) Collapse Project Hide window:- यह option project explorer में जब एक project को collapse किया जाता है या विंडो को hide किया जाता है। specify करता है ।
- (iv) Error Trapping:- यह option सुनिश्चित करता है की vb.net development environment में जैसे गलतियों को handle किया जाता है।

(v) Compile :- यह option सुनिश्चित करता है की vb.net में कैसे project को compile किया जाता है।

(e) Docking Tab:- इस tab से हमें यह देता है की हम कौन कौन सी window को dockable करना चाहते है | जब हम dockable window को move करते है | docking tab के अंतर्गत dockable tab उन window की लिस्ट को show करता है | जो window dockable इसमें से हम उन window को चुनते है | जिन्हें हम dockable करना चाहते है | और उन्हें clear करते है। जिन्हें हम नहीं चाहते है।

Unit-II

Variables:

Computer programming में variables बहुत महत्वपूर्ण स्थान रखते हैं। variables, named locations जिनका प्रयोग किसी value को store करने के लिए किया जाता है। प्रत्येक variable का एक नाम और value होती है साथ ही इसका एक Type होता है। visual basic में कई प्रकार के Data Types होते हैं। जैसे - Integer, String, Date, Char, Decimal etc.

Declaring Variables:

Visual Basic.Net में variable को declare करने के लिए Dim statement का प्रयोग किया जाता है। जिसका Syntax निम्नलिखित हैं।

Dim <Variable Name> As <Data Type>

जैसे

Dim name As String

इस statement में Dim keyword variable का dimension, variable name के साथ पर variable का नाम, As keyword और data Type बताता है। Visual Basic.Net में साधारणतया variables इसी प्रकार से declare किए जाते हैं। इसके अलावा scope के अनुसार variable declaration statement बदल जाती है। public variable declare करने के लिए public keyword का प्रयोग Dim के स्थान पर किया जाता है।

Public <Variable Name> As <Data Type>

इसी प्रकार private और friend keyword भी variable declare करने के लिए प्रयोग किए जाते हैं। private, public, protected और friend variable के scope के लिए प्रयोग होते हैं। VB.Net में multiple variables को single statement में declare किया जा सकता है।

Dim a, b, c as Integer

इसी प्रकार इसमें multiple type के variables को भी single dim statement में declare किया जा सकता है।

Dim a As Integer, name As String, b As Boolean

Variable Naming Conventions:

- Variable का नाम हमेशा letter से शुरू होता है।
- Variable name हमेशा 255 character से ज्यादा नहीं होता चाहिए।
- Variable name में numbers और underscore को छोड़कर किसी अन्य character का प्रयोग नहीं किया जाता है।
- Variable name हमेशा अपने scope में unique होना चाहिए।

Guided by: *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

- Visual Basic के keywords को Variable name में use नहीं किया जा सकता है।
- Variable name में blank space नहीं होना चाहिये ।

Types of Variables:

Data Types:

Programming में data types Data का type define करते हैं। Data types store होने वाले data के type के साथ ही उसकी data storage length भी बताते हैं। Visual Basic में variables को निम्नलिखित पाँच categories में बांटा गया है।

- Numeric
- String
- Boolean
- Date
- Object

1. Numeric Variables: इसमें numbers store करने वाले data type आते हैं। इनका प्रयोग mathematical operations perform करने के लिए किया जाता है। इसमें निम्न data type's आते हैं-

□ Short (Int16): यह integer value को स्टोर करने लिए use होता है। इसमें -32768 से 32767 तक value store की जा सकती है और यह memory में 2 bytes लेती है।

```
Dim a As Short
a = 10000
```

□ Integer (Int32): यह integer value को स्टोर करने लिए use होता है। इसमें -2147483648 से 2147483647 तक value store की जा सकती है और यह memory में 4 bytes लेती है।

```
Dim a As Integer
a = 1000000
```

□ Long (Int64): यह integer value को स्टोर करने लिए use होता है। इसमें integer से बहुत अधिक value store की जा सकती है और यह memory में 8 bytes लेती है।

□ Single: यह single precision floating no को स्टोर करता है। इसमें -3.402823×10^{38} से $-1.401298 \times 10^{-45}$ और 1.401298×10^{-45} से 3.402823×10^{38} तक value store कर सकता है। यह 4 byte memory लेता है।

□ Double: यह double precision floating no को स्टोर करता है। इसमें $-1.797693134486232 \times 10^{308}$ से $-4.94065645841247 \times 10^{-324}$ और $4.94065645841247 \times 10^{-324}$ से $1.797693134486232 \times 10^{308}$ तक value store कर सकता है। यह 8 byte memory लेता है।

□ Decimal: इसमें 0 से 7.9228162514264337593543950335 positive और negative value store कर सकते हैं। यह memory में 16 Byte लेता है।

□ Byte: यह भी एक numeric data type है जो की 0 से 255 तक की value store की जाती है। यह memory में 2 byte लेता है।

2. String: यह characters को store करने के लिए प्रयोग होता है। इसमें निम्न 2 type होते हैं।

1. String: यह केवल text को store करने के लिए use किया जाता है। यह set of characters को स्टोर करता है। इसमें 2 GB तक text store कर सकते हैं।

2. Char : यह single character को store करने के लिए use होता है। और memory में 2 byte memory लेता है।

3. Boolean: इस data type का प्रयोग True या false value store करने के लिए किया जाता है। यह conditions check करने में अधिकतर प्रयोग होता है।

4. Date: यह data type date और time store करने के लिए प्रयोग किया जाता है। इसमें date या time को store करने के लिए # कैरक्टर का प्रयोग करते हैं।

```
Dim a As Date
a = #1/4/2005#
```

5. Object: यह एक common data type है जो की सभी data types के common features के साथ सभी प्रकार की value store करने के लिए use किया जाता है।

Forcing Variable Declaration:

Visual Basic 6 में variable को declare करना आवश्यक नहीं होता है। यह visual basic 6 की default setting के कारण होता है। उसी प्रकार Visual Basic में भी variable को बिना declare किए use किया जा सकता है। default setting के कारण VB.Net में variable को declare करना आवश्यक होता है। ऐसा करने पर VB.Net error generate कर देता है। variable को declare नहीं करने पर Visual basic इसे Object type का declare कर देती है। Visual Basic के इस default behavior को बदलने के लिए निम्नलिखित statement को code window में सबसे ऊपर लिखा दिया जाता है।

```
Option Explicit Off
```

इस statement को लिखने पर लिखे गए code module के लिए variable declaration करना compulsory नहीं होगा। इस setting को पूरे project में apply करने के लिए project की properties का प्रयोग किया जाता है। यहाँ पर इसे बदलने पर यह सेटिंग पूरे project में apply हो जाती है। Default setting में यह हमेशा on रहती है जिसके कारण variable को declare करना जरूरी होता है। Variable Declaration करने से program में error आने की संभावना कम होती है। variable declaration force करने के लिए Option Explicit को On कर दिया जाता है। या की निम्न statement को code विंडो में सबसे ऊपर लिखने पर variable declare करना compulsory हो जाता है।

```
Option Explicit On
```

Scope and Life Time of Variable:

Programming में variable का scope variable के use होने की limit बताता है। कोई भी variable कहाँ तक प्रयोग किया जा सकता है यह scope के द्वारा ही define होता है। इसे visibility भी कहते हैं। Visual Basic.Net में चार प्रकार के scope पाये जाते हैं।

1. Block Scope
2. Procedural Scope
3. Module Scope
4. Namespace Scope (Public Scope)

I. Block Scope: जब कोई variable किसी codes के block के अंदर declare किए जाते हैं तो उस variable को उस block के बाहर नहीं किया जा सकता है। इसमें निम्न statements होती हैं।

- a. Do and Loop
- b. For [Each] and Next

```

c.    If and End If
d.    Select and End Select
e.    Try and End Try
f.    While and End While
g.    With and End With
If n < 1291 Then
    Dim cube As Integer
    cube = n ^ 3
End If

```

II. Procedural Scope (Local Scope) : यदि कोई भी variable किसी भी procedure के अंदर declare किया जाता है तब इस स्कोप को procedural scope कहते हैं। इस scope में declare किए गए variable को केवल declare किए गए procedure में use किया जा सकता है। यह variable उस procedure के बाहर कहीं भी प्रयोग नहीं किया जा सकता है।

Program 1: Printing Sum of Even from 1 to 100

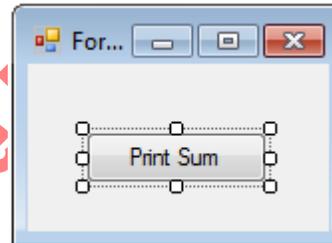
```
Public Class Form1
```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Dim i As Integer
    Dim sum As Integer
    For i = 0 To 100 Step 2
        sum = sum + i
    Next
    MsgBox("Sum of All Even No = " & sum)
End Sub
End Class

```

इस example में i और sum दोनों procedural scope



में हैं।

2. Module Scope: यह scope variable को पूरे code module में उसे करने की सुविधा देता है। इस scope में variable declare करने के लिए dim या private statement को code window में public class form के नीचे लिखते हैं। इस scope में declare किए गए variables को सभी procedures में उसे किया जा सकता है। जो उस code module में available हैं। इस scope को module scope कहते हैं।

Program 2: Module Scope

```

Public Class Form1
    Dim x As Integer
    Sub Exp1()
        Dim y As Integer
        x = 10
        y = 20
        MsgBox(y)
    End Sub

    Sub Exp2()
        Dim y As String
        y = "VB.NET"
        MsgBox(y)
        MsgBox(x)
    End Sub

    Sub Exp3()
        MsgBox(x)
    End Sub

```

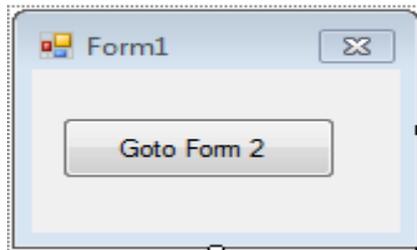
End Class

इस example में x module scope का variable है जिसके कारण इसे Exp1, Exp2 और Exp3 तीनों sub procedures में use किया गया है। जबकि y local scope का variable है।

3. Namespace Scope: यह scope variable को पूरे project में use करने की सुविधा देता है। इस scope में variable को declare करने के लिए public statement को code window में public class form के नीचे लिखा दिया जाता है। इसमें dim या private के स्थान पर Public का use किया जाता है। यहाँ पर declare किया गया variable project में कहीं भी use किया जा सकता है। इसे use करते समय code module का नाम लिखना पड़ता है। जैसे -

Program 3: Public Module- इस program में दो form use होंगे:

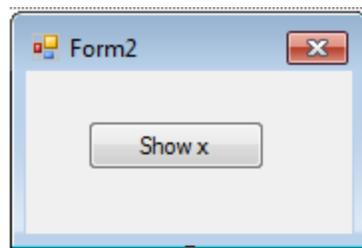
Form1.vb



```
Public Class Form1
    Public x As Integer
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        x = 100
        Form2.Show()
    End Sub
End Class
```

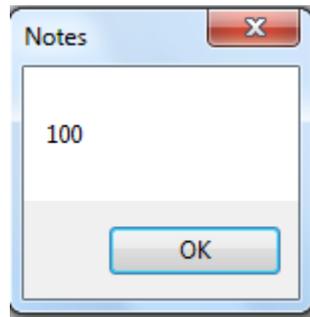
Form2.vb



```
Public Class Form2
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        MsgBox (Form1.x)
    End Sub
End Class
```

Output:



Constants:

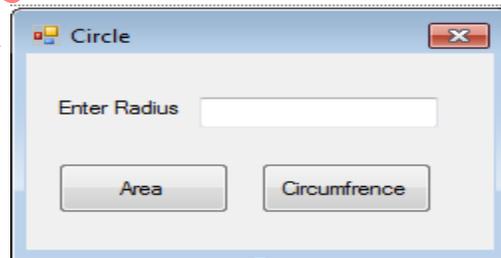
ऐसे variables जिनकी value run time में change नहीं होती है उसे constant कहते हैं। यह constants mathematical operations में ज़्यादातर use किए जाते हैं। इसमें store की गई value program के execute होने पर नहीं बदलती है। यह programming में कई बार आते हैं। जैसे किसी mathematical calculator बनाते समय $\pi = 3.14159$ ($\pi = 3.14159$) कई बार प्रयोग किया जाता है। ऐसे में इसे constant के रूप में declare कर दिया जाता है।

Circumference = $2 * \pi * \text{radius}$

Constants को मुख्यतः दो कारणों से use किया जाता है।

- Constants don't change value: constants की value run time में कभी नहीं बदलती है जिससे इसमें store की value में program का कोई प्रभाव नहीं पड़ता है।
- Constants are processed faster than variable: program के run होने पर constants normal variables से ज्यादा तेजी से execute होते हैं। जिससे program execution की speed बढ़ जाती है।

Program 4: Calculating area and Circumference of Circle:



```
Public Class Form1
```

```
    Const pi As Decimal = 3.14159
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim radius, area As Decimal
        radius = Val(TextBox1.Text)
        area = pi * radius * radius
        MsgBox(area)
    End Sub
```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim radius, cir As Decimal
    radius = Val(TextBox1.Text)
    cir = 2 * pi * radius
    MsgBox(cir)
End Sub
End Class

```

Arrays:-

array एक standard structure है जो की data को store करने के लिए use किया जाता है। individual variable केवल एक ही value को store करता है जबकि array data के set को store करने के लिए प्रयोग किया जाता है। इसमें single variable में एक से ज्यादा डाटा store किया जा सकता है। जैसे - a set of numbers, a series of dates etc.

Array is used to hold a set of data of the same type.

Array variable में value store करने और access करने के लिए array की index का use किया जाता है।

Declaring Array:

simple variable की तरह ही array को भी Dim, private या public statement का use कर declare किया जाता है। इसे declare करते समय array के नाम के बाद parentheses में array की last index की value दी जाती है। जिसे subscript, index या size कहते हैं

Syntax: Dim <array name > (subscript) As <Data Type>

Exp- Dim employee_name (20) As String
Dim salary (20) As Integer

Types of Array:-

Visual Basic में Array दो प्रकार के होते हैं।

- i. One Dimensional Array
- ii. Multi-Dimensional Array

- One Dimensional Array: ऐसे array जिसमें केवल एक index या subscript होती है उसे one dimensional array कहते हैं। इसमें multiple values को स्टोर किया जा सकता है। इसमें array में कोई value store करने या access करने के लिए इसकी index का use किया जाता है। जैसे-

Dim marks (10) As Integer

इस declaration में marks array में 0 से 9 index तक या 10 values store की जा सकती हैं।

Initializing one dimensional Array: 1D array में design और run time दोनों में values को insert किया जा सकता है।

Design time में value store करने के लिए array index या curly brackets {} का use किया जाता है।

Syntax:

```
Dim array_name() As DataType = {element1, element2,...}
```

Exp-

```
Dim name () As String = {"Raj", "Shyam", "Ashu"}
```

या

```
Dim Name (2) As String
```

```
Name (0) = "Raj"
```

```
Name (1) = "Shyam"
```

```
Name(2) = "Ashu"
```

```
Public Class Form1
```

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

```
Handles Button1.Click
```

```
Dim a(5) As Integer
```

```
Dim i As Integer
```

```
For i = 0 To 4
```

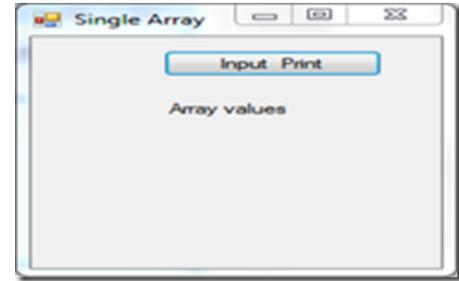
```
a(i) = InputBox("Enter Value")
```

```
Label1.Text = Label1.Text & Chr(13) & a(i)
```

```
Next
```

```
End Sub
```

```
End Class
```



Multi-Dimensional Array: VB.Net में कई dimensions के array create किए जा सकते हैं। ऐसे array जिसमें एक से जायदा subscripts होती हैं उन्हें multi-dimensional array कहते हैं। MD array को declare करने के लिए भी Dim, private या public statement का use किया जाता है। इसे declare करते समय इसमें एक से ज्यादा subscripts दी जाती हैं। यह subscripts के multiple के बराबर values store कर सकते हैं जैसे n(2,3) array में 6 values store की जा सकती हैं।

Syntax:

```
Dim <array name> (script1, script2 ...) As <Data
```

Type>

Exp-

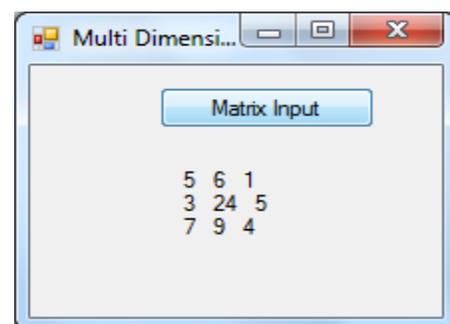
```
Dim matrix (3, 3) as Integer
```

```
Dim T (2, 3, 4, 2) As Double
```

Initializing Multi-Dimensional Array:

MD Array को initialize करने के लिए curly braces का use करते हैं।

```
Dim n (3, 3) As Integer = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}
```



```
Public Class Form1
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim mat(3, 3) As Integer
    Dim i, j As Integer
    'Take Input

    For i = 0 To 2
        For j = 0 To 2
            mat(i, j) = InputBox("Enter value on Matrix")
        Next
    Next
    'Printing Matrix
    Label1.Text = ""
    For i = 0 To 2
        For j = 0 To 2
            Label1.Text = Label1.Text & "    " & mat(i, j)
        Next
        Label1.Text = Label1.Text & Chr(13)
    Next

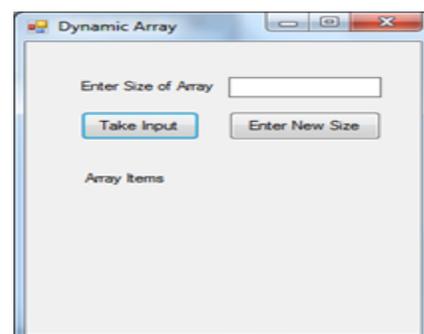
End Sub
End Class
```

Control Array:- Visual Studio की तरह vb.net में भी control array का use किया जाता है। एक ही प्रकार के control के group को control array कहते हैं यदि एक form में हमें 50 button का use करना है | इसके लिए 50 button को form में design करते हैं। इसका दूसरा option यह भी है की 50 button की control array का प्रयोग करे यद्यपि visual basic की तरह vb.net control array का सपोर्ट नहीं करता है। क्यों की इसमें control key index property नहीं होती है। फिर भी event module का use करके हम control key duplicate और expand कर सकते हैं।

Ex:- हम window application में 3 label तथा इसमें control array के माध्यम से button1 के click event का प्रयोग करके कुछ text display करेंगे |

```
Private sub form1_load()
    Me.text="Control Array"
    Button1.text="Click Me"
End Sub

Private Sub Button1_Click
    Dim t as Integer
    Dim M(2) as Label
    M(0) =label1
    M(1)=label2
    M(2)=label3
    For t=1 to 3
        M(t-1).text=t*20
    Next
End Sub
```

**Dynamic**उसकी
values**Array:-****Preserve Keyword:**

Dynamic array में Redim statement का use कर size को बदला जा सकता है पर ऐसे करने से इसमें store की गई

Guided by: Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)

lost हो जाती हैं। जब भी redim statement use किया जाता है इसमें store की गई सभी values lost हो जाती हैं और array blank हो जाता है। इस समस्या को हल करने के लिए Redim statement के साथ में Preserve keyword का use किया जाता है। यह keyword use करने पर array की size change होने पर भी उसमें store की गई values lost नहीं होती हैं। इस keyword को Redim के बाद use किया जाता है। जिसका syntax निम्न है।

```
Redim Preserve array_name (new_size)
```

```
Redim Preserve md (10)
```

Program 7: Dynamic Array

```
Public Class Form1
```

```
    Dim dy() As Integer
```

