Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.

2) The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate.

3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).

4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.

5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.

6) In case of some questions credit may be given by judgment on part of examiner of relevant answer based on candidate's understanding.

7) For programming language papers, credit may be given to any other program based on equivalent concept.

Marks

1. (a) Attempt any THREE of the following: 12

(i) Explain any four access specifiers in VB.Net.

(Any four access specifier-1 Mark each)

Ans: Access Specifiers describes as the scope of accessibility of an Object and its members. We can control the scope of the member object of a class using access specifiers. We are using access specifiers for providing security of our applications.

Visual Basic .Net provide five access specifiers, they are as follows:

Public, Private, Protected, Friend and Protected Friend.

1. **Public :**

   Public is the most common access specifier. It can be access from anywhere that means there is no restriction on accessibility. The scope of the accessibility is inside class also in outside the class.

2. **Private :**

The scope of the accessibility is limited only inside the classes in which they are declared. The Private members cannot be accessed outside the class and it is the least permissive access level.

3. **Protected :**

   The scope of accessibility is limited within the class and the classes derived (Inherited)from this class.

4. **Friend :**

   The Friend access specifier can access within the program that contain its declarations and also access within the same assembly level. You can use friend instead of Dim keyword.

5. **ProtectedFriend :**

   ProtectedFriend is same access lebels of both Protected and Friend. It can access anywhere in the same assebly and in the same class also the classes inherited from the same class.

(ii) Explain do while loop in VB.Net with syntax. Give one example.

(Explanation-1 Mark, Syntax - 1Mark, Example- 2 Marks)

Ans: Do Loop:

The do loop can be used to execute a fixed block of statements indefinite number of times. The So loop keeps executing its statements while or until the condition is true. Two keywords, while and until can be used with the do loop. The Do loop also supports an exit Do statement which makes the loop to exit at any moment. The syntax of do loop looks like this;

Syntax:

```
Do while/until condition
Statements
Loop
```

Example on do while loop:

This example shows repeat the loop until condition is true.

```
Private sub Button1_clck(ByVal sender As System. Object, Byval e As
System.EventArgs)_Handles Button1.Click
Dim I As Integer=1
Do while (i<5)
ListBox1.Item.Add("Item" & i)
```

```
I=i+1
Loop
End sub
```

(iii)Explain with example how database connectivity is achieved in ASP.Net using SQL.

(Explanation- 2 Mark, Example -2 Mark)

Ans: The Sql Connection Object is handling the part of physical communication between the application and the SQL Server Database. An instance of the SqlConnection class in .NET Framework is supported the Data Provider for SQL Server Database The four Objects from the .Net Framework provide the functionality of Data Providers in ADO.NET. They are Connection Object, Command Object ,DataReader Object and DataAdapter Object.

Connection Object The Connection Object provides physical connection to the Data Source. Connection Object connects to the specified Database and open a connection between the application and the Database. When the connection is established , SQL Commands may be executed, with the help of the Connection Object, to retrieve or manipulate data in the Database

Data Reader The DataReader Object is a stream-based , forward-only, read-only retrieval of query results from the Data Source, which do not update the data. DataReader Object in ADO.NET is a stream-based , forward-only, read-only retrieval of query results from the Data Source, which do not update the data. The DataReader cannot be created directly from code, they created only by calling the ExecuteReader
method of a Command Object.

DataReader = Command.ExecuteReader() DataReader Object provides a connection oriented data access to the data Sources. A Connection Object can contain only one DataReader at a time and the connection in the DataReader remains open and cannot be used for any other purpose while data is being accessed. When started to read from a DataReader it should always be open and positioned prior to the first record. The Read() method in the DataReader is used to read the rows from DataReader and it always moves forward to a new valid row, if any row exist.

DataReader.Read() There are two types of DataReader in ADO.NET. They are SqlDataReader and the OleDbDataReader. The System.Data.SqlClient and System.Data.OleDb are containing these DataReaders respectively

DataSet stores many DataTables in VB.NET programs. A DataSet is conceptually a set of DataTables and other information about those tables. It is an abstraction that makes programs simpler to develop. This is a container for multiple DataTables. You can use it to create XML. It is a useful abstraction for simplifying programs.

