

UNIT -3

Visual Basic Object

विजुअल basic ऑब्जेक्ट , कोड और डेटाओं का combination है जो कि एक यूनिट को treat करता है , Object Application का समुह हो सकता है ।

जैसे-कन्ट्रोल और फॉर्म पुरे एप्लीकेशन के आब्जेक्ट हो सकते हैं , जब आप विजुअल बेसिक में कार्य करते हैं तो उसमें Object पहले से Include होता है । विजअल प्रोग्रामिंग भाषा में विभिन्न प्रकार के ऑब्जेक्ट होते हैं

जैसे –Control Forms , Data Access Object ,

विजुअल बेसिक में दूसरे Application के आब्जेक्ट को भी कन्ट्रोल कर सकते हैं , और साथ ही साथ आप अपने आब्जेक्ट का निर्माण कर सकते हैं और उन ऑब्जेक्ट्स के Properties और Method Define कर सकते हैं।

कुछ आब्जेक्ट्स के Example निम्नलिखित हैं:-

Command Button In Visual Basic Object

1.Command Button – विंडोज़ के वातावरण में कार्य करने वाले एप्लीकेशन में Command Button का प्रयोग मुख्य रूप से किया जाता है , प्रयोगकर्ता द्वारा input करने के बाद जब उसे इनका परिणाम अथवा किसी कार्य का Output देखना होता है तब उस command का प्रयोग करना होता है...!!

कमाण्ड बटन के महत्वपूर्ण Properties –

1 – Name Properties

2 – Caption Properties

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

- 3 – Style Properties
- 4 – Enable & Visible Properties
- 5 – Tab Index Properties
- 6 – Tab Stop Properties
- Etc...

Command Button के महत्वपूर्ण Event :-

- 1 – Click() , 2 – KeyUp() , 3 – DragDrop() ,
- 4 – LostFocus() , 5 – DragOver() , 6 – MouseDown() ,
- 7 – Mouse Move() , 8 – MouseUp() , 9 – DragOver() ,
- 10 – GotFocus() , 11 – KeyDown() , 12 – KeyPress() etc...

Command Button के Methods :-

Drag , Move , Refresh , OLE Drag , Set Focus, etc....

Example :-

```
Private Sub Command1_click()  
Msg Box "This is click event "  
End Sub
```

Text Box control

2.Text Box Control – Windows Application बनाने के लिए टेक्स्ट box control का प्रयोग बहुत ज़्यादा होता है । इस कन्ट्रोल का प्रयोग प्रयोगकर्ता द्वारा input लेने के लिए किया जाता है और इसके अलावा प्रोसेसिंग होने के पश्चात् output दिखाने के लिए इसका उपयोग किया जाता है ।

टेक्स्ट बॉक्स की प्रॉपर्टीज:-

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

- | | | |
|---------------------------|--------------------|------------------|
| 1 . Appearance , | 2 . Enable , | 3 . Name , |
| 4 . BackColor , | 5 . BorderStyle , | 6 . DataField , |
| 7 DataMember , | 8 . DataSource , | 9 . Font , |
| 10 . FontBold , | 11 . FontItalic . | 12 FontName |
| 13 . FontSize , | 14 , Fontunderline | 15 . FontColor , |
| 16 . FontStrike Through . | | |

उदाहरण •

```
Private Sub Command1_Click()  
MsgBox (Text1.Text) Text1 . Text = ""  
Text1 . SetFocus  
End Sub
```

Label -

3.Label : सामान्यतः लेबल का प्रयोग टेक्स्ट बॉक्स के बाँए ओर अथवा ऊपर किया जाता है । इसका उपयोग प्रयोगकर्ता को Prompt प्रदर्शित करने के लिये किया जाता है ।

Label के मुख्य प्रॉपर्टीज निम्न हैं :-

- 1 . Name प्रॉपर्टी
- 2 . Caption प्रॉपर्टी
- 3 . Auto size प्रॉपर्टी
- 4 . Border Style प्रॉपर्टी
- 5 . Back Color प्रॉपर्टी
- 6 . Back style प्रॉपर्टी .
- 7 . Font प्रॉपर्टी