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

```
        Dim size, i As Integer
```

```
        size = Val(TextBox1.Text)
```

```
        ReDim dy(size)
```

```
        For i = 0 To size - 1
```

```
            dy(i) = Val(TextBox("Enter Value"))
```

```
            Label2.Text = Label2.Text & Chr(13) & dy(i)
```

```
        Next
```

```
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles Button2.Click
```

```
        Dim n_size, o_size, i As Integer
```

```
        n_size = Val(TextBox("Enter New Size"))
```

```
        o_size = dy.Length
```

```
        ReDim Preserve dy(n_size)
```

```
        If n_size > o_size Then
```

```
            For i = o_size To n_size
```

```
                dy(i) = Val(TextBox("Enter New Items"))
```

```
                Label2.Text = Label2.Text & Chr(13) & dy(i)
```

```
            Next
```

```
        End If
```

```
    End Sub
```

```
End Class
```

Collections:

Visual Basic.Net collections ऐसे data structures हैं जो की different ways से Data को store कर different type के operations provide करते हैं। यह कई प्रकार के operations provide करते हैं जो की Array में available नहीं होती हैं। VB.Net में मुख्यतः चार प्रकार के collections provide करता है।

1. ArrayList

2. HashTable

3. Stack

4. Queue

ArrayList:

ArrayList collection vb.net में available एक powerful data structure है। यह array की तरह ही multiple elements को maintain करने की सुविधा देता है। यह array में data values को आसानी से add, delete, insert, view etc. के लिए methods provide करता है। इसमें किसी भी location में data को insert किया जा सकता है, remove किया जा सकता है, find किया जा सकता है। यह एक dynamic data structure है जिसकी size automatically change हो जाती है। यह data को sort करने की भी सुविधा देता है।

Creating an ArrayList: ArrayList को उसे करने से पहले इसका एक object declare करना पड़ता है। इसे declare करते समय new keyword का use किया जाता है और इसकी size को नहीं define किया जाता है। जैसे

```
Dim alist As new ArrayList
```

Functions in ArrayList:

- Add: इस method का प्रयोग ArrayList में item को Add करने के लिए किया जाता है।

Syntax: ArrayList.Add (item)

Item: arraylist में add किया जाने वाला item.

```
Exp: alist.Add( "one")
```

```
alist.Add ("two")
```

```
alist.Add ("four")
```

- Insert: इस method को arraylist में item को specific position में insert करने के लिए किया जाता है। इसमें item की index की भी आवश्यकता पड़ती है जहां item को insert करना है।

Syntax: ArrayList.Insert (index, item)

Index: arraylist में item की position

Item: arraylist में add की जाने वाला item.

```
Exp: alist.Insert (2, "three")
```

- Remove: इस method का प्रयोग किसी item को arraylist से remove करने के लिए किया जाता है।

Syntax: ArrayList.Remove (item)

```
Ex: alist.remove ("one")
```

· RemoveAt: इस method का प्रयोग की item को उसकी index के द्वारा remove करने के लिए किया जाता है।

Syntax: ArrayList.removeat (index)

Ex- alist.Removeat (2)

· Sort: इस method का प्रयोग arraylist में उपलब्ध items को sort करने के लिए किया जाता है।

Syntax: ArrayList.Sort ()

Ex- alist.sort ()

· Item: इसका प्रयोग arraylist में store की गई values को access करने के लिए किया जाता है। items के access करने के लिए index का use किया जाता है।

Syntax: ArrayList.item(index)

Ex- alist.item(0)

· Count: इस property का use arraylist में available items को count करने के लिए किया जाता है। यह integer type की value return करती है।



```
Public Class Form1
```

```
    Dim alist As New ArrayList
```

```
    Private Sub RefreshLabel()
```

```
        Dim i As Integer
```

```
        Label1.Text = ""
```

```
        For i = 0 To alist.Count - 1
```

```
            Label1.Text = Label1.Text & alist(i) & Chr(13)
```

```
        Next
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button1.Click
```

```
        alist.Add(TextBox1.Text)
```

```
        RefreshLabel()
```

```
        TextBox1.Clear()
```

```
        TextBox1.Focus()
```

```
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button2.Click
```

```
        Dim i As Integer, item As String
```

```
        i = Val(TextBox1.Text)
```

```
        item = TextBox2.Text
```

```
        alist.Insert(i, item)
```

```
        RefreshLabel()
```

```
    End Sub
```

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Dim item As String
    item = InputBox("Enter item to Remove")
    alist.Remove(item)
    RefreshLabel()
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Dim i As Integer
    i = Val(InputBox("Enter index to remove"))
    alist.RemoveAt(i)
    RefreshLabel()
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    alist.Sort()
    RefreshLabel()
End Sub
End Class

```

HashTable:

hashtable का use paired type के data को store करने के लिए किया जाता है। यह collection भी arraylist की तरह ही होता है पर इसमें item को access करने के लिए key का use किया जाता है। इसमें सभी items की एक key होती है। value array में store होने वाली value ही होती केवल इसमें सभी values के साथ एक key भी होती है। hashtable में key और value दोनों object type के होते हैं।

Declaration of HashTable:

Syntax: Dim <HashTable name> As New HashTable

Ex- Dim htable As new HashTable

Important Functions used HashTable:

- Add: इस function का use hashtable में value को add करने के लिए किया जाता है। इसमें key - key value है और Value- add होने वाला item है।

Syntax: HashTable.Add (key, value)

Ex- htable.add (1, "A")

htable.add (2, "B")

htable.add (3, "C")

- ContainsKey(): इस function का use hashtable में key को check करने के लिए किया जाता है।

Syntax: HashTable.ContainsKey (Key)

Ex- htable.ContainsKey (1)

- ContainsValue: इस फंक्शन का use hashtable में value को check करने के लिए किया जाता है।

Syntax: HashTable.ContainsValue (Value)

Guided by: *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

Ex-htable.ContainsKey ("A")

- Remove: इसका use key और उसकी value को hashtable से delete करने के लिए किया जाता है।

Syntax: Hashtable.Remove(key)

Ex-htable.remove(2)

Program: HashTable:

```
Public Class Form1
```

```
Dim htable As New Hashtable
```

```
Private Sub RefreshLabel()
```

```
Dim objkey As Object
```

```
Dim objval As Object
```

```
Label1.Text = ""
```

```
For Each objkey In htable.Keys
```

```
objval = htable.Item(objkey)
```

```
Label1.Text = Label1.Text & "Key : " &
```

```
objkey.ToString & ", Value: " & objval.ToString &
```

```
Chr(13)
```

```
Next
```

```
End Sub
```

```
Private Sub Button1_Click(ByVal sender As
```

```
System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Dim oval, okey As Object
```

```
okey = InputBox("Enter Key")
```

```
oval = InputBox("Enter Value")
```

```
htable.Add(okey, oval)
```

```
RefreshLabel()
```

```
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button2.Click
```

```
Dim okey As Object
```

```
okey = InputBox("Enter Key to check")
```

```
If htable.ContainsKey(okey) = True Then
```

```
MsgBox("Key found")
```

```
Else
```

```
MsgBox("Not Found")
```

```
End If
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button3.Click
```

```
Dim oval As Object
```

```
oval = InputBox("Enter Key to check")
```

```
If htable.ContainsValue(oval) = True Then
```

```
MsgBox("Value found")
```

```
Else
```

```
MsgBox("Not Found")
```

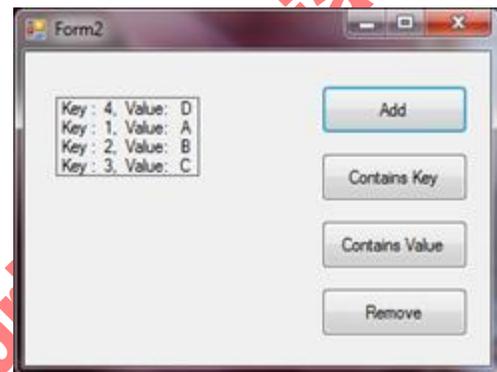
```
End If
```

```
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button4.Click
```

```
Dim okey As Object
```



```

    okey = InputBox("Enter Key to remove")
    htable.Remove(okey)
    RefreshLabel()
End Sub
End Class

```

Stack:

Stack भी एक VB.Net collections का टाइप है जो की Data structure के Stack की तरह ही कार्य करता है। यह last in first out के अनुसार काम करता है। इसमें डाटा input के लिए Push और Out के लिए Pop operation का use किया जाता है।

Declaration:

Syntax: Dim <stack name> As New Stack
Ex- Dim stk As new stack

Important Methods:

- Push: इसका प्रयोग stack में item को add करने के लिए किया जाता है।

Syntax: Stack.Push (object)

Ex- stk.push ("A")

- Pop: इसका प्रयोग stack से item को remove करने के लिए किया जाता है।

Syntax: Stack.pop

Ex- stk.pop

- Contains: इस method का use object को check करने के लिए किया जाता है।

Syntax: Stack.Contains (object)

Ex- stk.Contains ("A")

STACK

```

Public Class Form1

    Dim stk As New Stack

    Private Sub Refreshlabel()
        Dim i As Integer
        Dim ar() As Object

```

PROGRAM:

```

ReDim ar(stk.Count)
ar = stk.ToArray
Label1.Text = ""
For i = 0 To stk.Count - 1
Label1.Text = Label1.Text & ar(i) & Chr(13)
Next
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
stk.Push(TextBox1.Text)
TextBox1.Clear()
TextBox1.Focus()
Refreshlabel()
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

stk.Pop()
Refreshlabel()
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click

Dim s As String
s = InputBox("Enter value to check")
If stk.Contains(s) = True Then
MsgBox("Found")
Else
MsgBox("not Found")

End If
End Sub
End Class

```

Queue:

यह भी एक प्रकार का डाटा structure है जो की Queue की तरह work करता है। इसमें First in First out (FIFO) के अनुसार काम करता है। इसमें सबसे पहले insert किया गया item सबसे पहले बाहर जाएगा और सबसे last में input किया गया item last में बाहर जाएगा।

Syntax: Dim <Queue Name> As New Queue

Ex- Dim q as new queue

- Enqueue: इस method का use item को queue में add करने के लिए किया जाता है।

Syntax: Queue.Enqueue (object)

Ex- q.Enqueue ("A")

- Dequeue: इस method का use item को queue से delete करने के लिए किया जाता है।

Syntax: Queue.Dequeue ()

Guided by: *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

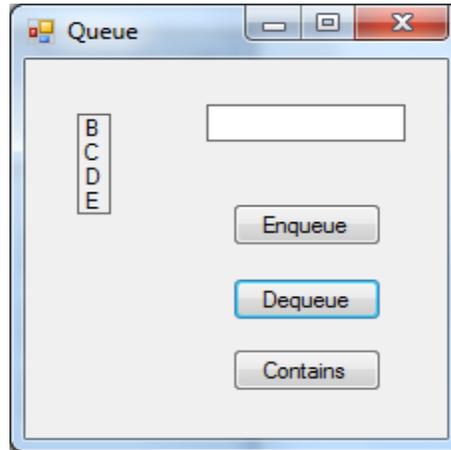
Ex- q.Dequeue ()

- Contains: इसका प्रयोग item को चेक करने के लिए किया जाता है।

Syntax: Queue.Contains (object)

Ex- q.Contains ("A")

Program: Queue:



```
Public Class Form1
```

```
Dim q As New Queue
```

```
Private Sub Refreshlabel()
```

```
Dim i As Integer
```

```
Dim ar() As Object
```

```
ReDim ar(q.Count)
```

```
ar = q.ToArray
```

```
Label1.Text = ""
```

```
For i = 0 To q.Count - 1
```

```
Label1.Text = Label1.Text & ar(i) & Chr(13)
```

```
Next
```

```
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button1.Click
```

```
q.Enqueue(TextBox1.Text)
```

```
TextBox1.Clear()
```

```
TextBox1.Focus()
```

```
Refreshlabel()
```

```
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button2.Click
```

```
q.Dequeue()
```

```
Refreshlabel()
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button3.Click
```

```
Dim s As String
```

```
s = InputBox("Enter value to check")
```

```
If q.Contains(s) = True Then
```

```
MsgBox("Found")
```

```
Else
```

```
MsgBox("not Found")
```

```
End If
```

```
End Sub
```

```
End Class
```

Subroutines:

Subroutines ऐसे statements के code blocks है जो की किसी define किए गए task को handle करने के लिए use किए जाते हैं। (a Subroutine is a block of statements that carries out a well-defined task.). Subroutines को Sub Procedures भी कहते है। Subroutines की statements Sub.....End Sub statement के block के बीच लिखी जाती हैं और इन्हे इनके नाम से identify किया जाता है। Subroutines C/C++ मे use होने वाले functions की तरह ही होते है केवल subroutines कोई value return नहीं करती है। Visual Basic को एक procedural लैङ्ग्वेज भी कहा जाता है क्योंकि इसमे ज़्यादातर codes sub procedures के अंदर लिखे जाते है। जैसे

```
Sub ShowDate()  
Msgbox( now( ) )  
End Sub
```

Declaring Subroutines: Subroutine को declare करने के लिए निम्नलिखित statement का प्रयोग किया जाता है।

```
<Access_modifier> Sub <Sub_Procedure_Name> ( [arguments] )  
[Statements]  
End Sub  
Ex- Private Sub ShowDate()  
Msgbox ( now())  
End Sub
```

Calling Syntax:

किसी भी sub procedure को call करने के लिए उसके name का use किया जाता है। sub procedure का name लिखकर use call करते हैं यदि उसमे arguments को use किया गया हो तो उन्हे भी define किया जाता है। call keyword का भी use subroutines को call करने के लिए किया जाता है। जैसे

```
Syntax- [Call] Sub_procedure_Name ([arguments])  
Ex- ShowDate ( )
```

या Call ShowDate ()

जब किसी भी subroutine को call किया जाता है तब program का control subroutine के procedure मे move हो जाता है। subroutine के execution के बाद end sub statement मे पहुंचने के बाद program का control फिर से calling program के पास वापस पहुंच जाता है। subroutine से program को directly exit करने के लिए exit sub statement का use किया जाता है।

Functions:

Functions subroutine के तरह ही code के blocks है जो की calling program को value return करते है। subroutines किसी define किए गए task को पूरा करने के बाद कोई भी value return नहीं करते है जबकि functions किसी define किए गए task को पूरा करने के बाद calling program को एक value return करते है। function subroutine की तरह ही arguments accept करते है। functions को declare करने के लिए Function... End function statement का use किया जाता है। जिसकी syntax निम्न लिखित है।

```
<Access modifier> Function <Function Name> ([arguments]) As <Return Type>
```

```
[Function's statements]
Return <returning value>
End Function
```

Ex-

```
Private Function Sum ( ) As Double
Dim a, b, c as Double
a = Val (Textbox1.Text)
b = Val (Textbox2.Text)
c= a + b
Return (c)
End Function
```

Access modifier function के scope को बताता है। यह private, public and friend हो सकता है।

Passing Variable:-

functions और subroutines directly program से interact नहीं कर सकते। अधिकतर procedures इन्हे call करने के लिए arguments का use करते हैं। इन arguments को इस procedures में value पास करने के लिए प्रयोग किया जाता है। यह arguments functions और subroutines के लिए input की तरह ही होते हैं जो इन्हे calling program के द्वारा दिया जाता है।

Ex-

```
Functions Min (ByVal a As Single, ByVal b as Single) As Single
If a<b then
    Min= a
else
    Min= b
End If
End Function
```

Argument- Passing Mechanisms:

VB.Net में किसी भी function या subprocedure में arguments को Pass किया जा सकता है। इसका passing mechanism determine करता है की pass किए जाने वाले arguments की value को subprocedure modify कर सकता है या नहीं।

Syntax- [Byval / ByRef] <variable_name> As <Data Type>

VB.Net procedures में arguments को pass करने के लिए निम्नलिखित दो प्रकार के machanisms का use किया जाता है।

□ **Passing Arguments by Values:** जब arguments by value pass किए जाते हैं तब procedure argument की copy को use करता है और उसमें किए गए changes का main argument पर कोई effect नहीं होता है। arguments को by value pass करने पर केवल उसी procedure में उसे change किया जा सकता है जहां इसे declare किया गया है। इसके लिए Byval keyword का use किया जाता है।

```
Private Function Sum(ByVal a As Integer, ByVal b As Integer) As Integer
    Return (a + b)
End Function
```

□ **Passing Arguments By Reference:** जब arguments को By Reference pass किया जाता है तब argument में actual variable pass हो जाता है। इसमें calling procedure के द्वारा main

variable का address pass किया जाता है जिससे called procedure में argument की value change करने पर pass किए गए argument को original value change हो जाती है। इसके लिए ByRef keyword का use किया जाता है।

```
Private Function Sum(ByRef a As Integer, ByRef b As Integer) As Integer
    Return (a + b)
End Function
```

Optional Argument:-

VB.Net में optional arguments भी दिये जा सकते हैं। optional arguments देने के लिए argument को declare करते समय ByVal/ Byref keyword के पहले Optional keyword का use किया जाता है और इसमें default value दी जाती है। Default value देना compulsory होता है। जिसका syntax निम्न है।

```
[Optional] [ByVal / ByRef] <variable_name> As <Data Type> = <Default Value>
```

Ex-

```
Private Function Sum(ByVal a As Integer, Optional ByVal b As Integer = 10) As Integer
    Return (a + b)
End Function
```

Optional argument देने पर call करने पर उसमें value को pass करना जरूरी नहीं होता है।

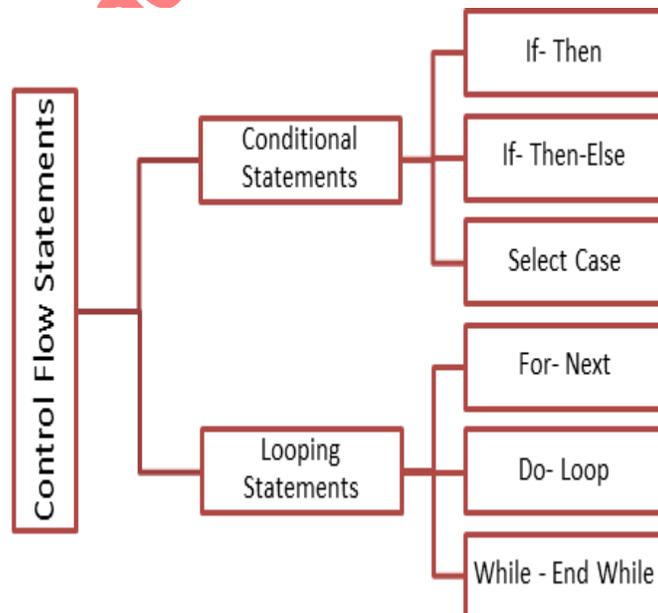
Ex- Dim s as Integer = Sum(10)

Control Flow Statements:

VB.Net में control flow statements Program के flow को control करती है। यह प्रोग्राम में use होने वाले data को control करती हैं। यह statements दो प्रकार की होती है।

- Conditional Statements
- Looping Statements

Conditional Statements:



VB.Net में conditional statements का use conditions को check करने के लिए किया जाता है। यह मुख्यतः तीन प्रकार की होती है।

- If- Then
- If - Then - Else
- Select Case

1. If- Then : if-then statement का use normal conditions check करने के लिए किया जाता है। इसमें लिखी गई condition के true होने पर If- End If block की statements execute हो जाती है। और condition false होने पर control बाहर चला जाता है।

Syntax-

```
If Condition Then
    Statements
End If
```

Ex- If a > b Then

```
    MsgBox ("A is greater")
```

```
End If
```

If- Then- Else:

इस conditional statement में दो block होते हैं। condition के true होने पर If block execute होता है और condition के false होने पर Else block execute होता है।

Syntax- If Condition Then

```
    [If block Statements]
```

```
Else
```

```
    [Else Block Statements]
```

```
End If
```

Ex- If a > b Then

```
    MsgBox ("A is greater")
```

```
Else
```

```
    MsgBox ("B is greater")
```

```
End If
```

Select Case: Select Case statement का use किसी expression की value को अलग अलग cases पर check करने के लिए किया जाता है। इसमें select case में लिखी गई expression की value cases के साथ match की जाती हैं और सही match found होने पर उस block की statement execute हो जाती है। किसी case के match न होने पर Case Else की statement execute होती है।

Syntax-