OR

1) Add the namespace

- System.Data
- System.Data.SqlClient.

2) Create connection object using connection parameters like server name, database name user name and password.

3) Create sqlcommand object and pass connection object and sql query as parameters

4) Execute the command using ExecuterReader() method of sqlcommand object. This will return resultset and get it into sqldatareader object.

5) Navigate the resultset.

6) Close the connection object as well as command object to achieve connectivity one can establish connection with the database server by using following command

```
cn = New SqlConnection ("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Documents and Settings\vaishali\My Documents\mydb.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")
```

Once Connection is established then we can fire a query on those databases

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="Default.aspx.vb"
Inherits="empl._Default" %>
<%@ Import Namespace= "System.Data.SqlClient" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div style="height: 174px">
```

```
<%  Dim con As SqlConnection
   Dim cmd As SqlCommand
   Dim dr As SqlDataReader
         con = New SqlConnection("Data
         Source=.\SQLEXPRESS;AttachDbFilename=C:\Documents and Settings\vaishali\My
         Documents\mydb.mdf;Integrated Security=True;Connect
         Timeout=30;User Instance=True")
   con.Open()
   cmd = New SqlCommand("select * from emp", con)
   dr = cmd.ExecuteReader
   While dr.Read()
      Response.Write("<li>")
      Response.Write("empname")
      Response.Write("desg")
      Response.Write("salary")
   End While
   dr.Close()
   con.Close()
%>
</div>
</form>
</body>
</html>
```

***** Note : Other logic for the program also can be**

(iv) List and explain any four Namespaces use in .Net.

(Any four namespaces – 1 Mark each)

Ans:

Sr. No.	**Namespace**	**Description**
1	System.Collections	Contain interfaces and classes that define various collections of objects, such as lists, queues, arrays, hashtables and dictionaries.
2	System.componentModel	Contans classes that are used to implement the runtime and design-time behavior of components and controls.
3	System.Configuration	Contains classes that allow the creation of custom installers for software components.
4	System.Data	Consists mostly of the classws that condtitute the ADO.NET architecture and are used for database connectivity.
5	System.Diagnostics	Contains classes that allow debugging of applications and code tracking.
6	System.Drawing	Contains classes that provide access to GDI+ basic graphics functionality, (more advanced functionality is provided in the: System. Drawing. Drawing.2D System. Drawing.Imaging and System.Drawing.Text namespaces)
7	System.IO	Contains classes that allow synchronous and asynchronous reading from and writing to data stream and files.
8	System.Security	Contains classes that provide access to the underlying structure of the .NET Framework security system.
9	System.Text	Contains classes representing ASCII, Unicode, UTF-7 and UTF-8character encoding, as well as abstract base classes for converting blocks of characters to and from blocks of bytes and more.
10	System.Text.RegularExp	Contains classes that provide access to the .NET

	ressions	Framework regular expression engine.
11	System.Windows.Forms	Contains classes for creating Windows-based applications that full advantage of the rich user-interface features available in the Microsoft Windows operating system. In this namespaces, the form class and many other controls that can be added to forms to create user interfaces
12	System.Xml	Contains classes that provide standards-based support for processing XML.

(v) Explain any four drawbacks of previous languages.

(Any four drawbacks – 1 Mark each)

Ans: **Drawbacks of Previous language(VB)**

The drawbacks are given below:

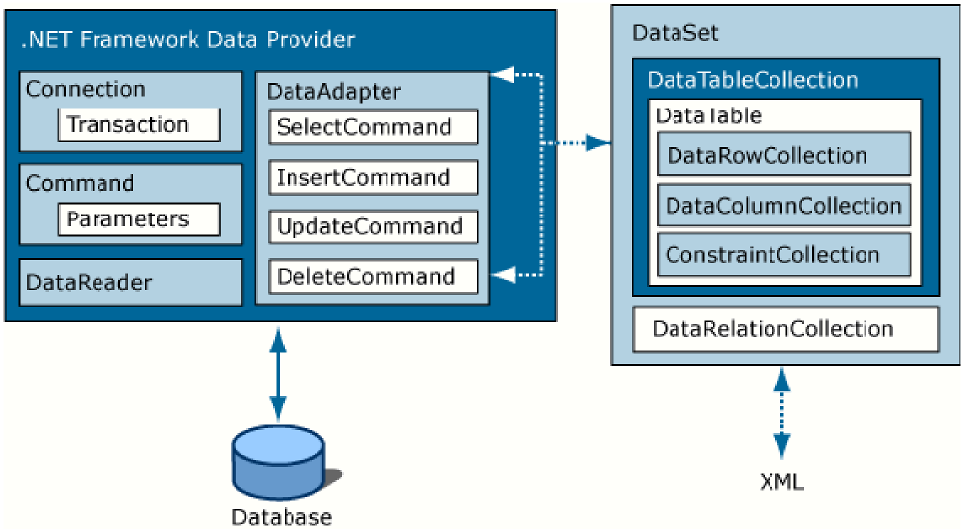
1. VB6 is interpreter based language.
2. VB6 is not a type-safe language.
3. VB6 does not allow developing the multithreaded applications.
4. VB6 is good for desktop windows application. But not so good for web applications, distributed applications.
5. VB6 has less object oriented support.

(b) Attempt any ONE of the following: 6

(i) Draw and explain ADO.Net architecture.

(Diagram -2 Marks and Explanation of all objects of diagram-4 Marks)

Ans:



The architecture of ADO.net in which data retrieved from database can be accessed even when connection to database was closed is called as disconnected architecture.

ADO.NET Components

- There are two components of ADO.NET that you can use to access and manipulate data: .NET Framework data providers the DataSet.

ADO.NET objects

- The four Objects from the .Net Framework provide the functionality of Data Providers in ADO.NET. They are