Option Button Control

4.Option Button Control :- GUI Application में Option Button Control का प्रयोग Single selection के लिये किया जाता है –

- 1 . Name प्रॉपर्टी
- 2 . Option प्रॉपर्टी
- 3 . Alignment property
- 4 . Style प्रॉपर्टी
- 5 . Value प्रॉपर्टी ।

Option Button के Events :

- 1 – Click() .
- 2 – KeyDown () . .
- 3 – Mouse Move()
- 4 – DbClick()
- 5 . KeyPress ()
- 6 . MouseUp ()
- 7 DragDrops , .
- 8 . KeyUp ()
- 9 . LostFocus ()
- 10 . GotFocus()
11. MouseDown()

Option Button Methods:-

- 1 – Drag
- 2 – Move
- 3 – SetFocus
- 4 – Refresh

Check Box Control

5.Check Box Control :- Check box control का उपयोग Applications में विकल्प को चुनने के लिए किया जाता है । Check box से Multiple choice कर सकते हैं । Check box में किसी Check box को क्लिक करते हैं तो ऑन होने पर 1 – Checked तथा unchecked होने पर 0 तथा अनुपलब्ध होने पर 2 – grayed मान लौटता है ।

Check box control की सामान्यतः प्रयोग होने वाली प्रॉपर्टीज निम्न हैं-

- 1 . Name प्रॉपर्टी ।
- 2 . Option प्रॉपर्टी
- 3 . Alignment प्रॉपर्टी
- 4 . Style प्रॉपर्टी
- 5 . Value प्रॉपर्टी ।

Check Box Control के Events :

- 1 . Click ()
- 2 . KeyPress ()
- 3 . KeyUp ()
- 4 . DragDrop ()
- 5 . DragOyer ()
- 6 . LostFocus ()
- 7 . GOTFocus ()
- 8 . MouseDown ()
- 9 . MouseUp ()
- 10 . MouseMove () इत्यादि ।

Check Box के लिए प्रयोग होने वाले Method निम्नलिखित हैं

- 1 . Drag

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

- 2 .OLEDrag
- 3 . SetFocus
- 4 . Zorder .
- 5 . Move ,
- 6 . Refresh .

उदाहरण:-

```
Private Sub Command1_Click ( )  
If Check1.Value = 1 And Check1.Value = 0 Then  
  
MsgBox ( " 15th August" ) Else  
If Check1 .Vale = 0 And Check2 .Value = 1 Then  
  
MsgBOX ( " 26 January )  
Else  
  
If Check1.Value = 1 And Check2.Value = 1 Then  
MsgBox ( " 15 August" – vbCrLf – "26 January" )  
End If  
  
End If End Sub End Sub
```

Event driven programming :-

Visual basic ऐसी programming language है जिसके द्वारा windows आधारित एप्लीकेशन सॉफ्टवेयर का निर्माण किया जाता है। चूँकि इसके द्वारा user friendly application program तैयार होते हैं जिसमें user output किसी एक माध्यम से नहीं बल्कि विभिन्न माध्यम से प्राप्त कर सकता है।

यह एक ऐसी प्रक्रिया है जिसमें सम्बन्धित कोड तथा controls स्वतः activate नहीं होते हैं यह User के विभिन्न response के द्वारा execute होते हैं।

इस प्रक्रिया में प्रोग्राम के कोड तब तक Idle रहते हैं जब तक कि किसी Event (Button) (Click) इत्यादि कॉल नहीं हो जाता हो अर्थात् system के Hardware तथा software के पारस्परिक मिलन को Event कहा

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

जाता है। इसी प्रमुख features के कारण visual basic को event driven programming कहा जाता है।