```
Select Case textexpression
```

```
Case ExpressionList.....N
```

```
    ['Statements ... .. N']
```

```
[Case Else
```

```
    [Statement Else]]
```

```
End Select
```

Ex-

```
Select Case Now.DayOfWeek
```

```
Case DayOfWeek.Sunday
```

```
    MsgBox("Its Time to Enjoy")
```

```
Case DayOfWeek.Monday
```

```
    MsgBox("Have a nice week")
```

```

        Case Else
            MsgBox("Enjoy this week")
    End Select

```

Else- If Ladder:

यह if-then-else का ही extended form है जिसमें multiple conditions और multiple results के लिए use किया जाता है। इसमें एक condition के false होने पर second, second के false होने पर third statement execute होती है और किसी के true न होने पर Else block execute होता है।

Syntax-

```

If condition1 Then
    'Statements1
ElseIf condition2 then
    'Statement2
ElseIf condition3 then
    'Statement3
...
...
ElseIf Condition N Then
    'Statement N
Else
    'Statement Else
End If

```

```

Ex- Dim per As Decimal
per = Val(TextBox1.Text)
If per > 60 Then
MsgBox("Firse")
ElseIf per > 50 Then
MsgBox("Second")
ElseIf per > 40 Then
MsgBox("Third")
Else
MsgBox("Fail")
End If

```

Looping Statements:

Looping statements किसी code या task को कई बार repeat करने के लिए use की जाती हैं। यह statements एक condition के according task को बार बार repeat करती है। VB.Net में मुख्यतः तीन प्रकार की looping statements पाई जाती हैं।

- 1) For- Next
- 2) While- End While
- 3) Do- Loop

- For-Next: यह looping statement एक popular looping statement है। इसका use एक series में statements को एक loop index के अनुसार बार बार execute करने के लिए किया जाता है।

Syntax-

```
For counter = start to end [Step increment/ decrements ]
    [Statements]
    [Exit For]
    [Statements]
Next [counter]
```

जहाँ counter एक variable है, start और end loop के starting and ending point को बताते हैं। step का use 1 से ज्यादा increment/Decrements के लिए किया जाता है। for loop को बीच में रोकने के लिए Exit For statement का use किया जाता है।

Ex-
For i = 0 To 10 Step 2
MsgBox(i)
Next

While- End While: यह loop condition के अनुसार loop को चलाने के लिए प्रयोग की जाती है। इस loop में एक condition को define कर किया जाता है जिसके True होने पर looping statements execute होती हैं और false होने पर यह directly बाहर हो जाता है। इसका syntax निम्नलिखित है।

Syntax-

```
While Condition
    [Statements]
    [Exit while]
    [Statements]
End While
```

Exit while का प्रयोग while loop को break करने के लिए किया जाता है।

Ex-
Dim i As Integer = 0
While i <= 10
MsgBox(i)
End While

Do-Loop :

यह loop while loop की तरह ही होता है। इसका प्रयोग किसी statement को indefinite time तक repeat करने के लिए किया जाता है। do loop statement को तब तक execute करता है जब तक while या until में लिखी condition false नहीं हो जाती है। इसमें दो keywords while और until का use किया जाता है। इसमें exit do statement होती है जो की loop को बीच में ही break करने के लिए use की जाती है। इस loop का syntax निम्नलिखित है।

Syntax-

```
Do {[While/ until] condition}
[Statements]
```

```
[Exit Do]
[Statements]
Loop {[While/ until] condition}
```

इस प्रकार Do- Loop चार प्रकार के होते हैं। जो निम्नलिखित हैं।

- i) Do While - Loop
- ii) Do Until - Loop
- iii) Do- Loop while
- iv) Do- Loop until

Do While- Loop : इसमें looping statement condition के true होने पर execute होती है। इसमें while को Do के साथ use किया जाता है।

```
Syntax-      Do While Condition
                [Statements]
                [Exit Do]
                [Statements]
                Loop
```

```
Ex-
Dim i, sum As Integer
i = sum = 0
Do While i <= 100
sum = sum + i
i = i + 1
Loop
MsgBox (sum)
```

Do Until - Loop: इसमें statements condition के false होने तक execute होती रहती है।

```
Syntax-
Do Until Condition
[Statements]
[Exit Do]
[Statements]
Loop
```

```
Ex-
Dim i, sum As Integer
i = sum = 0
Do Until i = 100
sum = sum + i
i = i + 1
Loop
MsgBox (sum)
```

Do- Loop While: इसमें looping statement condition के true होने पर execute होती है। इसमें while को loop के साथ use किया जाता है।

```
Syntax-      Do
```

```
[Statements]
[Exit Do]
[Statements]
Loop While Condition
```

Ex-

```
Dim i, sum As Integer
i = sum = 0
Do
sum = sum + i
i = i + 1
Loop While i <= 100
MsgBox(sum)
```

Do- Loop Until : Loop: इसमें statements condition के false होने तक execute होती रहती है।

Syntax-

```
Do
[Statements]
[Exit Do]
[Statements]
Loop Until Condition
```

Ex-

```
Dim i, sum As Integer
i = sum = 0
Do
sum = sum + i
i = i + 1
Loop Until i = 100
MsgBox(sum)
```

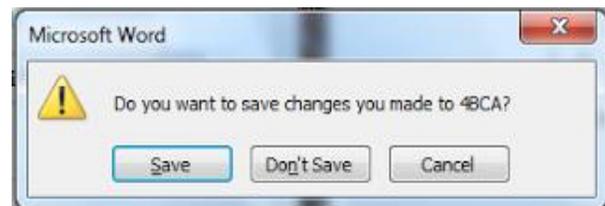
Msgbox: -

इस function का use message dialog box display करने के लिए किया जाता है। यह message box user click का wait करता है और click करने पर एक value return करता है जो की click की गई button को indicate करती है। यह function इसी तरह से कई प्रकार के message dialog box display करने के लिए use होता है।

इसका syntax निम्नलिखित है।

```
Public Function MsgBox(ByVal Prompt As Object[, ByVal Buttons As MsgBoxStyle =
MsgBoxStyle.OKOnly] [,ByVal Title As Object = Nothing]) As MsgBoxResult
```

Arguments:



1. Prompt: Required, String Expression, इसका use dialog box में display होने वाले Text message को define करने के लिए किया जाता है। prompt का maximum size 1024 characters होती है। prompt में multiple lines हो सकती हैं इसके लिए chr(13) और chr(10) का use किया जाता है।

2. Buttons: Optional , numeric expression: इसका use msgbox में display होने वाले buttons, icons, default button etc. के लिए किया जाता है। यह buttons, icon style, default button, text position etc. के लिए use किया जाता है।

3. Title: optional , String expression: इसका use message box के title bar में display होने वाले title को define करने के लिए किया जाता है।

Settings: इन settings का प्रयोग msgbox में buttons के place पर लिखा जाता है। यह msgbox के appearance और behavior को define करती है।

Member	Value	Description
OKOnly	0	OK button display करने के लिए
OKCancel	1	Cancel और OK button को display करने के लिए
AbortRetryIgnore	2	Abort, retry और Ignore button के लिए
YesNoCancel	3	Yes, no और Cancel के लिए
YesNo	4	Yes और no के लिए
RetryCancel	5	Retry और Cancel के लिए
Critical	16	Critical icon के लिए
Question	32	Question icon के लिए
Exclamation	48	Exclamation icon के लिए
Information	64	Information icon के लिए
DefaultButton1	0	First button को default करने के लिए
DefaultButton2	256	Second button को default करने के लिए
DefaultButton3	512	Third button को default करने के लिए
ApplicationModal	0	इस mode में user को application में work करने के लिए msgbox में click करना आवश्यक होता है।

SystemModal	4096	इसमें user के click करने तक सभी application suspend हो जाते हैं।
MsgBoxSetForeground	65536	Msgbox को foreground window बनाने के लिए
MsgBoxRight	524288	Text को right align करने के लिए
MsgBoxRtlReading	1048576	इसमें text Arabic system के अनुसार right to left हो जाते हैं।

इसमें 0-5 तक value buttons, 16,32,48,64 icons, 0,256,512 default button के लिए होते हैं।

Return Value

Ex-

```
Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
```

```
    Dim mr As MsgBoxResult
```

```
    mr = MsgBox("Are you want to exit", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Close")
```

```
    If mr = MsgBoxResult.Yes
```

```
Then
```

```
    e.Cancel = False
```

```
Else
```

```
    e.Cancel = True
```

```
End If
```

```
End Sub
```

Constants	Value
OK	1
Cancel	2
Abort	3
Retry	4
Ignore	5
Yes	6
No	7

INPUT BOX:- इसका use user से input लेने के लिए किया जाता है। इसमें एक dialog box display होता है जिसमें एक textbox होता है जिसमें डाटा input कर OK button पर क्लिक करते हैं। यह input की गई value को calling program को return कर देता है। इसमें एक textbox और दो button तथा एक message होता है। Public Function InputBox(ByVal Prompt As String [,ByVal Title As String = ""]

[ByVal DefaultResponse As String = ""],ByVal XPos As Integer = -1]

[, ByVal YPos As Integer =-1]) As String

Arguments:

- **Prompt:** Required, string expression, इसका use input box में display होने वाले message को define करने के लिए किया जाता है। इसकी maximum length 1024 character होती है।
- **Title:** Optional, string expression , इसे input box के title bar में title को define करने के लिए करते हैं।
- **Default Response:** optional, string expression: यह किसी भी प्रकार की value न देने पर default प्रयोग की जाती है।
- **XPos:** Optional, numerical expression : इसका प्रयोग input box की position define करने के लिए होती है। यह input box के left edge की दूरी screen से define करता है।

Guided by: Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)

• YPos: Optional, numerical expression : इसका प्रयोग input box की position define करने के लिए होती है। यह input box के Top edge की दूरी screen से define करता है।
Return Value: input box function input box के अंदर लिखे गए text को string form में return करता है।

Ex-

```
Dim s As String
s = InputBox("Enter your name", "Input Box", "AMAN")
MsgBox(s)
```

Unit-III

Working with Forms:-

Windows Form:

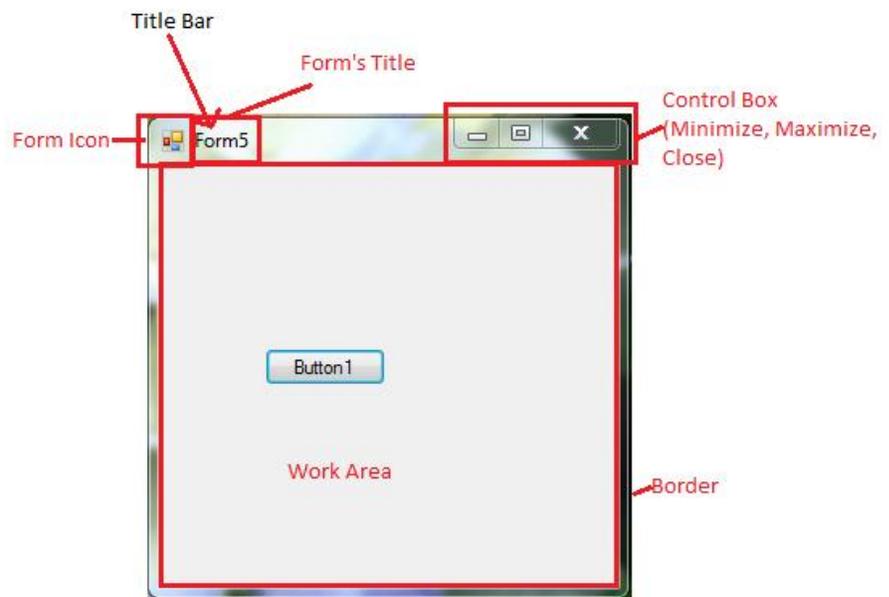
यह एक graphical application programming interface है अर्थात यह एक GUI component है जो की windows में GUI के लिए use किया जाता है। windows में display होने वाली सभी window, programming में windows form कहलाती है। form एक control है जो की .Net में windows interface provide करने के लिए use किए जाते हैं। यह control other windows controls के लिए container का काम करता है। इसमें windows के सभी controls को

place कर windows applications के लिए User Interface तैयार करते हैं। Visual Basic के programs को run करने पर display होने वाली window ही design time में form कहलाती है।

Form को बनाने के लिए user को कोई भी extra work नहीं करना पड़ता। यह built in functionality provide करता है। form को design करने के लिए form designer window के प्रयोग किया जाता है। जो visual studio IDE का main component है।

Windows form में Form का title bar, work area और border होता है जिसमें Title Bar में Form Icon, Form's Title, Control box जिसमें maximize, minimize और close को buttons होती हैं। इसके लिए windows form की properties होती हैं जैसे- controlbox, icon, Text etc. Work area वह स्थान है जहां user controls को place कर form को desing करता है।

Properties of Windows Form:



1. Name: form का name codes लिखते समय form को identify करने के लिए किया जाता है। इसे केवल design time में change किया जा सकता है।

Msgbox (Me.Name)

2. Text: यह property form के title bar में display होने वाले title को get या set करने के लिए use की जाती है।

Me.Text = "Windows Form"

3. AcceptButton: इस property का use form में available किसी एक button को default set करने के लिए किया जाता है जो की ENTER key press करने पर automatically click हो जाती है।

Me.AcceptButton = Button1

4. CancelButton: इस property का use form में available किसी एक button को default set करने के लिए किया जाता है जो की Esc key press करने पर automatically click हो जाती है।

Me.AcceptButton = Button2

5. AutoScroll: इस property के true होने पर form की size, available controls से कम होने पर automatically scrollbar display करने के लिए किया जाता है।

Me.AutoScroll = True

6. BorderStyle: इस property का use form के border style को set करने के लिए किया जाता है। इसकी निम्न वैल्यू होती हैं-

- (1) Sizable: Default, resizable window
- (2) None: Border less window can not resize
- (3) Fixed: window with visual border but can't resize
- (4) Fixeddialog: a fixed size dialog box
- (5) FixedSingle: fixed window with single line border
- (6) FixedToolWindow: contain only close button just like toolbox and can't resize
- (7) SizableToolWindow: contains only close button just like toolbox and can resize.

7. ControlBox: इस property का use control box (Maximize, Minimize और close button) को display या hide करने के लिए किया जाता है। इसके true होने पर यह display होंगे और false होने पर hide.

8. Minimizebox /MaximizeBox: इसके true होने पर minimizebox/ maximizebox display/enable होगा।

9. MinimumSize/MaximumSize: इस property का use form की minmum size और maximum size को set करने के लिए किया जाता है। इसमें दो sub properties height और width होती हैं।

10. Size: इस property को form की size get या set करने के लिए प्रयोग किया जाता है। इसमें दो subproperties height और winth होती हैं।

11. StartupPosition: इसका प्रयोग form के run होने पर initial position को सेट करने के लिए किया जाता है।

12. BackColor: इसका use form के background color को set या get करने के लिए किया जाता है।

13. BackgroundImage: इसे form के background में image set या get करने के लिए किया जाता है।

14. Font: इसे font get या set करने के लिए प्रयोग किया जाता है।

15. ForeColor: इसे form के font color को सेट या get करने के लिए प्रयोग किया जाता है।

Events of Form: सभी common event भी form में प्रयोग की जाती है।

• Load: Form Run होने से पहले memory में load होता है। इसके लोड होने पर Load Event होती है।

- Activate: यह form के activate होने पर perform होती है।
- Deactivate: यह form के deactivate होने पर perform होती है।
- FormClosing: यह form की close button पर click करते हैं perform होती है
- FormClosed: यह form के close होने के बाद perform होती है।
- Resize: यह form को resize करने पर perform होती है।
- Paint: यह form के desktop पर paint होने पर होती है।
- Move: यह form को move करने पर perform होती है।

Loading, Showing and Hiding forms:-

जब किसी भी windows application को Run किया जाता है तब पहला form automatically load होकर run हो जाता है। जब multiform application का use किया जाता है तब form को दूसरे form से load और manipulate किया जाता है। किसी अन्य form से दूसरे form के access करने के लिए सबसे पहले form का एक variable create करते हैं जिसमें second form का address हो।

Dim frm As New Form2

Declaration statement के call होते ही form automatically load हो जाता है। form के load होने के बाद form को show करने के लिए show () method का use किया जाता है।

Form.Show () Declaration statement के call होते ही form automatically load हो जाता है। form के load होने के बाद form को Hide करने के लिए Hide () method का use किया जाता है।

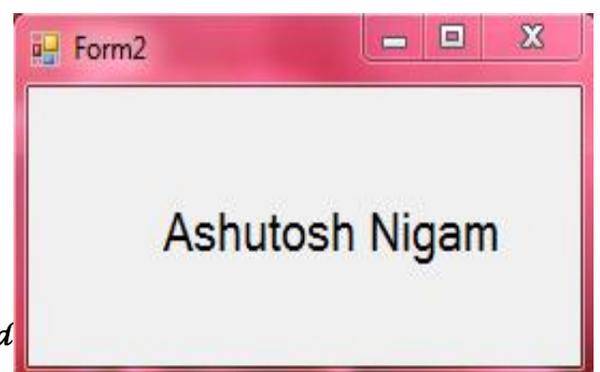
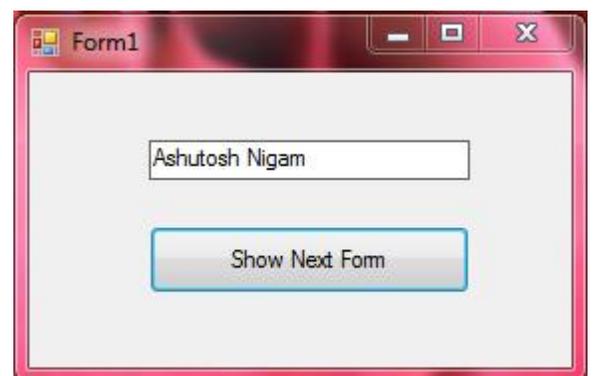
Controlling One Form within another

Form.Hide ()

हम जानते हैं multiform application forms में एक से अधिक form होते हैं। इस कारण एक form का दूसरे form से interaction आवश्यक होता है। इसके लिए एक form के दूसरे form से control करने की आवश्यकता पड़ती है। सभी forms independently operate होते हैं। उनके बीच communication के लिए public variables की आवश्यकता पड़ती है। other form को control करने के लिए उस form के controls और settings को access और set करना पड़ता है।

For ex- notepad application में notepad और find& replace को मिलाकर दो forms होते हैं जहां find और replace करते समय second form से first form के textbox control को access कर text को find और replace किया जाता है। इसी प्रकार एक form से एक form से other form को control किया जाता है।

Multiple forms के बीच communication बनाने के लिए public variables का use किया जाता है। इसके अलावा other form के controls और settings को access और set करने के लिए form के name का use किया जाता है। form के name का प्रयोग कर उसमें available सभी controls, उनकी properties,



form की properties, public methods और subroutines को access किया जा सकता है।

Ex- Form2.Label1.Text = "Second Form"

Program:

```
Public Class Form1
```

```
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Button1.Click
```

```
Form2.Label1.Text = Me.TextBox1.Text
```

```
Form2.Show ()
```

```
End Sub
```

```
End Class
```

GUI Programming with Windows Form:- TextBox Textbox एक control है जो कि user

को Text input करने की सुविधा देता है। यह text को display और उसे edit करने की facility देता है।

यह windows का सबसे common और popular control है जो की text input की facility provide करता

है। यह एक text editor है जो की text को input करने के साथ edit करने के सभी facility provide

करता है जैसे- Text insert, cut, copy, paste, color change, font changing etc.

Basic Properties: सभी common properties इसमें apply होंगी।

Name: (String)

Textbox control का नाम जो की coding में control को identify करने के लिए use होता है। इसे runtime में change नहीं किया जा सकता है।

```
MsgBox(TextBox1.Name)
```

Text: (String)

यह textbox में display होने वाले text को get या set करने के लिए किया जाता है।

```
TextBox1.Text = "VB.Net"
```

Multiline: (True, False)

इस property को textbox को single या multiline करने के लिए किया जाता है। इसके true होने पर textbox में multiple lines लिखी जा सकती है। defaultly यह false रहती है। textbox में केवल single line input की जा सकती है।

```
TextBox1.Multiline = True
```

WordWrap: (True, False)

इसके true होने पर text के multiline होने पर words automatically warp हो जाते हैं।

```
TextBox1.WordWrap = True
```

Scrollbar: (None, Horizontal, Vertical, Both)

यह property scrollbars को add करने के लिए किया जाता है।

```
TextBox1.ScrollBars = ScrollBars.Horizontal
```

PasswordChar: (char)

इस property को password character set करने के लिए किया जाता है। इसे केवल design time में सेट कर सकते हैं। इसमें single character input करते हैं।

MaxLength: (Integer):

इसका use textbox में input किए जाने वाले text की maximum length को set करने के लिए किया जाता है।

```
TextBox1.MaxLength = 15
```

SelectedText: (string)

इसका use select किए गए text को get करने के लिए किया जाता है।

```
MsgBox(TextBox1.SelectedText)
```

ReadOnly: (True, False)

इसका use Textbox को read only करने के लिए किया जाता है। इसके true होने पर textbox में text input नहीं किया जा सकता है।

```
TextBox1.ReadOnly = True
```

Methods of Textbox: सभी common methods textbox में use होती हैं जैसे- Show (), Hide (), Focus (), Clear () .