- **Connection Object:**

  Its provides a connection to the database or data source. The Connection Object provides physical connection to the Data Source. Connection Object connects to the specified Database and open a connection between the application and the Database. When the connection isestablished, SQL Commands may be executed, with the help of the Connection Object, to retrieve or manipulate data in the Database
- **Command Object:**

It is used to execute a command and enables access to database commands to return data, modify data, run stored procedures, and send or retrieve parameter information.

- **DataReader object:**

  It provides a forward-only, read only, connected recordset. The DataReader Object is a stream-based, forward-only, read-only retrieval of query results from the Data Source, which do not update the data. DataReader Object in ADO.NET is a stream-based retrieval of query results from the Data Source, which do not update the data. The DataReader cannot be created directly from code; they created only by calling the ExecuteReader method of a Command Object.

- **DataAdapter object:**

  DataAdapter populates a disconnected DataSet with data and performs update and it provides DataAdapter provides the bridge between the **DataSet**object and the data source. The **DataAdapter**uses**Command**objects to execute SQL commands at the data source to both load the **DataSet**with data, and reconcile changes made to the data in the **DataSet**back to the data source.

.NET Framework Data Providers

- The .NET Framework Data Providers are components that have been explicitly designed for data manipulation and fast, forward-only, read-only access to data. The
- **DataSet**

The ADO.NET DataSet is explicitly designed for data access independent of any data source. As a result, it can be used with multiple and differing data sources, used with XML data, or used to manage data local to the application. The **DataSet**contains a collection of one or more DataTable objects made up of rows and columns of data, as well as primary key, foreign key, constraint, and relation information about the data in the **DataTable**objects. The following diagram illustrates the relationship between a .NET Framework data provider and a **DataSet.**

- **XML and ADO.NET**

ADO.NET leverages the power of XML to provide disconnected access to data. ADO.NET was designed hand-in-hand with the XML classes in the .NET Framework; both are components of a single architecture. ADO.NET and the XML classes in the .NET Framework converge in the **DataSet**object.

(ii) Differentiate between ASP and ASP.Net (Any six points).

(Any 6 point - 1 Mark each)

Ans:

Sr. No.	ASP	ASP.NET
1	ASP uses VBScript	ASP.NET uses languages like C#,VB.NET,etc.
2	It uses interpreter when page is executed	It uses Compiler, i.e code is compiled to MSIL
3	It is not language independent	ASP.NET is a part of .Net framework and so it is language independent.
4	ASP use ADO to connect with database.	ASP.NET uses ADO.NET for database connectivity.
5	ASP does not allow to separate the code from page content.	ASP.NET allows to separate the code from page content by placing all the code in a separate file.
6	User input validation is hard to achieve by code in ASP	ASP.NET provides a set of server controls that checks the user input.
7	Depends on developer to implement own security.	ASP.NET features built-in security configuration in its application setting files. It has two type of Security, viz window based and web based
8	Server controls are not used in ASP.	Server control concept is introduced in ASP.NET
9	ASP does not provide User Input validation.	ASP.NET provides a set of server controls that check whether the information users have entered is in correct from.
10	Event-Driven Programming: This type of event divan programming was not available with classic ASP.	Event-Driven Programming: This type of event divan programming is available with classic ASP.NET.

2. Attempt any FOUR of the following: 16

(a) What is XML? Describe the creation of XML file.

(Explanation of XML – 2 Marks, Creation – 2 Marks)

Ans: XML is a meta-markup language which means that it lets us create our own markup language (our own tags). XML is popular for the following reasons:

1. XML allows easy data exchange.
2. It allows us to customize markup languages.
3. Makes the data in the document self-describing.
4. XML allows for structured and integrated data.

To create XML file:

1) Go to Solution explorer-> right click of project name (eg : windowsapplication1) -> select Add new Item-> select XML file.
2) Provide name of XML file which has an extension as .xml.

Example: Save the following code with a .XML extension.