Visual basic event :-

Visual Basic में प्रमुख रूप से निम्न Event का प्रयोग किया जाता है-

- Mouse related events
- Keyboard related event
- Memory related events

(a) Mouse Related Events :-

यह ऐसा Event है जिसमें Mouse का उपयोग किया जाता है। Mouse के Right Click, left click अथवा scroll एवं move करके इस events का प्रयोग किया जाता है। यह सर्वाधिक प्रयोग किया जाने वाला Event है।

(b) Keyboard Related Events :-

जैसा की नाम से ज्ञात होता है। इस Event में Keyboard के द्वारा user output प्राप्त कर सकता है जिसमें Key up, Key down, Key press, इत्यादि प्रमुख events है।

(c) Memory Related Events:-

इस event के द्वारा केवल एक ही event का use होता है जिसे हम load event कहते हैं। By default visual basic के form load event पर ही होते हैं। यह ऐसा event है जिसमें अपने आप ही Memory के द्वारा output प्रदर्शित हो जाता है।

Vb.net Constructor:-

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

vb.net Constructor एक स्पेशल मेम्बर method है जिसका प्रयोग क्लास के variables को initialize करने के लिए होता है।

Constructor, class की एक स्पेशल procedure है। हर बार जब class का कोई object create होता है तो class के जितने member वेरिएबल्स हैं उनको initialize करने के लिए उस समय पर constructor अपने आप call होता है।

उसे क्लास का ' फंक्शन ' नहीं बोल सकते क्योंकि वो कुछ return नहीं करता और VB.NET में फंक्शन यानि की method जो कोई वैल्यू return करे ।

Types of vb.net constructor :-

जैसे class के मेम्बर्स (वेरिएबल्स और methods) 2 प्रकार के होते हैं वैसे ही vb.net Constructor के भी 2 प्रकार होते हैं:-

1:- Instance Constructor

2:- Shared Constructor

Instance Constructor:-

Instance वेरिएबल्स को starting वैल्यू असाइन करने के लिए इसका प्रयोग किया जाता है।

VB.NET में constructor को डिफाइन करने के लिए New कीवर्ड का यूज किया जाता है।

vb.net Constructor को डिफाइन करने का example:

Class Student

Private sub1Mark As Integer

Private sub2Mark As Integer

Private sub3Mark As Integer

Private totalMarks As Integer

Public Sub New()

totalMarks = 0

End Sub

Sub Main()

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

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

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

End Sub

End Class

यहाँ पर हमने स्टूडेंट एक्सामिनेशन सिस्टम का उदाहरण लिया है। इस सिस्टम में हमने Student क्लास डिफाइन किया है। हम ऐसा console application बना रहे हैं जिसमें स्टूडेंट्स के अलग अलग सब्जेक्ट के मार्क्स की मदद से टोटल मार्क्स और उसके ऊपर से ग्रेड decide कर सकते हैं।

इस एप्लीकेशन के लिए मैंने totalMarks नाम का एक वेरिएबल डिफाइन किया है। इस वेरिएबल की वैल्यू यूजर enter नहीं करेगा। जब यूजर सारे individual सब्जेक्ट के मार्क्स insert करेंगे तब उनको जोड़ करके totalMarks calculate हो जाएंगे। इसलिए शुरुआत में उसे 0 assign कर सकते हैं। ये काम इस code snippet में constructor का यूज़ करके हो रहा है।

Shared Constructor : Shared Constructor का प्रयोग करके class के shared वेरिएबल्स को initialize किया जा सकता है।

जैसे कि : अगर हम इसी एप्लीकेशन के बारे में सोचें तो हम एक ऐसा वेरिएबल डिफाइन कर सकते हैं जो count करेगा कितने स्टूडेंट्स के मार्क्स एंटर किये गए हैं। इस वेरिएबल को हमें class के shared वेरिएबल के तौर पर define करना पड़ेगा क्योंकि उसकी एक ही कॉपी होनी चाहिए जिससे मालूम होगा कि टोटल कितने स्टूडेंट्स exam में appear हुए थे। इस वेरिएबल को अगर initial वैल्यू असाइन करनी है तो इसको हम following कोड से प्राप्त कर सकते हैं :