```
TextBox1.Show()
```

```
TextBox1.Hide()
```

```
TextBox1.Clear()
```

```
TextBox1.Focus()
```

□ `Select (StartIndex, Length)`: इस method का use textbox में text को select करने के लिए किया जाता है। इस method में दो arguments होते हैं।

a. `StartIndex`: integer value जो की select start करने के लिए text की index के लिए use होता है।

b. `Length`: integer value, select होने वाले text की length define करने के लिए use किया जाता है।

Syntax- `Textbox.Select (ByVal StartIndex as Integer, ByVal Length as Integer)`

Ex- `Textbox1.Select (3, 8)`

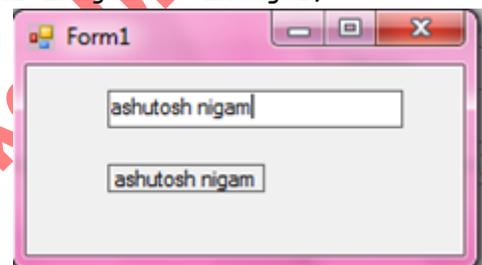
Events: सभी Common Event इसमें apply होती है। इसके अलावा

निम्न मुख्य events होती हैं।

□ `TextChanged`: यह event textbox में text के modify (change) होने पर perform होती है।

□ `KeyPress`: यह event textbox में focus होने पर keyboard की किसी भी keypress करने पर perform होती है।

□ `KeyDown/ keyUp`: यह event Key को press और release करने पर perform होती है।



Public Class

```
Private Sub TextBox1_TextChanged (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
```

```
Label1.Text = TextBox1.Text
```

```
End Sub
```

```
End Class
```

Label :-

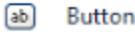
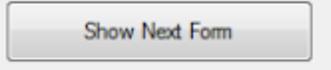
Label control windows में use होने वाला सबसे important control है जो कि किसी भी message को display करने के लिए किया जाता है। इसमें लिखा गया text को runtime में बिना coding के नहीं modify किया जा सकता है। इसके use others controls के बारे में information display करने के लिए किया जाता है।

Properties:

A Label

□ Autosize: इसके तुरंत होने पर label कि size text के according automatically change हो जाती है।

BUTTON:-

यह भी windows operating system में use होने वाला most important कंट्रोल है जो कि किसी work को perform करने के लिए use किया जाता है। इसकी मुख्य event click होती है। button के click event के लिए लिखे गए codes, button में click करने  पर execute होते हैं और define किए गए work को perform करते हैं। 

Properties: सभी common properties इसमें apply होती हैं।

Methods: सभी common methods, clear () को छोड़कर इसमें apply होती हैं। performClick () इसकी important method है जो कि coding के द्वारा button पर click करने के लिए use की जाती है।

```
Button1.PerformClick ()
```

Events: सभी common events इसमें apply होती हैं। click button कि सबसे important event है।

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

Listbox:-

Listbox control का use form में list को display करने के लिए किया जाता है। Listbox user को selection provide करता है जिसमें user single या multiple items को select कर सकता है। इसमें items की एक list display होती है। जिसमें user items select कर सकता है।

Properties: सभी common properties इसमें apply होंगी। इसकी other important properties निम्न हैं।

1. Items: यह listbox की सबसे important property है। यह एक collection है जो कि listbox में items को contain करती है और उन्हें manipulate करने के लिए use की जाती है। यह कई methods provide करती है जो methods में define हैं।

```
MsgBox (ListBox1.Items.Item (2))
```

2. MultiColumn: इस property के तुरंत होने पर listbox में items multiple columns में display होते हैं।

```
ListBox1.MultiColumn = True
```

3. SelectionMode: यह property list items के selection mode को define करने के लिए use की जाती है। इसके निम्न values होती हैं।

None	No selection
One	Default, Single item can be selected
MultiSimple	Simple multiple selection, mouse click is used to select items
MultiExtended	Extended Multiple Selection, Mouse Click is used

	with Control key to select items.
--	-----------------------------------

```
ListBox1.SelectionMode = SelectionMode.One
```

4. Sorted: इसके तुरंत होने पर list के items automatically ascending या descending order में sort हो जाएंगे।

```
ListBox1.Sorted = True
```

5. Text: इस property का use listbox में select किए गए item या text को get करने के लिए किया जाता है।

```
MsgBox(ListBox1.Text)
```

6. SelectedText/ SelectedItem: इन दोनों properties का use listbox में select किए गए item को get करने के लिए किया जाता है।

```
MsgBox(ListBox1.SelectedItem)
```

7. SelectedIndex: इसका use listbox में select किए गए item की index get करने के लिए किया जाता है।

```
MsgBox(ListBox1.SelectedIndex)
```

Methods: इसमें show (), hide () और focus () method को apply किया जाता सकता है। साथ ही इसमें items collection में निम्न methods होती हैं।

1. ADD: इस method का use Items collection में items को add करने के लिए किया जाता है। इसमें single argument होता है। item argument object type का होता है।

```
Syntax-      Listbox.Items.Add (Item)
Ex-          ListBox1.Items.Add ("Monday")
ListBox1.Items.Add ("Tuesday")
ListBox1.Items.Add ("Wednesday")
ListBox1.Items.Add ("Thursday")
ListBox1.Items.Add ("Friday")
ListBox1.Items.Add ("Saturday")
ListBox1.Items.Add ("Sunday")
```

2. Remove: इस method का प्रयोग items collection से किसी item को remove करने के लिए किया जाता है। किसी item को remove करने के लिए उस item के नाम का use किया जाता है। इसमें Item का name, object type में होती है।

```
Syntax-      Listbox.Items.Remove (Item)
Ex-          ListBox1.Items.Remove ("Sunday")
```

3. RemoveAt: इस method का प्रयोग items collection से किसी item को remove करने के लिए किया जाता है। किसी item को remove करने के लिए उस item की index का use किया जाता है। इसमें Item का index, integer type में होती है।

```
Syntax-      Listbox.Items.RemoveAt (Item)
Ex-          ListBox1.Items.RemoveAt (5)
```

4. Clear: इस method का use, items collection से सभी items को एक साथ remove करने के लिए किया जाता है।

```
Syntax-      Listbox.Items.Clear ()
Ex-          ListBox1.Items.Clear ()
```

5. Insert: इस method का use Items collection में किसी Item को insert करने के लिए किया जाता है। Item को collection में insert करने के लिए उस position की index का use किया जाता है।

इसमें दो arguments होते हैं। पहला argument index होती है जो integer type की होती है। और second argument, Item होता है जो की object type का होता है।

Syntax- `Listbox.Items.Insert (Index, Item)`
Ex- `ListBox1.Items.Insert (1, "Friday")`

6. Contains: इस method का use, Items collection में किसी Item को check करने के लिए किया जाता है। यदि वह item collection में available होता है तो यह true value return करती है और यदि item collection में नहीं होता है तो यह false return करती है।

Syntax- `Listbox.Items.Contains (Item)`
Ex- `Dim b as Boolean = ListBox1.Items.Contains ("Friday")`

7. Count: यह एक property है जो कि items collection में items को count करने के लिए use की जाती है।

Syntax- `Listbox.Items.Count`
Ex- `MsgBox (ListBox1.Items.Count)`

Events: इसमें सभी common events apply उपस्थित होती हैं। इसके अलावा इसकी सबसे main event SelectedIndexChanged है।

SelectedIndexChanged: यह event Listbox में किसी भी Item को select करने पर होती है। जब किसी भी item को select करते हैं तब selected index change हो जाती है तब यह event perform होती है।

```
Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListBox1.SelectedIndexChanged
```

End Sub

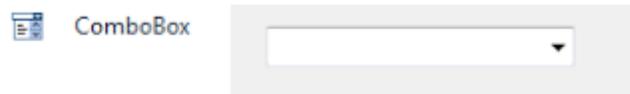
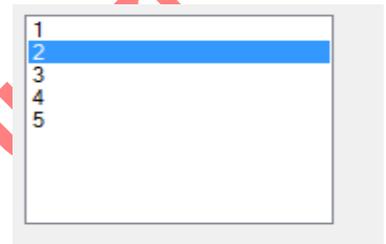
ComboBox: -Listbox और Textbox का combination है जिसका use list को display करने के लिए किया जाता है। यह list को dropdown

format में display करता है। इसमें एक textbox होता है जिस पर text input कर सकते हैं साथ ही एक arrow

button पर click करने पर एक list display होती है जिस पर items को select कर सकते हैं। यह listbox की जगह कम space लेता है। और list को भी display करता है। ComboBox में केवल एक item को select कर सकते हैं।

Properties: इसमें Listbox की सभी properties apply होती हैं। साथ ही इसमें कुछ special property होती हैं जो निम्न हैं।

DropDownStyle: इसका use combobox की style set करने के लिए किया जाता है। इसकी निम्न value होती हैं।



<p>DropDown</p>	<p>Default, इसमें combobox में एक textbox display होता है जिस पर लिखा सकता है और list से items select भी कर सकता है.</p> 
<p>DropDownList</p>	<p>इसमें user text input नहीं कर सकता है केवल पहले से available items को सिलैक्ट कर सकता है।</p> 

Simple	<p>इसमे textbox और list दोनों display होते है। इसमे list dropdown type नहीं होती है।</p> 
--------	---

Methods: इसमे Listbox की सभी methods apply होती हैं।

Events: इसमे Listbox की सभी events apply होती हैं।

CHECKBOX:-

CheckBox control का use user को option select करने के facility provide करने के लिए किया जाता है।

यह user को multiple selection provide करता है अर्थात multiple checkbox मे multiple checkbox को एक साथ select किया जा सकता है। checkbox control मे text display होता है और text से पहले एक rectangular button display होती है जिसके सिलैक्ट होने पर एक check का symbol इसमे display होने लगता है और unselect होने पर यह check symbol display नहीं होता है।

Properties: सभी Common

properties इस control मे apply होती हैं। साथ ही इसमे कुछ properties निम्न हैं।

Checked: इसका use check box को check या uncheck करने के लिए और उसकी वैल्यू को get करने के लिए किया जाता है। इसके true होने पर checkbox checked होता है और false होने पर यह unchecked होता है।

Methods: सभी common methods इसमे apply होंगी।

Events: सभी common events भी इसमे apply होती हैं। अन्य events निम्न हैं।

CheckChanged: यह checkbox के select या unselect होने पर होती है।

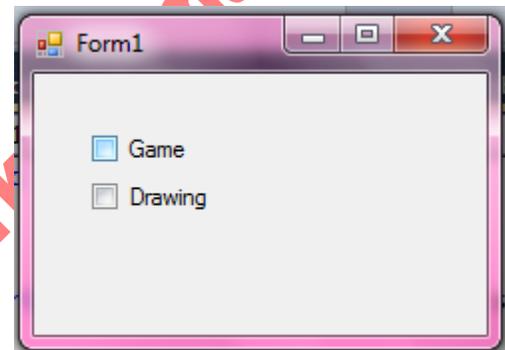
```
Private Sub CheckBox1_CheckedChanged
(ByVal sender As Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
End Sub
```

```
Public Class Form1
```

```
Private Sub CheckBox1_CheckedChanged (ByVal sender As Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged, CheckBox2.CheckedChanged
If CheckBox1.Checked = True Then
MsgBox ("Game")
End If
If CheckBox2.Checked = True Then
MsgBox ("Drawing")
End If
End Sub
End Class
```

Picture Box:-

Guided by: *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

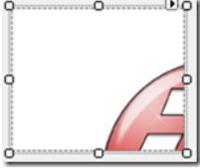


Picture Box Windows GUI का एक important हिस्सा है। यह windows में Images को display करने में help करता है। Visual Basic.Net में picture box control पाया जाता है जो  PictureBox की Windows Form में किसी भी image को display करने के लिए use किया जाता है। इसे BMP, JPEG, JPG, PNG, icons etc. के लिए use किया जाता है। यह कई प्रकार की properties और methods provide करता है जो कि image को set और manipulate करने में help करती हैं। इसमें निम्न properties, methods और events होती हैं।

Properties: सभी common properties, Text छोड़कर इसमें apply होंगी। other properties निम्न हैं।

- Image: यह picture box की सबसे important property है। इसका use picture box में picture को get और set करने के लिए किया जाता है।
- BorderStyle: यह control के border style को set करने के लिए use की जाती है। इसमें तीन values- None, FixedSingle और Fixed3D होती हैं।
- SizeMode: यह property image को picture box में arrange करने के लिए use की जाती है। इस property की निम्न values होती हैं।



Normal	इस mode में image normal size में रहती है अर्थात image की size picture box से अधिक होने पर यह display नहीं होती है।	
StretchImage	इसमें image, picture box के size में automatically arrange हो जाती है।	
Autosize	इसमें picture box, image की size के equal size का हो जाता है।	
CenterImage	इसे use करने पर न picture box की size change होती है और न ही image की। इसमें image का center part picture box में display होता है।	
Zoom	इसमें image, picture box में fill हो जाती है।	

Methods: सभी common methods इसमें use होती हैं। साथ ही इसकी कुछ methods निम्न हैं-

• RotateFlip: इसका use image को rotate करने के लिए किया जाता है। इसे निम्न प्रकार से use किया जाता है।

Syntax- Image.RotateFlip (Type)

Ex- pictureBox.Image.RotateFlip (Rotate180FlipNone)

• Save: इसका use image को save करने के लिए किया जाता है। इसके लिए image के path का use किया जाता है।

Syntax- Image.Save (path)

Ex- PictureBox1.Image.Save ("C:\tmp.bmp")

Events : इसमें सभी common events use की जा सकती हैं।

Radio



RadioButton

Button:-

Radio button control का use भी checkbox की तरह ही options provide करने के लिए किया जाता है। Radio button checkbox जैसे की selection provide करती है लेकिन इसमें केवल एक group में केवल एक ही radio button को select किया जा सकता है। यह single selection provide करता है जबकि checkbox multiple selection provide करता है।

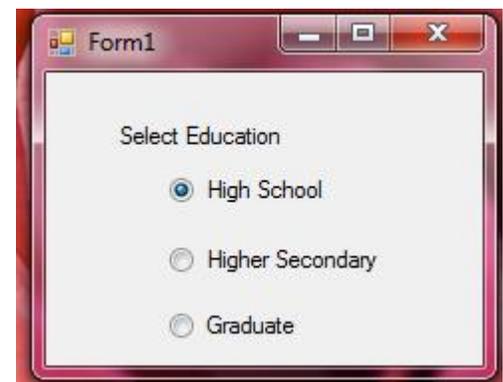


Radiobutton में एक circle display होता है जब किसी RadioButton को select किया जाता है तब उस circle के अंदर एक और filled circle बन जाता है। इसमें जब group के किसी दूसरे radiobutton को select किया जाता है तब पहले वाली radiobutton unselect हो जाती है और select की गई button select हो जाती है। RadioButtons का group बनाने के लिए container control जैसे - Panel, Group Box etc. का use किया जाता है।

Properties: Checkbox की सभी properties radio button में भी use होती हैं।

Methods: Checkbox की सभी methods भी radiobutton में use होती हैं।

Events: CheckBox की सभी events Radio Button में available होती हैं।



```
Public Class Form1
```

```
Private Sub RadioButton1_CheckedChanged (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged,
    RadioButton2.CheckedChanged,
    RadioButton3.CheckedChanged
```

```
    If RadioButton1.Checked = True Then
```

```
        MsgBox ("High School")
```

```
    ElseIf RadioButton2.Checked = True Then
```

```
        MsgBox ("Higher Secondary")
```

```
    Else
```

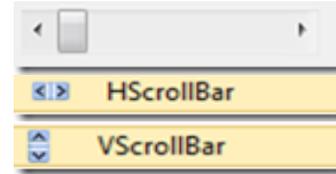
```
        MsgBox ("Graduate")
```

```
    End If
```

End Sub

End Class

Scroll Bar: - Scroll Bar windows में use होने वाला एक महत्वपूर्ण GUI component है। Scrollbar एक long strip होती है जो की use को किन्ही दो values के बीच selection provide करने के लिए use की जाती हैं। यह मुख्यतः windows में use होने वाले gui components में screen



display न होने वाले components को show करने के लिए use किया जाता

है। scrollbar control में दो end points होते हैं। और इसके बीच एक

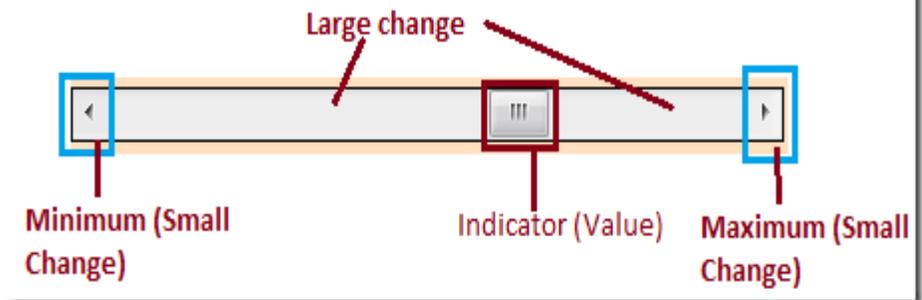
indicator होता है जो कि दोनों end points के बीच value को select करने के लिए use होता है।

Scrollbar control Horizontally और vertically दोनों प्रकार से use किया जाता है। Horizontal Scrollbar के लिए HScrollBar और Vertical Scrollbar के लिए VScrollBar Select किया जाता है।

Scrollbar के दोनों ends पर click करने इसकी value धीरे धीरे change होती है जबकि indicator और ends के बीच click करने पर यह जल्दी

change होती है।

Properties: इसमें सभी common properties, text को छोड़कर available होती हैं। other properties निम्नलिखित हैं।



- **Minimum:** इस property का use scrollbar control की minimum value set और get करने के लिए किया जाता है। इसकी default value 0 होती है।
- **Maximum:** इस property का use scrollbar control की Maximum value set और get करने के लिए किया जाता है। इसकी default value 100 होती है।
- **SmallChange:** इसका use scrollbar के ends पर click करने पर indicate की value में होने वाले change को set और get करने के लिए किया जाता है।
- **LargeChange:** इसका use scrollbar के ends और Indicator के बीच के space पर click करने पर indicate की value में होने वाले change को set और get करने के लिए किया जाता है।
- **Value:** इसका use Indicator कि current position को get या set करने के लिए किया जाता है।
- **Dock:** इसका use scrollbar कि position को set करने के लिए किया जाता है जिसकी निम्न वैल्यू होती हैं।- None, Left, Right, Top, Bottom और Fill.

Methods: इसमें सभी common methods available होती हैं।

Events: सभी common events के साथ ही इसमें निम्न events भी होती हैं।

- **Scroll:** यह event scrollbar के indicator को scroll करने पर perform होती है।

```
Private Sub HScrollBar1_Scroll (ByVal sender As Object, ByVal e As System.Windows.Forms.ScrollEventArgs) Handles HScrollBar1.Scroll
```

Guided by: Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)

End Sub

- ValueChanged: यह scrollbar के indicator कि value (Position) change करने पर perform होती है।