```
<?xml version="1.0"encoding="UTF-8"?>
<DOCUMENT>
<WELCOME>welcome to XML
</WELCOME>
</DOCUMENT>
```

Read XML is a method used to fill a Dataset with data from XML. It can read data from a file, a stream or an XmlReader.

The syntax of this method is as follows:

Public Function ReadXml(fileName As String) As XmlReadMode

WriteXML is a method used to write the XML representation of the DataSet into a file, a stream, an XmlWriter object or a string.Its syntax is:

Public Sub WriteXml(filename As string)

(b) Write down steps to create Global.asax file.

(Correct Steps - 4 Marks)

Ans: The Global.asax file, sometimes called the ASP.NET application file, provides a way to respond to application or module level events in one central location. You can use this file to implement application security, as well as other tasks. The Global.asax file is in the root application directory. it's actually an optional file. The .asax file extension signals that it's an application file.

The Global.asax file is so configured that any direct HTTP request (via URL) is rejected automatically, so users cannot download or view its contents. The ASP.NET page framework recognizes automatically any changes that are made to the Global.asax file. The framework reboots the application, which includes closing all browser sessions, flushes all state information, and restarts the application domain.

The Global.asax file, which is derived from the HttpApplication class, maintains a pool of HttpApplication objects, and assigns them to applications as needed.

It contains events such as Application_Init, Application_Disposed, Application_Error, Application_Start, Application_End, Application_BeginRequest, Application_EndRequest etc.

Global.asax can be created as

- Create new Web Site/ Web Application.
- In solution explorer, right click on project name -> Add new item->select ‘Global Application class’ option.
- Write required code in the events provided in global.asax.

OR

Creation of global.asaxfile:

1. Create a new web site →File → new →

2. Click on website menu →Add New Item and

Select “Global Application Class” icon from the dialog box and click add button.