Class Student

Private sub1Mark As Integer

Private sub2Mark As Integer

Private sub3Mark As Integer

Private totalMarks As Integer

Public Shared studentCount As Integer

Shared Sub New ()

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

```
studentCount = 0
```

```
End Sub
```

```
Public Sub New()
```

```
studentCount = studentCount + 1
```

```
totalMarks = 0
```

```
End Sub
```

```
Sub Main()
```

```
Dim student1 As Student = New Student ()
```

```
Dim student2 As Student = New Student ()
```

```
End Sub
```

```
End Class
```

यहाँ मैंने एक और constructor define किया है जो shared वैरिएबल को initialize करता है। अभी इस shared constructor की खासियत ये है कि program के execution के टाइम वो सिर्फ एक बार कॉल होता है जब की instance constructor जितनी बार नया ऑब्जेक्ट create करेंगे उतनी बार कॉल होगा। तो इसी लिए मैंने स्टार्टिंग का studentCount है वो 0 सेट कर दिया है। जब भी पहला ऑब्जेक्ट create होगा उस टाइम ये shared constructor सबसे पहले कॉल होगा। उसके बाद instance constructor कॉल होगा। तो उस constructor में मैंने एक लाइन ऐड कर दी है : studentCount = studentCount + 1

इसका use ये है की जितनी बार कोई नया ऑब्जेक्ट create होगा उस टाइम ये constructor कॉल होगा और वो existing studentCount में एक add करेगा।

जब हम पहला ऑब्जेक्ट बनाएंगे Student class का (student1) उस टाइम सबसे पहले Shared Constructor कॉल होगा और studentCount को 0 सेट करेगा। उसके बाद इंस्टैंस constructor कॉल होगा और वो studentCount में एक ऐड करेगा जिससे studentCount = 1 हो जाएगा।

जब हम Main procedure में नया ऑब्जेक्ट बनाएंगे student2 तो उस टाइम सिर्फ इंस्टैंस constructor कॉल होगा क्योंकि shared constructor तो एक बार कॉल हो चुका है। तो यहाँ

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

पर instance variable कॉल होगा और studentCount = 1 + 1 = 2 होगा। इसका मतलब अब तक Student क्लास के 2 ऑब्जेक्ट हमने बनाये हैं जो कि सही है।

Parameterized constructor:-

अभी तक हमने जो constructor देखे उनमें हम कोई पैरामीटर पास नहीं कर रहे थे। उन्हें डिफॉल्ट या तो non-argument constructor भी बोलते हैं।

अगर हमें मेंबर वेरिएबल्स को इनिशियलाइज़ करने के लिए कोई पैरामीटर पास करना है तो हम वो भी कर सकते हैं। जैसे हम किसी method में पैरामीटर पास करते हैं उसी तरह से constructor में भी पैरामीटर्स पास कर सकते हैं। उसे parameterized constructor कहा जाता है। और वो shared या instance दोनों constructors के लिए प्रयोग किया जाता है।

Parameterized constructor का example :

हम एक system बना रहे हैं जिसमें ये decide करते हैं कि employee को सैलरी कितनी देनी है। जैसे की यहाँ पर एक employee के लिए एक fix amount प्रति घंटे के लिए 1000 decide की है।

तो इसको हम क्लास में implement करें तो इस 1000 को हम पैरामीटर के तौर पर ऑब्जेक्ट create करते समय पास कर सकते हैं।

Class Employee

Private employee_Id As Integer

Private hourlyPay As Integer

Public Sub New(ByVal empld As Integer, ByVal amt As Integer)

employee_Id = empld

hourlyPay = amt

End Sub

Sub Main()