```
Private Sub HScrollBar1_ValueChanged (ByVal sender As Object, ByVal e As System.EventArgs) Handles HScrollBar1.ValueChanged
```

End Sub

Example:

```
Public Class Form5
```

```
Private Sub Form5_Load (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
Me.HScrollBar1.Maximum = 255
```

```
Me.HScrollBar2.Maximum = 255
```

```
Me.HScrollBar3.Maximum = 255
```

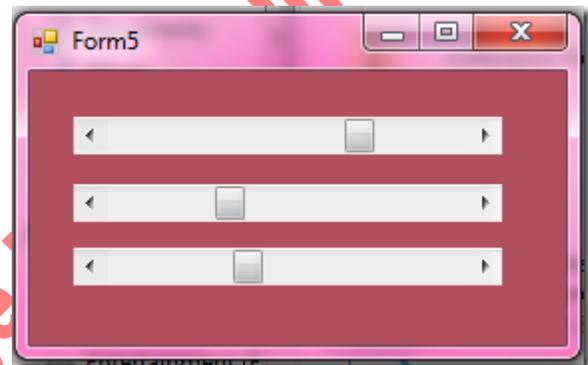
End Sub

```
Private Sub HScrollBar1_Scroll (ByVal sender As System.Object, ByVal e As System.Windows.Forms.ScrollEventArgs) Handles HScrollBar1.Scroll, HScrollBar2.Scroll, HScrollBar3.Scroll
```

```
Me.BackColor = Color.FromArgb (HScrollBar1.Value, HScrollBar2.Value, HScrollBar3.Value)
```

End Sub

End Class



Timer

Timer Control एक Looping control है जो कि किसी task को एक given time of Interval में repeat करने के लिए use किया जाता है। यह किसी भी task जो कि इसके Tick event handler में define किया जाता है उसे इसके interval property में define किए गए time के अनुसार repeat करता है। यह timer के start () method के call होने पर start हो जाता है। और तब तक work को repeat करता रहता है जब तक Timer को stop नहीं किया जाता है। यह control runtime में display नहीं होता है। इसकी properties, methods और events निम्न हैं।

Properties: इस control में सबसे कम properties होती है। जो निम्न हैं।

- Name: यह control के name को set करने के लिए use की Msgbox (Timer1.Name)



जाती है।

- Enabled: यह Timer control को enable और disable करने के लिए use कि जाती है। यह defaultly True होती है।

```
Timer.Enabled = True
```

- Interval: इसका use Timer control के repetition interval को get और set करने के लिए use की जाती है। इसी interval के अनुसार ही timer task को repeat करता है।

```
Timer1.Interval= 2000
```

Methods: Timer में निम्न दो methods होती है।

- Start: इस method का use Timer को Start करने के लिए किया जाता है।

```
Timer1.Start ()
```

- Stop: इसका use Timer को stop करने के लिए किया जाता है।

```
Timer1.Stop ()
```

Events:

- Tick: यह event Timer के interval property में set किया गए Time के अनुसार Task को Repeat करने के use किया जाता है। यह Timer की default events है जो Timer के start() और stop() method के बीच task repeat करती है।

```
Private Sub Timer1_Tick (ByVal sender As Object, ByVal e As System.EventArgs) Handles  
Timer1.Tick
```

```
End Sub
```

Program:

```
Public Class Form1
```

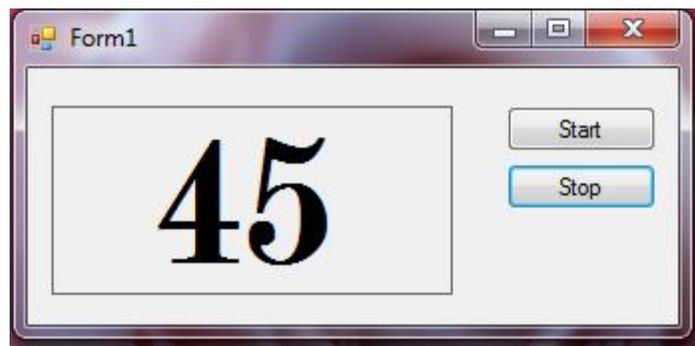
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
HandlesButton1.Click
```

```
Timer1.Start()  
End Sub
```

```
Private Sub Button2_Click(ByVal  
sender As System.Object, ByVal e As  
System.EventArgs) HandlesButton2.Click  
Timer1.Stop()  
End Sub
```

```
Private Sub Timer1_Tick(ByVal  
sender As Object, ByVal e As  
System.EventArgs) Handles Timer1.Tick  
Label1.Text = Val(Label1.Text) + 1  
End Sub  
End Class
```

List View:-

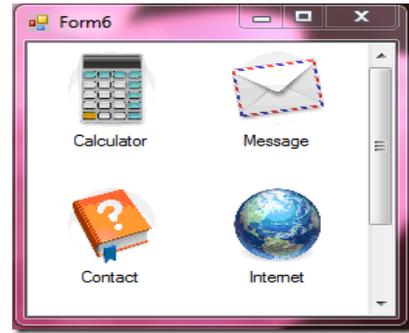


List View Control VB.Net में provide किया गया एक important control है। Windows में list View control का बहुत use किया जाता है। इसे items को List में display करने के लिए use किया जाता है जिसमें items का text, icons, type etc. information भी display होती है। Windows Explorer में Display होने वाले folders और files की List इसका एक example है। इसमें कई प्रकार की property होती हैं जो इसके appearance को control करती है जैसे - Small Icon, Large Icon, Report, List etc. My Computer, My Document etc. इसकी example हैं।

Properties: इसमें text को छोड़कर सभी common properties होती है।

- Name: इसका use Control के name को set और get के लिए किया जाता है। जो की program में control को identify है।

- View: इसका use Listview control में display वाले items के view को set करने के लिए किया जाता है। इसमें निम्न values होती हैं-



use

करने करता

होने

LargeIcon	इसे प्रयोग करने पर icons large size में display होते हैं.
SmallIcon	इसे प्रयोग करने पर icons small size में display होते हैं.
List	इसमें items list में display होते हैं और केवल उनका caption display होता है।
Report	इसमें items और उसके sub items display होते हैं।

- Arrange: इसे item के arrangement के set करने के लिए use किया जाता है।

Default	Item को move करने पर यह उसी drop किए गए place पर रहेगा
Left	यह control के left side में automatically align हो जाएगा
SnapToGrid	Item automatically invisible grid में align हो जाएगा drop करने पर
Top	Item Control के top पर align हो जाएगा

- HeaderStyle: यह property Report View Select करने पर Listview के Header text को सेट करने के लिए use की जाती है। इसकी निम्न value होती हैं -

Clickable	Visible column जहाँ header click करने पर respond करेगा
NonClickable	Visible column header लेकिन click करने पर कोई resond नहीं होता है।
None	No visible column header

- Activation: यह property listView के items के activation को define करने के लिए किया जाता है।

OneClick	Items single click करने पर activate हो जाता है। items का color, shape और caption cursor के जाने पर change हो जाता है।
Standard	Items double click करने पर activate होता है। उसके color, shape etc. में कोई change नहीं होता।
TwoClick	Items double click में activate होता है। cursor के item के ऊपर होने पर Textcolor, shape etc. change हो जाता है।

- FullRowSelect: इसके true होने पर item को select करने पर पूरी row select हो जाती है। यह केवल report view में work करती है।
- GridLines: इसके true होने पर items और उसके subitems के बीच lines draw हो जाती है। यह report view में work करती है।
- LabelEdit: इसके true होने पर user items के text को modify कर सकता है। यह defaultly false होती है।
- MultiSelect: यह determine करती है की user multiple items को select कर सकता है या नहीं। इसके true होने पर multiple items को select किया जा सकता है।
- Scrollable: इसके true होने पर scrollbar automatically display होता है।
- Sorting: यह property determine करती है की items को sort होंगे या नहीं। इसके true होने पर items automatically sort हो जाते हैं।

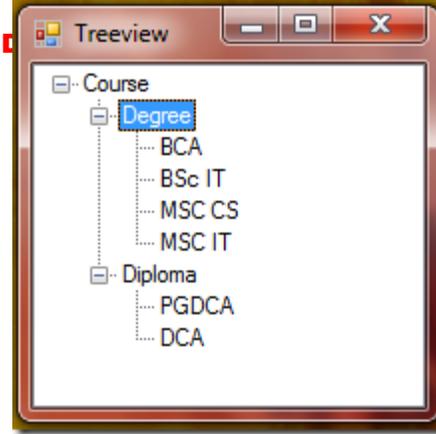
Methods: इस control में common methods use होती हैं। इसमें कई methods होती हैं जो की इसके collections में available होती हैं।

TREE VIEW:-

TreeView control का use डाटा को Tree format में display करने के लिए किया जाता है। यह Data structure के Tree डाटा स्ट्रक्चर को implement करता है। इसमें information higherachical structure में display करता है। यह windows explorer की तरह ही items को explorer करती है।

Properties:

- Name: इसका use treeview control के name को set करने के लिए किया जाता है। जो की coding में use होता है।
- CheckBoxes: इसके true होने पर Tree items में checkbox भी display होता है। जो selection में use होता है। इसे multiple selection के लिए use किया जाता है।
- ShowLine: इसके true होने पर tree control में child और parent node के बीच line display होती है।
- ShowPlusMinus: इसके true होने पर tree nodes में plus/ minus की button display होती हैं यदि कोई child item available होता है।
- FullRowSelect: इसके true होने पर किसी node को select करने पर उसके सभी child node भी select हो जाते हैं।
- ShowRootLine: इस property के true होने पर Tree control में एक rootline display होती है।
- Sorted: इसके true होने पर Tree view के nodes sort होकर display होंगे।
- Text: यह property Tree view control में select किए गए node के text को represent करती है।
- TopNode: इसका use TreeView के First node को get करने के लिए किया जाता है।



items

दोनों

- VisibleCount: यह सभी visible tree nodes के को count करके return करती है।

TreeView control में display होने वाले node, nodes collection के अंदर पाये जाते हैं। nodes collection, TreeView control में nodes को add, remove etc. के लिए methods provide करता है। nodes collection में design और run time में nodes को manipulate किया जा सकता है। design time में यह nodes collection editor provide करता है जिसे treeview के nodes property पर click कर display किया जा सकता है। Runtime में इसमें nodes को manipulate करने के लिए निम्न methods का use किया जाता है।

- Add: इस method का use nodes collection में nodes को add करने के लिए किया जाता है।

Syntax: Nodes.Add (Node_Caption)

Nodes.Add (Node_Object)

Nodes.Add (index, Node_Object)

Ex-

```
Me.TreeView1.Nodes.Add("Course")
```

```
Me.TreeView1.Nodes(0).Nodes.Add("Degree")
```

```
Me.TreeView1.Nodes(0).Nodes.Add("Diploma")
```

```
Me.TreeView1.Nodes(0).Nodes(0).Nodes.Add("BCA")
```

```
Me.TreeView1.Nodes(0).Nodes(0).Nodes.Add("BSc IT")
```

```
Me.TreeView1.Nodes(0).Nodes(0).Nodes.Add("MSC CS")
```

```
Me.TreeView1.Nodes(0).Nodes(0).Nodes.Add("MSC IT")
```

```
Me.TreeView1.Nodes(0).Nodes(1).Nodes.Add("PGDCA")
```

```
Me.TreeView1.Nodes(0).Nodes(1).Nodes.Add("DCA")
```

- Count: इसका use nodes को count करने के लिए किया जाता है।

Syntax- Nodes.Count

Ex- MsgBox (TreeView1.Nodes.Count)

- Clear: इसे nodes collection से nodes को एक साथ remove करने के लिए किया जाता है।

Syntax- Nodes.Clear ()

Ex- TreeView1.Nodes.clear ()

- **Remove:** इसका use nodes collection से node को remove करने के लिए किया जाता है इसमें index का use इया जाता है।

Syntax- `Nodes.Remove (index)`

Ex- `Me.TreeView1.Nodes (0).Nodes (1).Remove()`

- **Item:** इसका use nodes collection से किसी item को get करने के लिए किया जाता है।

Syntax- `Nodes.Item (index)`

Ex- `MsgBox (Me.TreeView1.Nodes (0).Nodes.Item (0).Text)`

- **FirstNode/LastNode/PrevNode/NextNode:** इसका use क्रमशः First Node, Last Node, Previous Node और Next Node के get करने के लिए किया जाता है।

`Dim n As TreeNode`

`n = Me.TreeView1.Nodes (0).FirstNode`

`MsgBox (n.Text)`

Toolbar :- इस control के द्वारा vb.net के form में टूलबार create किया जा सकता है।

property:- 1 Apperance:- इसके अंतर्गत 2 value होती है | Normal, Flat normal सेट करने पर toolbar button clear दिखाई देता है | और Flat सेट करने पर button के बीच seperator दिखाई देता है।

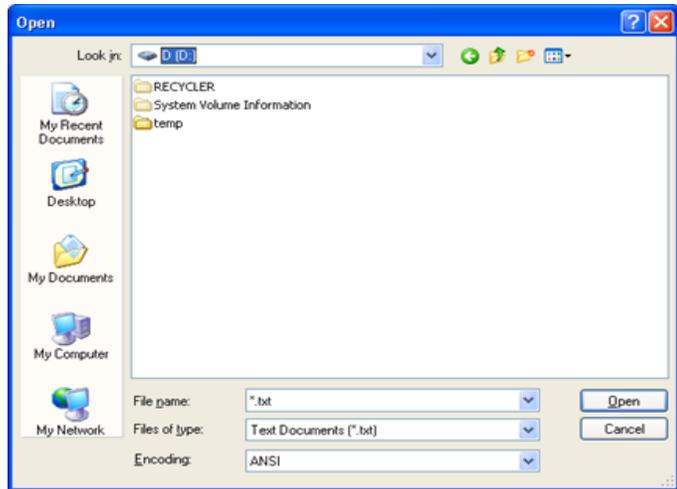
Status Bar:- status bar control का use area के रूप में होता है | जो application window के सबसे नीचे होता है |

तथा जहा application different type के status information को display करता है। status bar control का एक panel होता है। जो text icon अथवा animated icon को रखता है जो process के होने या अन्य सूचना को display करता है इसकी मुख्य property है।

1. **CanOverflow** :- उस value को set करता है जो यह दर्शाता है की status box overflow functionality को support करे |
2. **GripStyle**:-control की position बदलने में उपयोग होने वाले (Grip)की visibility को सेट करता है।
3. **LayoutStyle**:-status bar items के chain को layout करता है |
4. **ShowItemToolTip**:-status bar के लिए mouse को उस tool के ऊपर रखते ही tool tip दिखाया जाता है।
5. **Stretch**:-status bar character पर पूरी फैले या नहीं |

Open File Dialog:-

इसका use open file dialog box को display करने के लिए किया जाता है। यह dialog box user को file select और open की सुविधा प्रदान करता है। यह user को multiple files select करने के सुविधा देता इसमें files और folders की list display है। जिसमें से user files को select कर है और उन्हें process कर सकता है। इसे display के लिए ShowDialog() method का use किया है। यह एक वैल्यू return करती है जो की dialog में click की गई button को बताती है।



करने
है।
होती
सकता
करने
जाता

```
If OpenFileDialog1.ShowDialog = DialogResult.OK Then
```

```
MsgBox (OpenFileDialog1.FileName)
```

```
End If
```

इसमें निम्न property होती हैं।

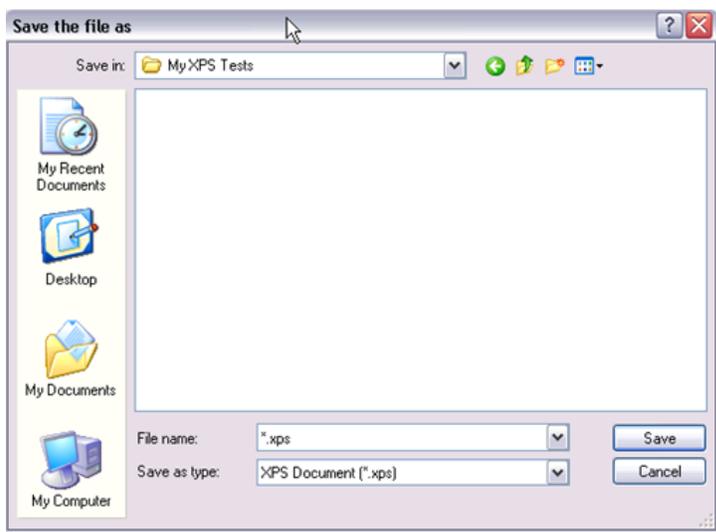
- Name: इसका use control का name define किया जाता है। जो dialog को codes में identify करता है।
- AddExtension: यह property determine करती है की dialogbox FileName में automatically extension add कर लेगा या नहीं यदि use उसे मिटा देता है।
- CheckFileExists: इसके true होने पर DialogBox automatically input किए गए file name को check कर लेगा। File के exist नहीं होने पर यह warning message display करता है।
- CheckPathExists: इसके True होने पर dialogbox File के path को automatically check कर लेगा। Path exist नहीं होने पर यह warning message display करता है।
- DefaultExe: इसका use Default filename extension set करने के लिए किया जाता है। जो की extension नहीं देने पर भी file के साथ automatically add हो जाता है।
- FileName: इसका use Dialog box में select की गई file को और उसके path को get करने के लिए किया जाता है। यह Selected File Name को return करती है।
- FileNames: यह property multiple files को select करने पर सभी files को उनके path के साथ get करने के लिए किया जाता है।
- Filter: यह openfiledialog box की एक important property है। इसका use Files को filter करने के लिए किया जाता है। यह dialog में display होने वाली files के type को specify करने के लिए किया जाता है। जैसे Text files को display करने के लिए Filter में "Text Files|*.txt" set करते हैं। Pipeline (|) character का use file description और Extension को अलग अलग करने के लिए किया जाता है। यह एक साथ कई file extension set करने के लिए semicolon character का use किया जाता है। जैसे - "BMP,GIF and JPG|*.bmp;*.gif;*.jpg"

- **FilterIndex:** इसका use Filter property में define किए गए Filters में default filter को set करने के लिए किया जाता है। यह interger type की वैल्यू contain करती है।
- **InitialDirectory:** इसका use dialog box के लिए initial directory set करने के लिए किया जाता है। जो default open होती है।
- **MultiSelect:** इसके ture होने पर Dialog box में multiple files को select किया जा सकता है।
- **RestoreDirectory:** इसे ture करने पर dialog box previous में open की गई directory को automatically save कर लेता है जो दुबारा open करने पर वही directory open हो जाती है।
- **ShowHelp:** इसके ture होने पर Help button display होती है।
- **Title:** यह property dialog box के title को set करने के लिए किया जाता है।
- **ShowReadOnly:** इसके ture होने पर ReadOnly का एक checkbox भी openfiledialog में display होता है।

Save File Dialog:-

Save dialog box किसी file को save करते समय display होता है। इसका use file को save करने के

लिए location और उसका name set करने की facility देता है। Save पर click करने पर display होने वाले dialog box को Save File Dialog box कहते हैं। चूंकि windows के ज्यादातर applications user को file save करने के लिए prompt करते हैं इसीलिए VB.Net में इसे common dialog control के अंदर रखा गया है। इस dialog box में user file की location और उसके name को set करता है और save पर click करने पर file save हो जाती है। SaveFileDialog control, Common Dialog control category के अंदर आता है। इसे form पर place कर ShowDialog() method को call कर इसे अपने



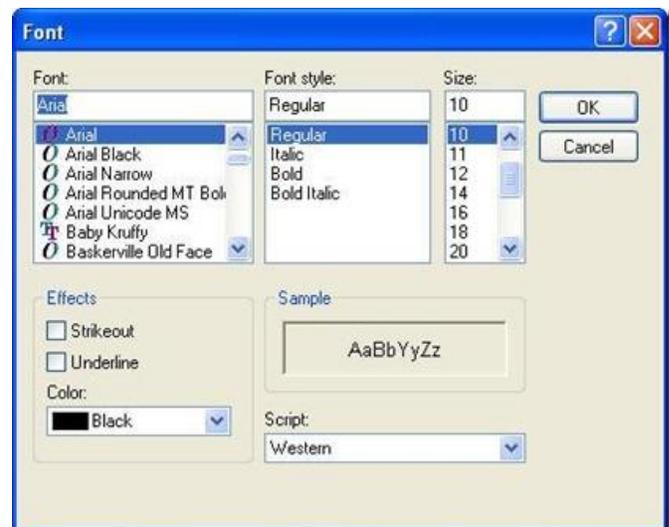
application में display किया जा सकता है। यह dialog control OpenFileDialog की निम्न properties को support करता है।