```
<%@ Application Language="C#" %>
        voidApplication_Start(object sender, EventArgs e)
  {
  }
```

```
        voidApplication_End(object sender, EventArgs e)
    {
    }
        voidApplication_Error(object sender, EventArgs e)
    {
    }
        voidSession_start(object sender, EventArgs e)
    {
    }
        voidSession_End(object sender, EventArgs e)
    {
    }
</script>
```

(c) What is IIS? State its purpose.

(Explanation – 2 Marks, Purpose – 2 Marks)

Ans: Internet Information Services (IIS) supports a variety of resources that let developers create applications to configure, manage, and extend the functionality of IIS servers and Web applications that run on IIS servers. It use to distribute the website over an internet.

Purpose

- It can be used as a web server.
- It sends duplicate copy of pages over internet.
- It manages multiple websites, web applications, and FTP sites
- Developers can also write programs that filter requests and get the correct Web pages for different users
- It sends only html code to client rather than ASP.net code.
- It also manages all resources.
- It helps in caching of pages and other resources.

(d) Explain use of web.config file.

(Any 4 uses - 1 Mark each)

Ans:

1. Each Web.config file applies configuration settings to the directory that it is in and to all of the child directories below it. Settings in child directories can optionally override or modify settings that are specified in parent directories.
2. Configuration settings in a Web.config file can optionally be applied to individual files or subdirectories by specifying a path in a location element.
3. The root of the ASP.NET configuration hierarchy is the systemroot\Microsoft.NET\Framework\versionNumber\CONFIG\Web.config file, which includes settings that apply to all ASP.NET applications that run a specific version of the .NET Framework. Because each ASP.NET application inherits default configuration settings from the root Web.config file, you need to create Web.config files only for settings that override the default settings.
4. At run time, ASP.NET uses the Web.config files to hierarchically compute a unique collection of configuration settings for each incoming URL request. These settings are calculated only once and then cached on the server. ASP.NET detects any changes to the configuration files and then automatically applies those changes to the affected applications, restarting the applications in most cases.

(e) Explain types of error and what is error provider?

(Types of error - 1 Mark each, Explanation of Error provider - 3 Marks)

Ans: Types of Errors:

Errors can be mainly classified as Syntactical and Logical errors.

Syntactical errors: The mistakes done in writing correct syntax of any statement in computer language causes syntactical error. It is generally identified at the time of compilation of program.

Logical errors: The errors caused by writing wrong logical steps of the program are logical errors. These are generally identified when output of program is examined with all possible inputs.

Error Provider:

The Windows Forms ErrorProvider component is used to validate user input on a form or control. It is typically used in conjunction with validating user input on a form, or displaying errors within a dataset. An error provider is a better alternative than displaying an error message in a message box, because once a message box is dismissed, the error message is no longer visible. TheErrorProvider component displays an error icon (❗) next to the relevant control, such as a text box; when the user positions the mouse pointer over the error icon, a ToolTip appears, showing the error message string.

The ErrorProvider component's key properties are DataSource, ContainerControl, and Icon. The ContainerControl property must be set to the appropriate container (usually the Windows Form) in order for the ErrorProvider component to display an error icon on the form. When the component is added in the designer, the ContainerControl property is set to the containing form; if you add the control in code, you must set it yourself.

The Icon property can be set to a custom error icon instead of the default.

When the DataSourceproperty is set, the ErrorProvider component can display error messages for a dataset.

The key method of the ErrorProvider component is the SetError method, which specifies the error message string and where the error icon should appear.

```
Private Sub TextBox1_Validating(ByVal Sender As Object,_
ByVal e As System.ComponentModel.CancelEventArgs) Handles _
TextBox1.Validating
    If Not IsNumeric(TextBox1.Text) Then
      ErrorProvider1.SetError(TextBox1, "Not a numeric value.")
    Else
      ' Clear the error.
      ErrorProvider1.SetError(TextBox1, "")
    End If
  End Sub
```

(f) Describe CDONTS object with syntax.

(List of objects - 1 Mark, Explanation - 2 Marks, Syntax - 1Mark)

Ans:

1 NewMail object
2 Session object.

1 NewMail Object:

The NewMail object gives you the ability to send a message within page with only few lines of code. The syntax for sending mail with the NewMail object of CDONTS is as follows:

objNewMail.Send([From] [, To] [, Subject] [, Body])

2 Session Object:

The LogonSMTP method initializes the Session object.

objSession.LogonSMTPDisplayName, Address

objSession

Required.The Session object.

DisplayName

Required.String.The display name to use for the messaging user logging on, such as "John".

Address

Required.String. The full e-mail address to use for the messaging user logging on, such as "user@example.com".

3. Attempt any FOUR of the following: 16

(a) Describe following terms:

(i) Methods

(ii) Events

(iii)Collections

(iv)Objects

(Description of each term – 1 Mark)

Ans:

i) Methods

A method created in a Class is nothing more that a Function or a Sub.