Dim employee1 As Employee = New Employee (1201, 1000)
End Sub

End Class

यहाँ हमने Employee क्लास डिफाइन किया है जिसमें अभी के लिए उसका id और hourly rate सेव करते हैं। तो हर employee के लिए अगर मुझे ये वैल्यूज constructor का यूज करके insert करनी है तो मैं उन्हें पैरामीटर्स के तौर पर पास करूँगा जैसे की मुझे employee का id 1201 और hourlyPay 1000 सेट करना है तो मैंने ये वैल्यूज ऑब्जेक्ट create करते समय pass की है ताकि जब employee1 नाम का ऑब्जेक्ट create हो और उसका constructor कॉल हो तब ये values उसमें पास हो। इस तरह से हम parameterized constructor का प्रयोग करके initialization के टाइम पर वैरिएबल्स को वैल्यू असाइन कर सकते हैं।

Access Specifier

ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग में [Data Hiding](#) एक महत्वपूर्ण concept है। हम access specifiers का प्रयोग करके data hiding को प्राप्त कर सकते हैं। access specifiers को access modifiers भी कहते हैं।

C++ में, Access specifiers का प्रयोग करके हम एक class members को accessibility प्रदान करते हैं। जिससे class members को restrict किया जाता है और इन class members को बाहर के functions के द्वारा direct access नहीं किया जा सकता।

Types of access specifiers-

c++ में, access specifiers तीन प्रकार के होते हैं:-

1. Public
2. Private
3. Protected

Public –

- [Class](#) members को public करने के लिए public कीवर्ड का प्रयोग किया जाता है।

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

- public के रूप में declare किये गये data members और member functions को किसी भी दूसरे class और functions के द्वारा access किया जा सकता है.
- class के public members को program में कहीं से भी access किया जा सकता है .
इन्हें dot operator (.) के द्वारा direct access किया जा सकता है.

इसका example –

```
class PublicAccess
{
//public access modifier
public:
int x;          // Data Member Declaration
void display(); // Member Function declaration
}
```

Private –

- private कीवर्ड का प्रयोग class members को private करने के लिए किया जाता है.
- private के रूप में declare किये गये class members को class के अंदर से ही access किया जा सकता है.
- इन्हें class के बाहर किसी function के द्वारा direct access नहीं किया जा सकता.
- class के private data members को केवल member functions या friend functions के द्वारा ही एक्सेस किया जा सकता है.
- अगर कोई private members को access करना चाहेगा तो उसे compile-time error आएगा.

इसका उदाहरण –

```
class PrivateAccess
{
// private access modifier
private:
int x;          // Data Member Declaration
void display(); // Member Function declaration
}
```

Protected –

- protected कीवर्ड का प्रयोग class members को protected करने के लिए किया जाता है.
- protected access modifier भी private access modifier की तरह समान होता है . इन्हें भी class के बाहर access नहीं किया जा सकता है.
- protected के रूप में declare किये गये class members को derived class के द्वारा भी एक्सेस किया जा सकता है.

```
class ProtectedAccess
{
// protected access modifier
protected:
int x;          // Data Member Declaration
void display(); // Member Function declaration
}
```

Note – अगर किसी class के अंदर members के लिए कोई भी access specifier नहीं specify किया गया हो तो default रूप में members का access specifier, **Private** होगा.

Access Specifiers का example –

```
#include <iostream>
using namespace std;
class MyClass {
public: // Public access specifier
int x;
private: // Private access specifier
int y;
};

int main() {
    MyClass myObj;
    myObj.x = 25; // Allowed (x is public)
    myObj.y = 50; // Not allowed (y is private)
    return 0;
}
```

इसका आउटपुट – इसमें compile time error आएगा. क्योंकि इसमें y private है. इसलिए इसे class के बाहर access नहीं किया जा सकता.