- **Name:** इसका use control का name define किया जाता है। जो dialog को codes में identify करता है।
- **AddExtension:** यह property determine करती है की dialog box FileName में automatically extension add कर लेगा या नहीं यदि use उसे मिटा देता है।
- **CheckFileExists:** इसके ture होने पर Dialog Box automatically input किए गए file name को check कर लेगा। File के exist नहीं होने पर यह warning message display करता है।

- **CheckPathExists:** इसके True होने पर dialogbox File के path को automatically check कर लेगा। Path exist नहीं होने पर यह warning message display करता है।
- **DefaultExe:** इसका use Default filename extension set करने के लिए किया जाता है। जो की extension नहीं देने पर भी file के साथ automatically add हो जाता है।
- **FileName:** इसका use Dialog box में input की गई file को और उसके path को get करने के लिए किया जाता है। यह Input की गई File Name को return करती है।
- **Filter:** यह Savefile dialog box की एक important property है। इसका use Files को filter करने के लिए किया जाता है। यह dialog में display होने वाली files के type को specify करने के लिए किया जाता है। जैसे Text files को display करने के लिए Filter में "Text Files|.txt" set करते हैं। Pipeline (|) character का use file description और Extension को अलग अलग करने के लिए किया जाता है। यह एक साथ कई file extension set करने के लिए semicolon character का use किया जाता है। जैसे - "BMP,GIF and JPG|.bmp;*.gif;*.jpg"
- **FilterIndex:** इसका use Filter property में define किए गए Filters में default filter को set करने के लिए किया जाता है। यह interger type की वैल्यू contain करती है।
- **InitialDirectory:** इसका use dialog box के लिए initial directory set करने के लिए किया जाता है। जो default open होती है।
- **RestoreDirectory:** इसे true करने पर dialog box previous में open की गई directory को automatically save कर लेता है जो दुबारा open करने पर वही directory open हो जाती है।
- **ShowHelp:** इसके true होने पर Help button display होती है।
- **Title:** यह property dialog box के title को set करने के लिए किया जाता है।

Font Dialog Control:-

इसका use font dialog box को display करने के लिए किया जाता है। font dialog box में Font का name, size, color, type etc. set करने के facility होती है। इन सभी settings को font dialog control का use कर display किया जाता है। यह font से related सभी प्रकार की settings के लिए use किया जाता है। जैसे - bold, italic, underline, script etc. इसे form में place करने के बाद ShowDialog method का use कर इसे dialog box को display किया जाता है। इसकी properties निम्नलिखित हैं।

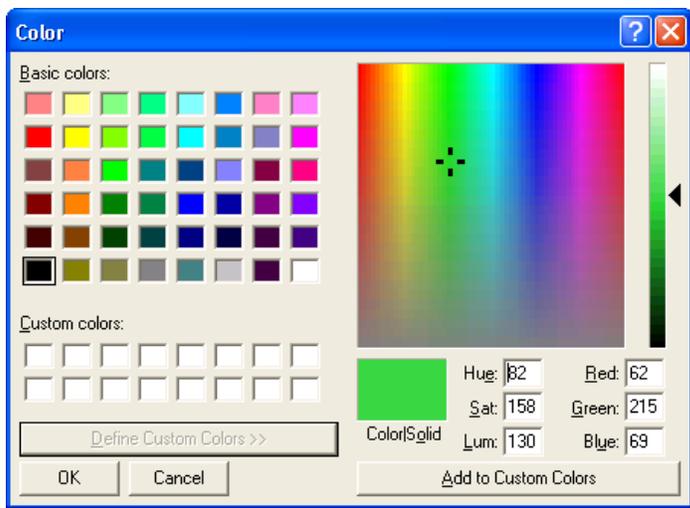


- **Name:** इसका use control का name define किया जाता है। जो dialog को codes में identify करता है।

- **Color:** इसका use font dialog में select किए गए color को get करने के लिए किया जाता है। इसमें colors केवल showcolor property के true होने पर display होते हैं।
- **Font:** यह एक font object है। इसका use font dialog box में set किए गए font को get और set करने के लिए use की जाती है। इसमें font की सभी settings available रहती हैं जैसे- Font name, size, bold, italic, underline etc.
- **FontMustExist:** इसके true होने पर यह dialog को केवल exist fonts को select किया जा सकता है। ऐसे नहीं करने पर warning message display होता है।
- **MaxSize/ MinSize:** यह दोनों property integer values लेती हैं और font की maximum size और minimum size get और set करने के लिए use की जाती है।
- **ShowApply:** इस property के true होने पर Apply button भी font dialog box में display होती है।
- **ShowColor:** इसके true होने पर dialog box में font color select करने के लिए भी option display होता है।

Color Dialog Control:-

Color dialog control का use color dialog box को display करने के लिए किया जाता है। यह dialogbox color को select करने और colors को मिलाकर नए colours बनाने की facility देता है। यह भी dialog controls की category के अंदर Toolbox में पाया जाता है। इसे display करने के लिए ShowDialog control का use किया जाता है। इसमें निम्न properties होती हैं।



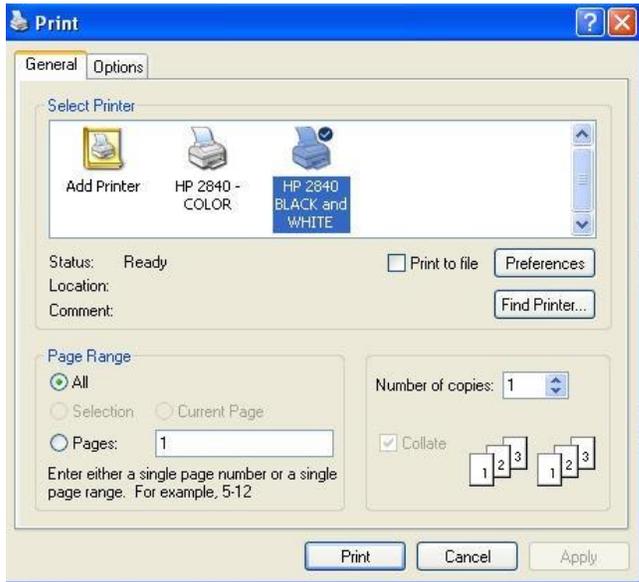
किए गए color को get या set करने के लिए किया जाता है।

- **CustomColors:** इसका use color dialog में custom color को set करने के लिए किया जाता है।
- **SolidColorOnly:** इसके true होने पर user केवल Solid colors को ही use कर सकता है। इसमें custom colors को use नहीं किया जा सकता है।

Print Dialog Control:-

- **Name:** इसका use control का name define किया जाता है। जो dialog को codes में identify करता है।
- **AllowFullOpen:** इसे true करने पर Color dialog को full mode में open किया जा सकता है। इसमें user custom colors भी बना सकता है।
- **AnyColor:** इसके true होने पर color dialog box सभी available colors को display करता है।
- **Color:** इसके use Color dialog में select

इसका use Print Dialog box को display करने के लिए किया जाता है। यह सबसे important dialog control है क्योंकि अधिकतर windows application documents को print करते हैं। अतः printing में इसका use किया जाता है। यह dialog box printing के सभी option automatically provide करता है। इसमें printer, pages, print size, printtype etc. सभी को set किया जा सकता है। इसे use करने से पहले form में place किया जाता है। या runtime में इसे form पर add कर लिया जाता है। इसे show करने के लिए ShowDialog method का use किया जाता है। इसके properties निम्न हैं।



- Name: इसका use control का name define किया जाता है। जो dialog को codes में identify करता है।
- AllowPrintToFile: इसके तुरे होने पर print to file check box enable हो जाता है।
- AllowSelection: इसके property के तुरे होने पर pages select करने के लिए Radio buttons enable हो जाती हैं।
- AllowSomePages: इसके तुरे होने पर user file में से केवल कुछ pages को select करने भी print कर सकता है।

- Documents: इसका use print किए जाने वाले document को set करने के लिए किया जाता है।
- PrintToFile: इसके तुरे होने पर print To file checkbox checked भी रहेगा।
- ShowHelp: इसके तुरे होने पर Print dialog box में help button भी display होती है।
- ShowNetwork: इसके True होने पर Network button display होती है। जिससे network में लगे printer को भी access किया जा सकता है।

Link Label:-

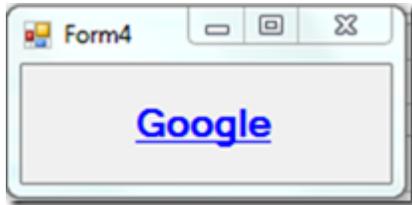
यह Label की तरह का ही होता है लेकिन यह HTML में दी जाने वाली Links की तरह होता है जिसमें click करने पर कोई page open होता है। इसी प्रकार इसमें भी click करने पर यह किसी page से लिंक होती है। इसमें label control की सभी properties use होती हैं इसके अलावा इसमें कुछ अन्य properties भी होती हैं।

- LinkColor: इसका use link का color get और set करने के लिए किया जाता है।
- VisitedLinkColor: इसका use visite की गई Link का color set या get करने के लिए किया जाता है।
- ActiveLinkColor: इसका use activate link का color get या set करने के लिए किया जाता है।

Events: सभी common events इसमें use की जा सकती हैं। इसकी main event LinkClicked होती है जो निम्न है-

- LinkClicked: यह linklabel पर click करने पर perform होती है।

Public Class Form4



```
Private Sub LinkLabel1_LinkClicked(ByVal sender As
System.Object, ByVal e
AsSystem.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked

System.Diagnostics.Process.Start("http://www.google.co.in")
```

End Sub

End Class

Menu Bar:-

vb.net में menu bar आसानी से create किये जा सकते हैं। vb.net की form में विभिन्न प्रकार के menus बनाये जा सकते हैं | form में जब menu bar बनाया जा सकता है तब form में सबसे ऊपर एक menu bar area create हो जाता है। जिसमे menu और sub menu बनाये जा सकते हैं | vb.net में menu bar 2 type बनाये जा सकते हैं।

1. simple menu bar.
2. context menu bar.

Simple Menu Bar :- इसे बनाने के लिए main menu नाम का control form में drag करना पड़ता है ।

Properties :- Checked इसकी दो value होती है True/False इसे true करने पर menu item checked होता है।

Shortcut Key :- इसके द्वारा किसी भी menu item की shortcut key assign की जाती है।

Show shortcut :- by default यह false होता है इसे true करने पर menu के सामने shortcut key दिखाई देता है।

```
Private sub menuitem_Click()

Dim frm as new form2

Frm.show()

End Sub
```

Context Menu/Popup Menu:- window desktop या किसी भी application software में right click करने पर single menu appear होता है। use popup menu कहता है।

popup menu बनाने के लिए context menu के name के control को form में drag करना पड़ता है popup menu creation के बाद run time में form पर right click करने पर context menu दिखाई देता है और form में context menu प्रॉपर्टी पर context menu का नाम सेट करना पड़ता है।

Unit-IV

Object Oriented Programming:-

VB.Net एक Object oriented programming language है इसलिए यह इसके सभी features को support करती है। जो निम्न हैं-

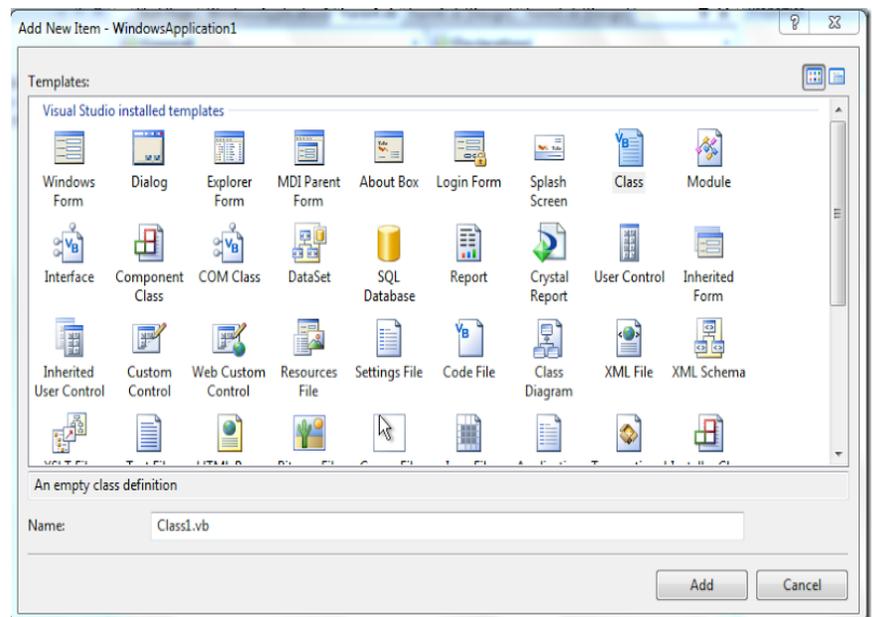
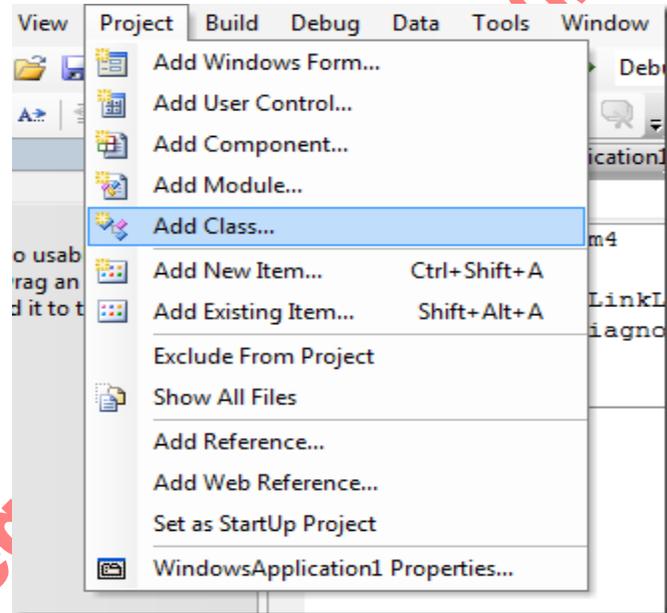
1. Class: class एक user defined data type है जो की data और functions को contain करता है। इसका कोई visible interface नहीं होता है और न ही ये खुद से run हो सकती हैं। VB.Net में Class fields, properties, methods और events को contain करती है।

Building a Class: VB.Net में class बनाने के लिए यह Class Library project provide करता है। जिसका use कर class का collection बनाया जा सकता है। इसके अलावा Windows Application project में भी Class Files को add कर use किया जा सकता है। इसके लिए सर्वप्रथम एक windows application project create कर लेते हैं। इसके बाद Project menu में जाकर Add New Windows Form या Add Class select करते हैं। इसके बाद डिस्प्ले होने वाले New Item Dialog box में Class templet को select करते हैं। और file का नाम enter कर Add button पर click करते हैं। इसके बाद Project में एक class file add हो जाती है और class declare हो जाती है।

Class Declaration Syntax:

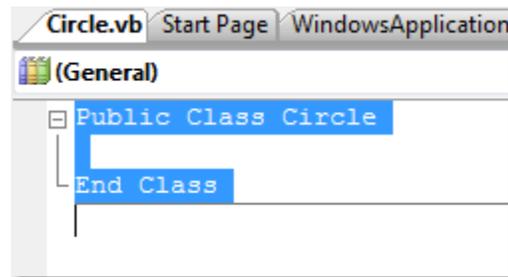
```
[Access Modifier] [MustInherit |
NotInheritable] Class <Class_Name>
    [Inherits
<Base_Class_Name>]
    [Statements]
End Class
```

Access_Modifier: Optional. Can be



one of the following:

- [Public](#)
- [Protected](#)
- [Friend](#)
- [Private](#)
- [Protected Friend](#)



Define

MultiInherit / NotInheritable: Optional,
class can be inherited or Not.

Class: Keyword, Define Class

Class_Name: Name of the Class

Inherits: used when Inheritance is used

Base_Class_Name: Name of base class at inheritance

Features of OOPS:-

1. Object: यह OOPs की basic run-time entity है। जो कि किसी object (person, place, a bank account etc.) को represent करता है। object class का variable है जो कि class को execute करता है और उसमें उपलब्ध methods को use कर डाटा को process करता है। object के create होने पर यह memory में अन्य variables की तरह ही space लेता है।

2. Class : Class एक user defined data type है जो कि data और code को contain करता है जो कि object द्वारा use किया जाता है। class एक structure है जो कि object कि working को define करता है। class को create करने के बाद उसके कई objects बनाए जा सकते हैं। इस प्रकार कहा जा सकता है कि class एक ही प्रकार के objects का collection है।

3. Data Abstraction and Encapsulation: data और functions को एक साथ bind करना Encapsulation कहलाता है। यह class का सबसे important feature है। इसमें डाटा को class के बाहर access नहीं किया जा सकता है केवल class के functions ही इसे access कर सकते हैं। data abstraction बिना background process की details के data को input और output करने से है जिसमें यह Class के functions के द्वारा perform होता है।

4. Inheritance: यह OOPs का एक महत्वपूर्ण feature है जो कि एक class को दूसरी class के features को access करने की facility provide करता है। इसमें एक class का object अन्य class की properties को भी access कर use कर सकता है। यह reuseability के feature को implement करता है जहां किसी class में नए features को add करने के लिए नई class बनाकर उसमें पुरानी class के features को भी implement किया जाता है साथ ही नई class में और codes भी जोड़ दिये जाते हैं।

5. Polymorphism: यह भी OOPs का important concept है जो की एक से अधिक form को बनाकर उसे use करने की facility देता है। यह Function और operators दोनों के द्वारा perform किया जाता है जहां function name या operator वही रहता है पर arguments की संख्या या type या operands के type

Guided by: *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

अलग होने पर अलग प्रकार से use होते हैं और task को perform करते हैं। जैसे + operator numerical addition और strings को जोड़ने के लिये use होता है।

6. Dynamic Binding: OOPs में inheritance और polymorphism इस feature को implement करते हैं जिसमें Binding या linking procedure के call होने पर perform होते हैं। इसे dynamic binding कहते हैं।

7. Message Passing: OOPs में objects परस्पर communicate करते हैं। objects information को send और receive कर communication करते हैं। OOPs में message passing function call के समय perform होता है। जिसमें information function argument के रूप में function के माध्यम से object में input होता है और object की method उसे process कर result generate का देती है।

Classes & Objects:-

Class: Class एक user defined data type है जो कि data और code को contain करता है जो कि object द्वारा use किया जाता है। class एक structure है जो कि object कि working को define करता है। class को create करने के बाद उसके कई objects बनाए जा सकते हैं। इस प्रकार कहा जा सकता है कि class एक ही प्रकार के objects का collection है।

Objects run-time entity हैं जो की class को run करने के लिए use की जाती है। class के variable को ही object कहा जाता है। यह run होकर class के data को store और use class के functions के द्वारा process करता है। इसके create होने पर यह memory में space लेता है और delete होने पर यह space system को वापस हो जाता है। VB.Net में object को declare करने के लिए variable declaration statement का use new keyword किया जाता है।

Syntax

```
Dim <obj_name> As New Class_Name
Or
Dim <Obj_Name> As Class_Name = New Class_Name
Ex:
Dim cir As New Circle
Or
Dim cir As Circle = New Circle
```

Fields:

Classes fields, properties, methods और events से मिलकर बनती है। यह सभी object information को contain करती है। Fields class में use होने वाले variables को कहते हैं। जिसमें value को directly read और write किया जा सकता है। Fields को normal variables की तरह ही declare किया जाता है। इन्हे member variables भी कहते हैं जो कि data को handle करने कि लिए किया जाता है। Previous program में cradius field है।

```
Private cradius As Double
```

Properties:

Properties class में use होने वाली fields की तरह होती हैं। लेकिन इन्हें Get और Set property के द्वारा implement किया जाता है। जो की data को control कर input और output को कंट्रोल किया जाता है। इसमें data को set करते समय उसे verify भी किया जा सकता है।

Set Section data को property में set करते समय use होता है। और Get Section data को property से retrieve करते समय invoke होता है। property में value को pass करते समय उसे Set section में verify किया जा सकता है। data के valid होने पर उसे local field में store का देते हैं। इसी प्रकार data को get करने के लिए उसी variable के द्वारा value return कर दी जाती है। जैसे previous example में Radius एक property है। property को निम्न प्रकार से declare करते हैं।

Syntax:

```
<access_specifier> [ReadOnly] property <property_name> () As <Type>
Get
[Statements]
End Get
Set (Byval Value As <Type>)
[Statements]
End Set
End Property
```

Exp:

```
Public Property Radius() As Double
Get
radius = cradius
End Get
Set(ByVal value As Double)
If value < 0 Then
msgbox("Wrong Input:")
Else
cradius = value
End If
End Set
End Property
```

Methods:

Methods object द्वारा perform किए जाने वाले actions को represent करते हैं। methods functions को ही कहा जाता है। इन्हें किसी special task को perform करने के लिए use किया जाता है। यह किसी भी problem को devide कर उसे easy बना देते हैं।

```
<Access modifier> Function <Function Name> ([arguments]) As <Return Type>
```

```
[Function's statements]
Return <returning value>
End Function
```

Ex:

```
Public Function Area() As Double
Return math.pi * radius ^ 2
End Function
```

Constructors:

किसी object को declare करते समय उसे construct करते समय new keyword का use कर उसका नया instance create किया जाता है। जब किसी class के object को declare करते हैं तब new keyword object का नया instance बनाने के लिए प्रयोग होता है। Constructors ऐसे member functions हैं जो कि किसी भी नए object में fields और properties को initialize करने के लिए use किए जाते हैं। जब object को create करने के लिए new keyword का use करते हैं तब यह automatically call होकर properties को initialize कर देते हैं।

किसी भी class में multiple constructors को भी use किया जा सकता है। जिसमें एक default और कई parameterized constructors हो सकते हैं।

Constructors new subroutine के द्वारा implement किए जाते हैं। यह constructor object के new instance create होने पर automatically call हो जाते हैं। constructor बनाने के लिए Class में कहीं भी new subroutine को define कर किया जाता है। parameterized constructor के लिए उसमें arguments को भी define कर दिया जाता है।

```
Public Class Circle
Public Sub New()
Radius = 0
End Sub
Public Sub New(ByVal R As Double)
Radius = R
End Sub
End Class
```

इस प्रकार object को declare करने पर constructor automatic call हो जाएगा और value initialize हो जाएगी।

```
Dim c As New Circle(10)
```

Inheritance:-

Inheritance VB.Net का एक महत्वपूर्ण feature है जो कि एक class को किसी अन्य class के features को use करने के facility provide करता है। यह पहले से बनी हुई class पर based नई class बनाने की सुविधा देता है। यह reusability को support करता है जहां किसी feature को जोड़ने या पुराने class के features को use करने के लिए उन्हें फिर से नहीं बनाना पड़ता है उन्हें inheritance के द्वारा पुरानी class से inherit कर नई class में add कर लेते हैं। inherit कि जाने वाली class को base-class या parent class और inherit करने वाली class को derived या child class कहते हैं।

Inheriting a Class:

Derived class में base class को inherit करने के लिए Inherits keyword का use कर base class का नाम define कर देते हैं। inherit करने के बाद base class की सभी properties, methods, events, fields और constants को derived class में use किया जा सकता है।

DerivedCircle.vb

```
Public Class DerivedCircle Inherits Circle Public Function Circumference() As
Double Return 2 * math.pi * radius End FunctionEnd Class
```

Form4.vb

Public Class Form4

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Handles Button1.Click Dim dc As New DerivedCircle dc.Radius =
Val(TextBox1.Text) MsgBox("Area = " & dc.Area) MsgBox("Circumfrence =" &
dc.Circumfrence) End Sub

End Class

Private:- किसी event procedure के अन्दर किसी variable के या फंक्शन को सिर्फ उसी procedure event के अन्दर से या अन्दर ही access किया जा सकता है। उस variable का scope local होगा या उस फंक्शन का scope भी local होगा।

Public:- form के अन्दर किसी भी event procedure से हम किसी variable को use कर सकते हैं तो या किसी class के member फंक्शन को किसी procedure में call कर सकते हैं। तो वह scope global या public कहलाता है।

Protected:- ऐसे variable को class तथा child class या किसी अन्य namespace की child class के अन्दर प्रयोग कर सकते हैं।

OVERLOADING:-

Overloading Object oriented programming का एक महत्वपूर्ण feature है जो की किसी procedure के multiple versions बनाने और उन्हें अलग अलग tasks के लिए use करने के सुविधा प्रदान करता है। overloading में procedure का नाम same होता है इसमें केवल parameters में difference होता है जो की उसी procedure को एक ही नाम से अलग अलग प्रकार से task perform करने में हेल्प करता है। इसका उद्देश्य किसी procedure के अलग अलग versions बनाना है जिनके नाम एक ही रहते हैं।

Overloading Rules:

Overloading करते समय निम्न बातों को ध्यान में रखना चाहिए।

- Same Name: सभी overloaded procedures के नाम same होने चाहिए।
- Different Signature: सभी overloaded procedures में निम्न में से कम से कम एक difference होना चाहिए तभी procedures को overload किया जा सकता है-

1. Number of Parameters
2. Order of Parameters
3. Data Types of Parameters
4. Number of Type Parameters
5. Return Type

Ex:

Public Class Shape

Public Function Area(ByVal radius As Double) As Double

Return Math.PI * radius * radius

End Function

Public Function Area(ByVal length As Double, ByVal width As Double) As Double

Guided by: *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

```

Return length * width
End Function
Public Function Area(ByVal a As Integer, ByVal b As Integer, ByVal c As Integer) As
Double
Dim s As Integer
s = (a + b + c) / 2
Return (Math.Sqrt(s * (s - a) * (s - b) * (s - c)))
End Function
End Class

```

MY BASE CLASS:-

Mybase की तरह की यह keyword किसी method को override करने पर derived class की overridden method को कॉल करने के लिए use की जाती है। यह निश्चित कर देती है की call की गई method इसी derived class की overridden method है।

```

Public Overrides Function Area() As Double
Return ((Math.PI * Radius * MyClass.length) + MyBase.Area())
End Function
Limitations of MyClass keyword:

```

- MyClass real object नहीं है इसीलिए इसके द्वारा variable में value को assign, procedure में pass etc. नहीं कर सकते हैं।
- Myclass का use केवल उसी क्लास के members के लिए किया जा सकता है।
- Myclass को standard modules में use नहीं कर सकते हैं।
- Myclass को ऐसे methods के साथ नहीं use किया जा सकता है जो की base class से override नहीं की जा सकती है।

Mybase keyword:-

Mybase keyword inheritance में base class की overridden method को call करने के लिए use की जाती है। जब inheritance करते समय कभी भी base class की overridden method की आवश्यकता पड़ती है तब mybase keyword का use कर use call कर सकते हैं। जैसे given example में जब Circle के area calculate किया जाता है साथ ही जब Cone का area calculate करते हैं तब Circle class के area को override करने के आवश्यकता पड़ती है इसमें base class (Circle) के area method को कॉल करने के लिए mybase का use किया जाता है।

```

Circle.vb
Public Class Circle
Private cradius As Double
Public Property Radius() As Double
Get
radius = cradius
End Get
Set(ByVal value As Double)
If value < 0 Then
msgbox("Wrong Input:")
Else
cradius = value
End If
End Set

```

```
End Property

Public Overridable Function Area() As Double
    Return Math.PI * Radius ^ 2
End Function
```

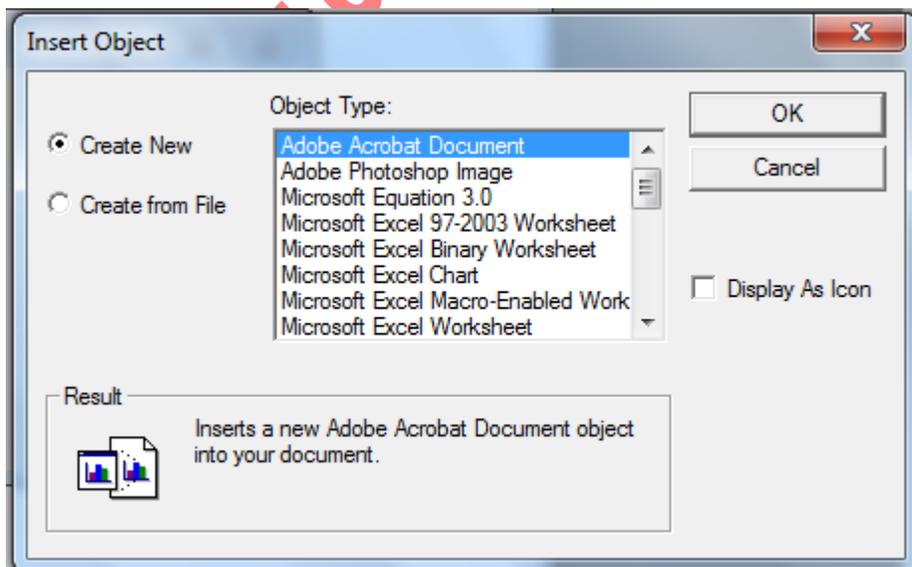
```
End Class
Cone.vb
Public Class Cone
    Inherits Circle
    Dim l As Double
    Public Property length() As Double
        Get
            length = l
        End Get
        Set(ByVal value As Double)
            l = value
        End Set
    End Property
    Public Overrides Function Area() As Double
        Return ((Math.PI * Radius * length) + MyBase.Area())
    End Function
End Class
```

Limitations of MyBase keyword:

- Mybase keyword केवल immediate base class के लिए use की जा सकता है।
- Mybase real object नहीं है इसीलिए इसके द्वारा variable में value को assign, procedure में pass etc. नहीं कर सकते हैं।
- Mybase को MustOverride base class के लिए use नहीं कर सकते हैं।
- Mybase base class के अंदर प्रयोग नहीं की जा सकती है।
- Mybase को friend member के लिए use नहीं किया जा सकता है।

Overview of OLE:-

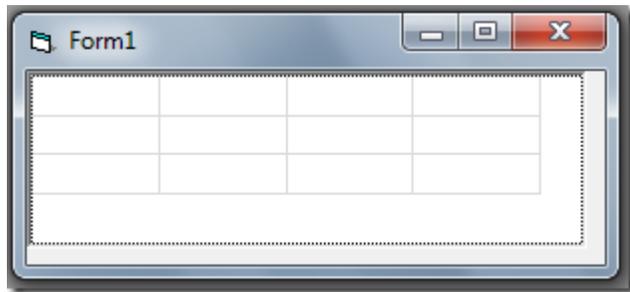
OLE एक technology है जो की एक application के object को दूसरे application से link करने की सुविधा provide करती है। इसका use कर किसी भी application program को अपने प्रोग्राम से link किया जा



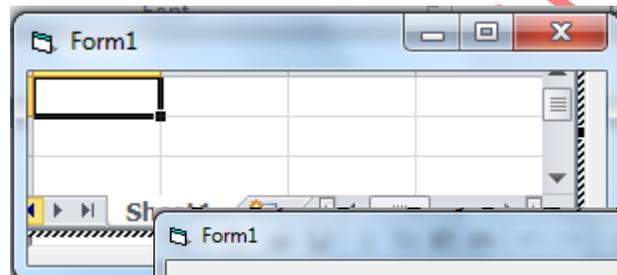
सकता है। यह object oriented programming के feature को implement करते हुए programs object को लिंक करता है। इसे use करने के लिए Toolbox से OLE control को form पर place कर लेते हैं इसके form पर place करते ही यह एक dialog box display होता है जिसमें OLE objects को select किया जा सकता है।

OLE object के लिए किसी application को या किसी पहले से बनी file का use किया जा सकता है। इसके लिए इस dialogbox में Create New और Create from File options होते हैं। नए application के link करने के लिए Create New select कर Object को select करते हैं। और OK button पर click करने पर form पर एक icon या select object बन जाता है। जिस पर double click कर उस application के features को अपने program में access किया जा सकता है।

Ex: Excel sheet select करने पर



Double click करने पर

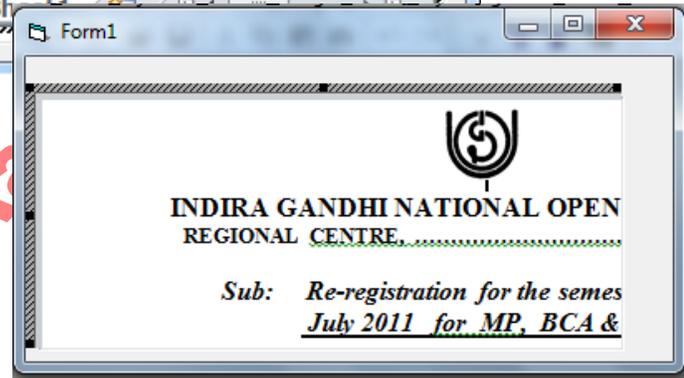


साथ

ही

किसी पहले से बनी फ़ाइल को भी इसके द्वारा link किया जा सकता है इसके लिए create from file select करते हैं और file को select कर Ok button पर click करने पर वह file application से लिंक हो जाती है।

Ex:-



Unit-V

Database programming with ADO.NET:- सभी नई computer technology data को store करने के लिए database application का use करती हैं। जैसे- Microsoft Access, Oracle, SQL Server etc. ये सभी programmes data को store करते हैं। जबकि इस data का use करने वाले सभी application programs किसी न किसी programming language में बनाए जाते हैं। इन्हें use करते समय दोनों application और database program के बीच information के share करने की आवश्यकता पड़ती है इसके लिए Microsoft ADO और ADO.Net की सुविधा प्रदान करता है।

Overview of ADO:- ADO का पूरा नाम ActiveX Data Object है जो की Microsoft Component Object Model (COM) का एक object है जो कि Data source से data access करने के facility provide करता है। यह Microsoft Data Access Component (MDAC) का एक part है जो की programming language और OLEDB Database program के बीच data और information को share करने में help करता है। यह developer को ऐसे codes लिखने में help करता है जिसमें use data कैसे access हो रहा यह जानने की जरूरत नहीं पड़ती है केवल उसे

Guided by: *Abhilash Pathak (8517906324) and Prakash Dwivedi (8982505087)*

connection का ध्यान रखना पड़ता है। ADO के द्वारा डाटा access करने के लिए use SQL के जानकारी के आवश्यकता नहीं पड़ती है। ADO विभिन्न प्रकार के database से data access करने में help करता है। इसका use Visual Basic और अन्य पुरानी programming languages के साथ किया जाता था। यह connection, command, recordset etc. objects के साथ data access provide करता था।

ADO.NET:

ADO.Net .Net framework की data access provide करने वाली library का set है जो की application program और data storage program के बीच data communication provide करता है। .Net framework के आने के बाद Microsoft ने ADO को फिर से define करते हुए उसे .Net के साथ मिलाकर ADO.Net बनाया। जो की सभी .Net applications को Data Source से data को manipulate करने में help करता है। यह Relational (Connected) और Nonrelational (Disconnected) दोनों प्रकार से data access provide करता है। यह मुख्यतः disconnected data access के अनुसार बनाई गई है पर यह connected data access भी provide करती है।

ADO.Net इस प्रकार से design की गई है की यह किसी भी प्रकार के data base से जुड़कर और किसी भी प्रकार से चाहे वह local system, internet या network में कहीं भी हो data communication provide कर सकती है। यह सभी प्रकार के data sources जैसे - Microsoft SQL Server, Oracle, Microsoft Access etc. से SQLDB, OLEDB, ODBC आदि के द्वारा data access प्रदान करता है।

ADO TO ADO.NET:-

ADO के disadvantages को दूर करने के लिए Microsoft से ADO.Net को realese किया। यह दोनों Microsoft की दो विभिन्न data access technology हैं जिनमें निम्न difference हैं-

	ADO	ADO.NET
Business Model	Connection-oriented Models used mostly	Disconnected models are used: Message-like Models.
Disconnected Access	Provided by Record set	Provided by Data Adapter and Data set
XML Support	Limited	Robust Support
Connection Model	Client application needs to be connected always to data-server while working on the data, unless using client-side cursors or a disconnected Record set	Client disconnected as soon as the data is processed. DataSet is disconnected at all times.
	ADO objects communicate	ADO.NET uses XML

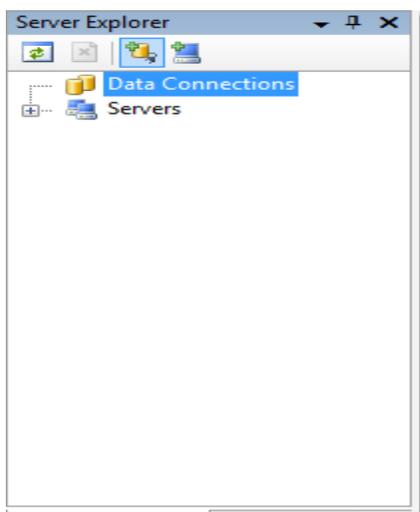
Data Passing

in binary mode.

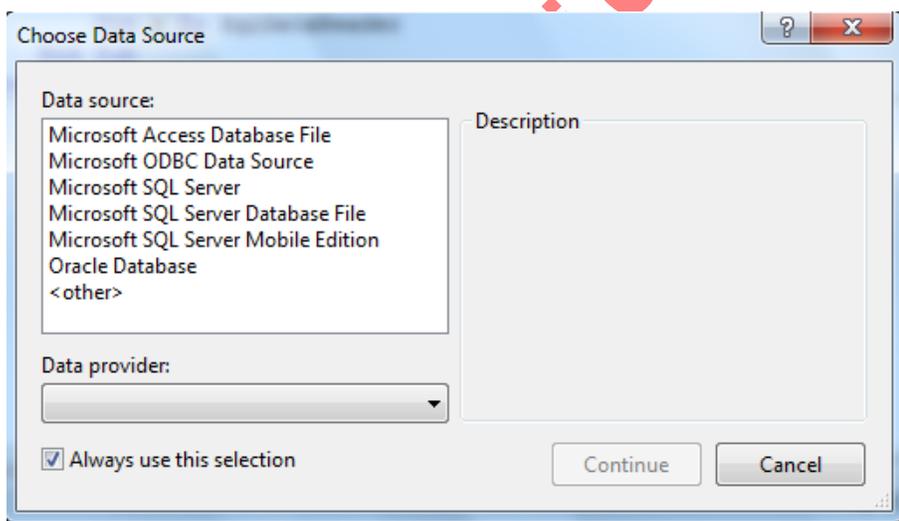
for passing the data.