```
Public Function Function_name(ByVal a As String) As String
Statemets
Return …
End Function
```

Sub procedures return no values. We implement and call a Sub procedure. Subs optionally haveargument lists and formal parameters. And these formal parameters can be separated with line continuations for readability.

```
Module Module1
Sub Main()
statemets
End Sub
End Module
```

ii) Event

An event is something that happens. Your birthday is an event. So is Christmas. An event in programming terminology is when something special happens. These events are so special that they are built in to the programming language. VB.NET has numerous Events that you can writecode for. And we're going to explore some of them in this section.

Exploring the The Click Event

Buttons have the ability to be clicked on. When you click a button, the event that is fired is the Click Event. If you were to add a new button to a form, and then double clicked it, you would see the following code :

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

This is a Private Subroutine. The name of the Sub is Button1_Click. The Event itself is at the end: Button1.Click. The Handles word means that this Subroutine can Handle the Click Event of Button1.

iii) Collection: - A Collection is similar to a data structure or array; individual items in the collection are accessed via a unique key assigned to that item. These are small properties of controls available in a ASP.NET framework.

For example, In case of Request object ServerVariables is a collection. To execute this collection one can use following command;

Request.ServerVariables (Server environment variable)

iv) Object

Objects are instances of a class. The methods and variables that constitute a class are called members of the class.

Example:

ByVal sender As System.Object (Any example)

(b) How repeaters are used to binding the data on ASP.NET?

(Explanation - 2 Marks, Example - 2 Marks)

Ans: The Repeater control provides two properties to support data binding. To bind data to any collection that implements the System.Collections.

Enumerableinterface (such as System.Data.DataView, a System.Collections.ArrayList,oran array), or the ListSource interface, use the DataSource property to specify the data source. When you set the DataSource property, you must manually write the code to perform data binding. To automatically bind the Repeater control to a data source represented by a data source control, set the DataSourceID property to the ID of the data source control to use. When you set the DataSourceID property, the Repeater control automatically binds to the specified data source control on the first request. Therefore, you do not need to explicitly call the DataBind method unless you have changed data-related properties of the Repeater control.A Repeater control binds its ItemTemplate and Alternating ItemTemplate to either the data model declared and referenced by its DataSource property or the data source control specified by its DataSourceID property. The HeaderTemplate, FooterTemplate, and SeparatorTemplate are not data-bound.If the Repeater control's data source is set but no data is returned, the control renders the HeaderTemplate and FooterTemplate with no items. If the data source is null, the Repeater is not rendered.

Templates

At a minimum, every Repeater control must define an ItemTemplate. However, other optional templates described in the following table can be used to customize the appearance of the list.

Template name	**Description**
ItemTemplate	Defines the content and layout of items within the list.
AlternatingItemTemplate	If defined, determines the content and layout of alternating (zero-based odd-indexed) items.
SeparatorTemplate	If defined, is rendered between items (and alternating items). If not defined, a separator is not rendered.
HeaderTemplate	If defined, determines the content and layout of the list header. If not defined, a header is not rendered.
FooterTemplate	If defined, determines the content and layout of the list footer. If not defined, a footer is not rendered

Example:

```
<%@ Page Language="vb" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>
<script runat="server">
Sub Page_Load(sender As Object, e As EventArgs)
Dim cnn As SqlConnection = New SqlConnection("server=(local);" & _
"database=pubs;Integrated Security=SSPI")
Dim cmd As SqlDataAdapter = New SqlDataAdapter("select * from authors", cnn)
Dim ds As DataSet = New DataSet()
cmd.Fill(ds)
Repeater1.DataSource = ds Repeater1.DataBind()
End Sub
</script>
<html>
<body>
<form id="Form1" method="post" runat="server">
```