Inheritance

Inheritance, VB.Net का एक महत्वपूर्ण फीचर है जो कि एक class को किसी अन्य class के फीचर को प्रयोग करने की सुविधा प्रदान 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 प्रयोग 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।

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।

Inheritance Modifiers

Inheritance के लिए Visual basic .Net कुछ class level modifiers provide करता है।

- NotInheritable: इसका प्रयोग करने पर class को inherit नहीं किया जा सकता है।
- MustInherit: इसका use करने पर class को base class में use करना अनिवार्य हो जाता है। इसे केवल base class की तरह ही use किया जा सकता है।

Overriding Properties and Methods in Derived Class

By default derived class, base class की सभी properties और methods को inherit कर लेती है। कभी-कभी Derived class में inherit की गई property या method को अलग प्रकार से प्रयोग करने की आवश्यकता पड़ती है इसे overriding करना कहते हैं। इसके लिए VB.Net Base class और Derived class दोनों में use होने वाली properties और methods के लिए कुछ modifiers प्रदान करता है जिन्हें प्रयोग कर किसी भी method आदि की overriding को control कर सकते हैं। जो निम्नलिखित हैं।

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

Properties and methods of Base Class

- **Overridable:** इसका use करने पर property या method को override किया जा सकता है।
- **NotOverridable:** इसका use करने पर property या method को override नहीं किया जा सकता है।
- **MustOverride:** इसका use करने पर property या method को override करना अनिवार्य हो जाता है।

Properties and methods of Derived Class

Overrides: यह overridable property या method को override करते समय प्रयोग की जाती है।

VB.net में सभी classes को inherit किया जा सकता है। लेकिन ऐसी सभी classes जिनमें NotInheritable keyword का use किया गया है inherit नहीं की जा सकती हैं।

- Classes को other projects से भी inherit किया जा सकता है।
- VB.Net केवल Single और Multilevel inheritance को support करती है।
- Inheritance करने के लिए Base और Derived class दोनों के access type equal होने चाहिए तभी inheritance possible है।

Inheritance

Inheritance, VB.Net का एक महत्वपूर्ण फीचर है जो कि एक class को किसी अन्य class के फीचर को प्रयोग करने की सुविधा प्रदान 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 प्रयोग 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।

Inheritance Modifiers

Inheritance के लिए Visual basic .Net कुछ class level modifiers provide करता है।

Visual Basic/VB.Net

PAARAS INSTITUTE OF EDUCATION (KASHYAP SIR)

- **NotInheritable:** इसका प्रयोग करने पर class को inherit नहीं किया जा सकता है।
- **MustInherit:** इसका use करने पर class को base class में use करना अनिवार्य हो जाता है। इसे केवल base class की तरह ही use किया जा सकता है।

Overriding Properties and Methods in Derived Class

By default derived class, base class की सभी properties और methods को inherit कर लेती है। कभी-कभी Derived class में inherit की गई property या method को अलग प्रकार से प्रयोग करने की आवश्यकता पड़ती है इसे overriding करना कहते हैं। इसके लिए VB.Net Base class और Derived class दोनों में use होने वाली properties और methods के लिए कुछ modifiers प्रदान करता है जिन्हें प्रयोग कर किसी भी method आदि की overriding को control कर सकते हैं। जो निम्नलिखित हैं।

Properties and methods of Base Class

- **Overridable:** इसका use करने पर property या method को override किया जा सकता है।
- **NotOverridable:** इसका use करने पर property या method को override नहीं किया जा सकता है।
- **MustOverride:** इसका use करने पर property या method को override करना अनिवार्य हो जाता है।

Properties and methods of Derived Class

Overrides: यह overridable property या method को override करते समय प्रयोग की जाती है।

Limitations of Inheritance

VB.net में सभी classes को inherit किया जा सकता है। लेकिन ऐसी सभी classes जिनमें NotInheritable keyword का use किया गया है inherit नहीं की जा सकती हैं।

- Classes को other projects से भी inherit किया जा सकता है।
- VB.Net केवल Single और Multilevel inheritance को support करती है।
- Inheritance करने के लिए Base और Derived class दोनों के access type equal होने चाहिए तभी inheritance possible है।