ACCESSING DATA USING SERVER EXPLORER:-

Server Explorer Visual Studio IDE का एक महत्वपूर्ण window है जो की Database connection करने में help करती है। database को use करने से पहले उसे अपने प्रोग्राम से connect करने की आवश्यकता पड़ती है। यह सुविधा server explorer की सहायता से आसानी से किया जा सकता है। यह विभिन्न प्रकार के databases के साथ connection बनाने में सहायता करता है। इसे display करने के लिए view menu में Server Explorer को select करते हैं या इसे Ctrl+Alt+S key press करके भी display किया जा सकता है।



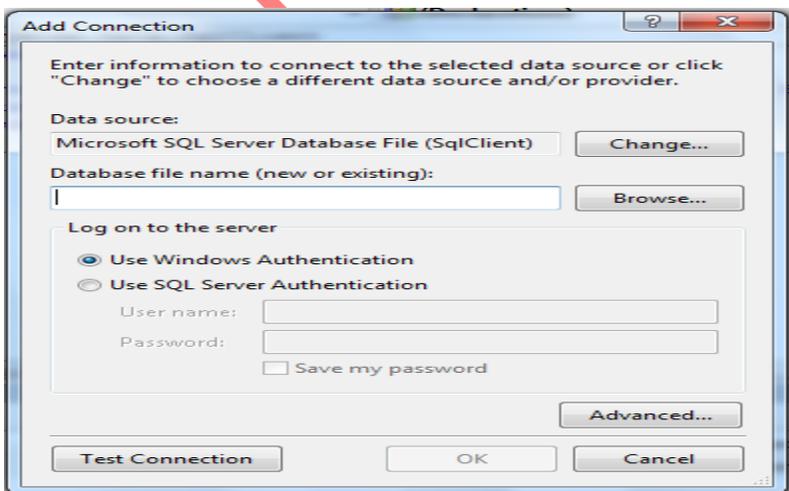
Server Explorer के toolbar में चार button Refresh, delete, connect to database और connect to server display होती हैं। Server



Explorer के द्वारा connection बनाने के लिए connect to database button पर click या

right click कर add connection select करते हैं। ऐसा करने पर Add connection Dialog box या कभी कभी Choose Data Source Window display होने लगती है।

Add Connection Dialog box, Choose Data Source Dialog box में select किए गए Data Source पर depend करती है। fig में Connection Dialog box Microsoft SQL Server Database file select करने पर इस प्रकार display होता है।



Selected data source के अनुसार Database file select करने के बाद Test connection button पर click करते हैं। Connection के OK होने पर OK button पर click

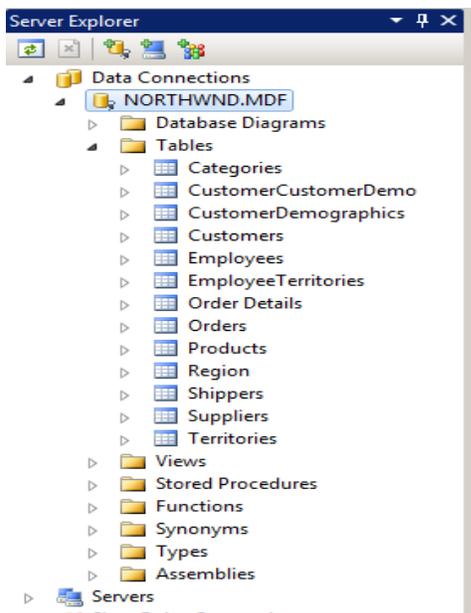
करते हैं। OK पर click करने पर Server Explorer में Selected database की सभी files, tables, views, relations etc. display होने लगते हैं। जहां से उन्हें dataset पर या किसी भी अन्य प्रकार से प्रयोग किया जा सकता है।

Creating Connection:-

इसका use SQL Server को छोड़कर अन्य सभी प्रकार के database से data access करने के लिए किया जाता है। इसमें Microsoft access और oracle प्रमुख हैं। इसका प्रयोग SQL Server के साथ भी किया जा सकता है। इसके लिए System.Data.OleDb namespace को import किया जाता है और इसके बाद सभी objects olddb के साथ जैसे - OleDbConnection, OleDbCommand etc. का use किया जाता है।

यह ADO.Net data provider का important part है जिसका use DataSource और application के बीच communication बनाने के लिए किया जाता है। यह communication path connection string पर depend करता है। connection object, specify किए गए database से connection बनाकर उसे open करता है। एक बार connection के सही तरह से create होने के बाद SQL commands को execute कर desired

operations perform किए जा सकते हैं। सभी tasks के perform होने के बाद इस connection को close भी किया जा सकता है। Connection को open और close करने के लिए connection object क्रमश open() और close() methods provide करता है। साथ ही यह connection string को define करने के लिए connectionString property provide करता है।



Connection: इसमें OleDbConnection object का use किया जाता है। इसे design time में Toolbox और runtime में coding के द्वारा use किया जाता है। इसमें निम्न connection string का use किया जा सकता है।

MS Access:

Provider=Microsoft.Jet.OLEDB.4.0;Data

Source=C:\mydatabase.mdb;Jet OLEDB:Database Password=MyDbPassword;

MYSQL:

Server=myServerAddress;Database=myDataBase;Uid=myUsername;Pwd=myPassword;

Ex: Dim con_str As String = "Provider=Microsoft.ACE.OLEDB.12.0; Data Source = C:\Users\Ashutosh\Documents\Students.accdb"

```

Dim con As New OleDbConnection (con_str)
Try

con.Open()

MsgBox("Connected Sucessfully")

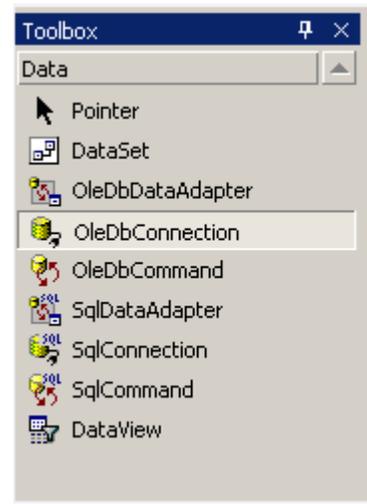
con.Close()

Catch ex As Exception

MsgBox(ex.Message)

End Try

```



Command: इसके लिए OLDDbCommand object का use किया जाता है जो की SQL statements को execute कर result display करता है।

```

Dim con_str As String = "Provider=Microsoft.ACE.OLEDB.12.0; Data Source =
C:\Users\Ashutosh\Documents\Students.accdb"

```

```

Dim con As New OleDbConnection(con_str)
Dim cmd As New OleDbCommand
cmd.CommandText = "Select * form Students"
cmd.Connection = con
Try
con.Open()
Dim dr As OleDbDataReader
dr = cmd.ExecuteReader
dr.Read()
MsgBox(dr.Item(0))
con.Close()
Catch ex As Exception
MsgBox(ex.Message)

End Try

```

DataAdapter और DataSet: इसमें DataAdapter OleDbDataAdapter और dataset normal dataset का object होता है।

```

Dim con_str As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source =
C:\Users\Ashutosh\Documents\Students.accdb"
Dim con As New OleDbConnection(con_str)
Dim ada As OleDbDataAdapter
Dim ds As New DataSet
Try
con.Open()
ada = New OleDbDataAdapter("Select * from Products", con)
ada.Fill(ds)
con.Close()
Dim i As Integer
For i = 0 To ds.Tables(0).Rows.Count - 1
MsgBox(ds.Tables(0).Rows(i).Item(1))
Next
Catch ex As Exception

```

MsgBox (ex.Message)
End Try

Command:-

यह ADO.Net का एक important component है जिसका use SQL Statements को execute करने के लिए किया जाता है। यह connection object में define किए गए data source से SQL statements को execute कर data को manipulate करता है। यह सभी प्रकार की SQL statements को execute करता है। command object connection object का use कर इन statements को execute करता है। इसे use करने के लिए सबसे पहले connection object को define कर उसे open करते हैं। इसके बाद अलग-अलग sql statements को execute किया जाता है। command object कुछ properties और methods contain करती है। जैसे-

- Connection: यह connection को define करने के लिए use की जाती है।
- CommandText: यह SQL command को define करने के लिए use की जाती है।
- CommandType: यह command type यानि text या stored procedure को define करने के लिए use की जाती है।

Methods

- ExecuteReader: यह Select command के execute होने पर retrieve होने वाले डाटा को store करने के लिए use की जाती है।
- ExecuteNonQuery: यह अन्य sql commands जैसे - insert, update और delete के लिए use की जाती है।
- ExecuteScaler: यह single result जैसे - count etc. के लिए use की जाती है।

Data Adapter and Dataset with OleDb and SqlDb:-

• Data Adapter

यह ADO.Net data provider का एक main part है। यह Dataset और DataSource के बीच communication के लिए use किया जाता है। DataAdapter को dataset के साथ use किया जाता है। यह दोनों objects मिलकर data access और data manipulation provide करता है। DataAdapter विभिन्न SQL statements जैसे- Select, Insert, update और delete operation को DataSource में जाकर perform करता है। इसमें Insert, update और delete perform करने पर Data Adapter से जाकर DataSource में changes को store होता है। जबकि Select operation का use Dataset को Fill करने के लिए किया जाता है। DataAdapter dataset के लिए operations perform करता है।

• DataSet

Dataset ADO.Net के disconnected data access model का एक हिस्सा है। यह DataTables collection और DataRelationCollection को contain करता है। यह data के collection को represent करता है जो की Datasource से retrieve किया जाता है। इसे dataadapter के साथ use किया जाता है। यह एक ऐसा object है जिसमें data को एक बार store करने के बाद उसे आसानी से manipulate किया जा सकता है। इसके लिए datasource और connection को open रखने की आवश्यकता नहीं पड़ती है। Dataset बिना data source की जानकारी के work perform करता है जहां यह सभी जानकारी dataadapter के पास होती हैं। dataset request किए गई data की एक copy अपने पास store कर लेता है जो की बाद में datasource में जाकर update हो जाती हैं। यह एक साथ कई tables और उनके relations को contain करता है।

Dataset and Data Adapter Working with OleDb:-

```
Imports System.Data.OleDb
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim connectionString As String
        Dim connection As OleDbConnection
        Dim oledbAdapter As OleDbDataAdapter
        Dim ds As New DataSet
        Dim sql As String
        Dim i As Integer

        connectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=Your mdb
filename;"
        sql = "Your SQL Statement Here"

        connection = New OleDbConnection(connectionString)
        Try
            connection.Open()
            oledbAdapter = New OleDbDataAdapter(sql, connection)
            oledbAdapter.Fill(ds)
            oledbAdapter.Dispose()
            connection.Close()

            For i = 0 To ds.Tables(0).Rows.Count - 1
                MsgBox(ds.Tables(0).Rows(i).Item(0) & " -- " &
ds.Tables(0).Rows(i).Item(1))
            Next
        Catch ex As Exception
            MsgBox("Can not open connection ! ")
        End Try
    End Sub
End Class
```

Dataset and Data Adapter Working with SQLDB:-

```
Imports System.Data.SqlClient
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim connectionString As String
        Dim connection As SqlConnection
        Dim command As SqlCommand
        Dim adapter As New SqlDataAdapter
        Dim ds As New DataSet
        Dim i As Integer
        Dim sql As String
```

```

        connetionString = "Data Source=ServerName;Initial
Catalog=DatabaseName;User ID=UserName;Password=Password"
        sql = "Your SQL Statement Here"

        connection = New SqlConnection(connetionString)

    Try
        connection.Open()
        command = New SqlCommand(sql, connection)
        adapter.SelectCommand = command
        adapter.Fill(ds)
        adapter.Dispose()
        command.Dispose()
        connection.Close()

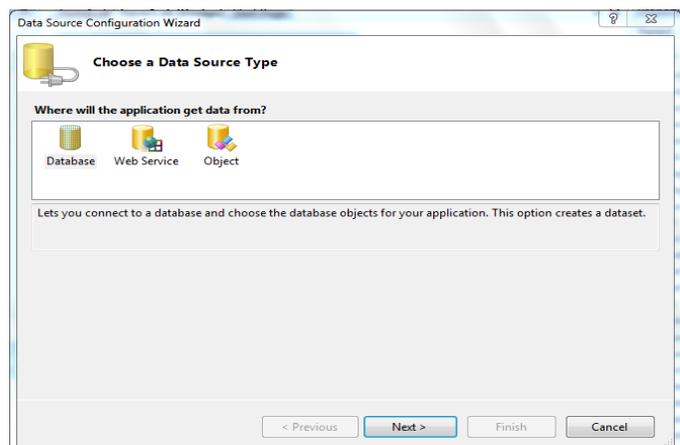
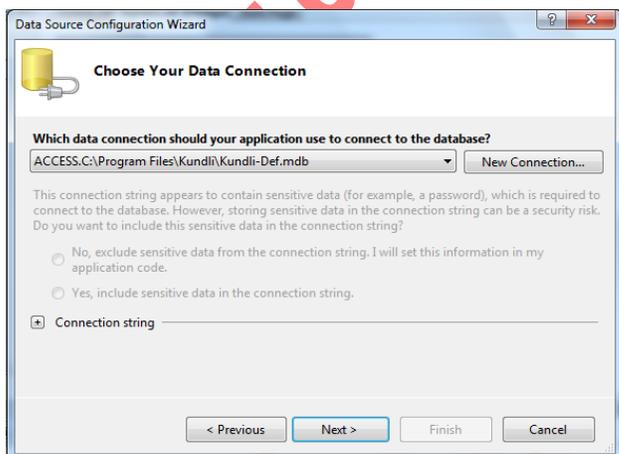
        For i = 0 To ds.Tables(0).Rows.Count - 1
            MsgBox(ds.Tables(0).Rows(i).Item(0) & " -- " &
ds.Tables(0).Rows(i).Item(1))
        Next

    Catch ex As Exception
        MsgBox("Can not open connection ! ")
    End Try
End Sub
End Class

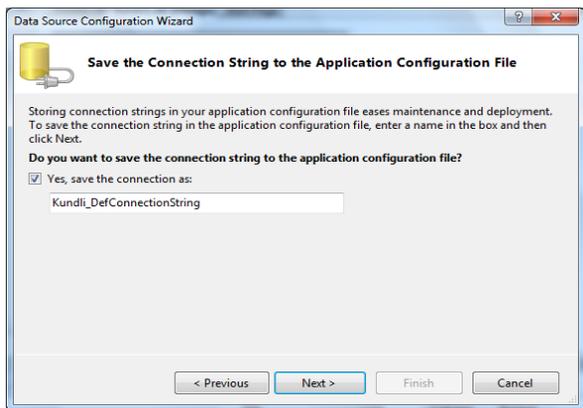
```

Display Data on Data Bound Control:- VB.Net के सभी standard windows control को ADO.Net के द्वारा Database से connect कर data को Display किया जा सकता है। इसके लिए सबसे पहले अपने solution को Data Source से connect कर लेते हैं। इसके लिए निम्न steps follow करते हैं।

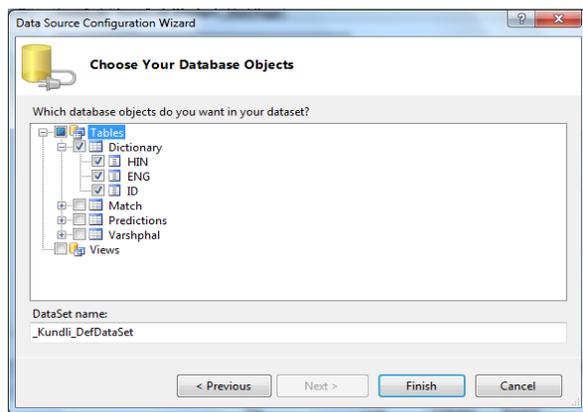
1. Open Visual Studio (Visual Studio 2005)
2. Create a New Windows Application Project
3. Select Data - Add New Data Source
4. In wizard window1: select Database and click to Next
5. In Wizard window2: Select Data connection. You can choose new connction or any Existing connection also. Click Next.
6. Select Connecting String Name. Click Next
7. Select Tables, views etc. and click To Finish



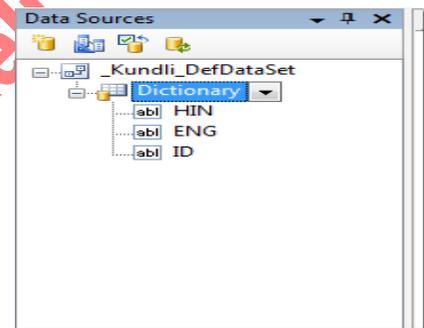
Finish button पर Click करने पर यह एक dataset project मे add कर देता है। जिस पर सभी selected tables, realtions, views etc. को define किया जा सकता है। अब Form पर data को display करने के लिए Form Designer को switch करते हैं। form पर data Grid और lables और Textbox दोनों मे display किया जा सकता है। इसमें data को डिस्प्ले करने के लिए सर्वप्रथम Data menu मे show data source select करते हैं। ऐसा करने पर एक DataSource IDE window display होने लगती है।



control को select कर उसकी DataBinding property मे (Properties Window में) जाकर data column को select कर लेते हैं। इस प्रकार control data से bind हो जाता है।



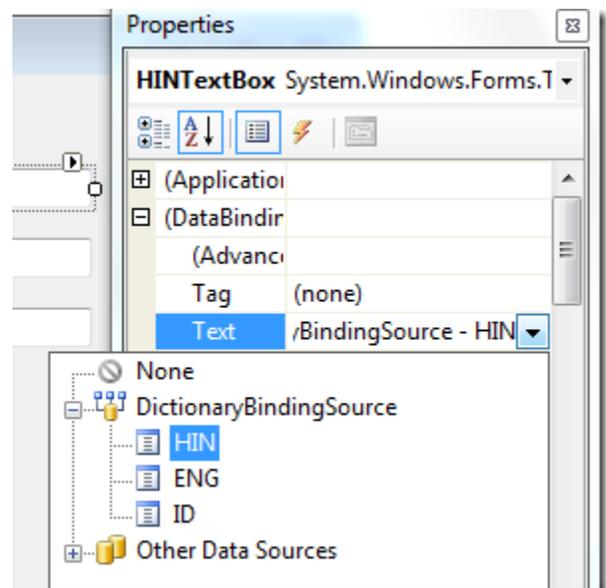
Data को standard controls मे डिस्प्ले करने के लिए सबसे पहले form को design कर लेते हैं। इसके बाद form पर control को select कर उसकी DataBinding property मे (Properties Window में) जाकर data column को select कर लेते हैं। इस प्रकार control data से bind हो जाता है।



इसके अलावा DataSource window से एक एक column को Form पर Drag कर Drop करने पर भी form

automatically design होकर data को display करने लगता

है।



Display data on datagrid

control:- Datagrid डाटा के लिए Visual Interface प्रदान करता है यह display करने का बहुत अच्छा तरीका है और डाटा को edit करने की भी सेवा प्रदान करता है । यह vb.net code के द्वारा प्राप्त किया जाता है । datagrid में ही डाटा को edit किया जा सकता है जो database में जा कर सेव हो जाता है।

```
Imports System.Data.OleDb
```

```
Public Class Form1
```

```
'Change "C:\Users\Jimmy\Documents\Merchandise.accdb" to your database location
```

```
Dim connString As String = "Provider=Microsoft.ACE.OLEDB.12.0; Data
```

```
Source=C:\Users\Jimmy\Documents\Merchandise.accdb"
```

```

Dim MyConn As OleDbConnection
Dim da As OleDbDataAdapter
Dim ds As DataSet
Dim tables As DataTableCollection
Dim source1 As New BindingSource
Imports System.Data.OleDb

```

```
Public Class Form1
```

```
    'Change "C:\Users\Jimmy\Documents\Merchandise.accdb" to your database location
```

```
    Dim connString As String = "Provider=Microsoft.ACE.OLEDB.12.0; Data
```

```
Source=C:\Users\Jimmy\Documents\Merchandise.accdb"
```

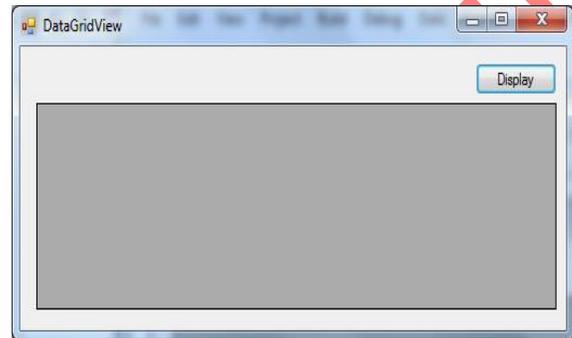
```
    Dim MyConn As OleDbConnection
```

```
    Dim da As OleDbDataAdapter
```

```
    Dim ds As DataSet
```

```
    Dim tables As DataTableCollection
```

```
    Dim source1 As New BindingSource
```



```
Display Button code:
```

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

```
    MyConn = New OleDbConnection
```

```
    MyConn.ConnectionString = connString
```

```
    ds = New DataSet
```

```
    tables = ds.Tables
```

```
    da = New OleDbDataAdapter("Select *
```

```
items]", MyConn) 'Change items to your
```

```
se name
```

```
    da.Fill(ds, "items") 'Change items
```

```
r database name
```

```
    Dim view As New DataView(tables(0))
```

```
    source1.DataSource = view
```

```
    DataGridView1.DataSource = view
```

```
End Sub
```