```
<asp:Repeater id="Repeater1" runat="server">
<ItemTemplate>
<%# DataBinder.Eval(Container.DataItem,"au_id") %><br>
</ItemTemplate>
</asp:Repeater>
</form></body></html>
```

OR

```
<%@PageLanguage="VB"AutoEventWireup="false"CodeFile="Default.aspx.vb"Inherits="_Def
ault"%>
<!DOCTYPEhtmlPUBLIC"-//W3C//DTD XHTML 1.0
Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@ImportNamespace="System.Data.OleDb"%>
<htmlxmlns="http://www.w3.org/1999/xhtml">
<headrunat="server">
</head>
<scriptrunat="server">
Subpage_load(ByVal sender AsObject, ByVal e AsSystem.EventArgs) HandlesMe.Load
Dim con AsOleDbConnection
DimcmdAsOleDbCommand
DimdrAsOleDbDataReader
con = NewOleDbConnection("Provider =Microsoft.Jet.OLEDB.4.0;Data
Source=c:\database1.mdb")
cmd = NewOleDbCommand("select * from table1", con)
con.Open()
dr = cmd.ExecuteReader()
      rpttable1.DataSource = dr
rpttable1.DataBind()
dr.Close()
con.Close()
EndSub
</script>
```

```
<body>
<formid="form1"runat="server">
<asp:RepeaterID="rpttable1"runat="server">
<ItemTemplate><%#Container.DataItem("FName")%></ItemTemplate>
</asp:Repeater>
<div>
</div>
</form>
</body>
</html>
```

(c) Explain the use of drive object.

(Use – 4 Marks)

Ans: The Drive object is used to return information about a local disk drive or a network share. The Drive object can return information about a drive's type of file system, free space, serial number, volume name, and more.

Note: You cannot return information about a drive's content with the Drive object. For this purpose you will have to use the Folder object.

To work with the properties of the Drive object, you will have to create an instance of the Drive object through the FileSystemObject object. First; create a FileSystemObject object and then instantiate the Drive object through the GetDrive method or the Drives property of the FileSystemObject object.

The Drive object's properties are described below:

Properties

Property	**Description**
AvailableSpace	Returns the amount of available space to a user on a specified drive or network share
DriveLetter	Returns one uppercase letter that identifies the local drive or a network share

DriveType	Returns the type of a specified drive
FileSystem	Returns the file system in use for a specified drive
FreeSpace	Returns the amount of free space to a user on a specified drive or network share
IsReady	Returns true if the specified drive is ready and false if not
Path	Returns an uppercase letter followed by a colon that indicates the path name for a specified drive
RootFolder	Returns a Folder object that represents the root folder of a specified drive
SerialNumber	Returns the serial number of a specified drive
ShareName	Returns the network share name for a specified drive
TotalSize	Returns the total size of a specified drive or network share
VolumeName	Sets or returns the volume name of a specified drive

(d) What is constructor in VB.NET? Give one example.

(Constructor - 2 Marks, Example- 2 Marks)

Ans: A class **constructor** is a special member Sub of a class that is executed whenever we create new objects of that class. A constructor has the name **New** and it does not have any return type.

Following program explains the concept of constructor:

```
Class Line
Private length As Double ' Length of a line
Public Sub New() 'constructor
Console.WriteLine("Object is being created")
End Sub
Public Sub setLength(ByVallenAs Double)
length= len
End Sub
Public Function getLength() As Double
Return length
```

```
End Function
Shared Sub Main()
Dim line As Line = New Line() 'set line length
line.setLength(6.0)
Console.WriteLine("Length of line : {0}",
line.getLength())
Console.ReadKey()
End Sub
End Class
```

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

```
Object is being created
Length of line: 6
```

A default constructor does not have any parameter, but if you need, a constructor can have parameters. Such constructors are called **parameterized constructors**. This technique helps you to assign initial value to an object at the time of its creation as shown in the following

Example:

```
Class Line
Private length As Double ' Length of a line
Public Sub New(ByVallenAs Double) 'parameterised constructor
Console.WriteLine("Object is being created, length = {0}", len)
length= len
End Sub
Public Sub setLength(ByVallenAs Double)
length= len
End Sub
Public Function getLength() As Double
Return length
End Function
Shared Sub Main()
Dim line As Line = New Line(10.0)
Console.WriteLine("Length of line set by constructor : {0}",
line.getLength()) 'set line length
line.setLength(6.0)
Console.WriteLine("Length of line set by setLength : {0}",
line.getLength())
Console.ReadKey()
End Sub
End Class
```

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

```
Object is being created, length = 10
Length of line set by constructor : 10
Length of line set by setLength : 6
```

(e) What two file make up a web form? What is the purpose of each file?

(Name of files - 2 Marks, Purpose - 2 Marks)

Ans: Two files make up a web form are

WebForm1.aspx and

WebForm1.aspx.vb

These two files make up a single Web Forms page.

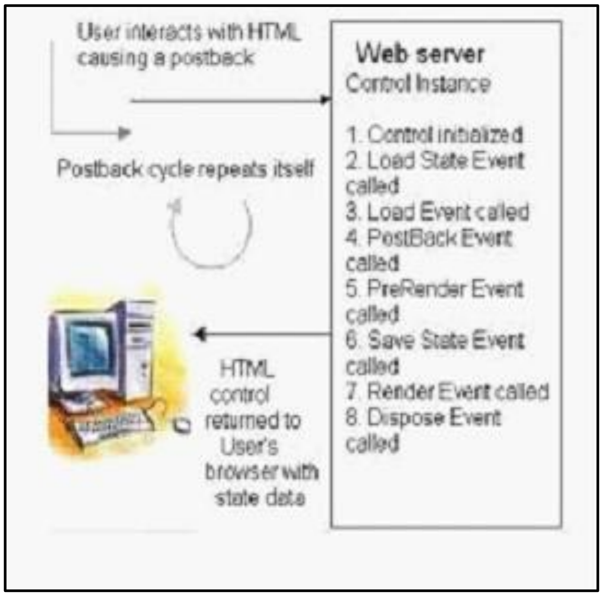
The .aspx file contains the visual elements of the Web Forms page, for example HTML elements and Web Forms controls.

The WebForm1.aspx.vb class file is a hidden, dependent file of WebForm1.aspx. It contains the code-behind class for the Web Forms page, containing event-handler code and the like.

(f) Explain the life cycle of ASP.NET web page with neat diagram.

(Explanation - 2 Marks, Diagram - 2 Marks)

Ans:


User interacts with HTML causing a postback
Postback cycle repeats itself
HTML control returned to User's browser with state data
Web server
Control Instance
1. Control initialized
2. Load State Event called
3. Load Event called
4. PostBack Event called
5. PreRender Event called
6. Save State Event called
7. Render Event called
8. Dispose Event called

When a page is requested, it is loaded into the server memory, processed and sent to the browser. Then it is unloaded from the memory. At each of this steps, methods and events are available, which could be overridden according to the need of the application. In other words, you can write your own code to override the default code.

The Page class creates a hierarchical tree of all the controls on the page. All the components on the page, except the directives are part of this control tree. You can see the control tree by adding trace= "true" to the Page directive. We will cover page directives and tracing under 'directives' and 'error handling'.

The page life cycle phases are:

Initialization

Instantiation of the controls on the page

Restoration and maintenance of the state

Execution of the event handler codes

Page rendering

Understanding the page cycle helps in writing codes for making some specific thing happen at any stage of the page life cycle. It also helps in writing custom controls and initializing them at right time, populate their properties with view-state data and run control behavior code.

Following are the different stages of an ASP.Net page:

Page request .when ASP.Net gets a page request, it decides whether to parse and compile the page or there would be a cached version of the page; accordingly the response is sent

Starting of page life cycle .at this stage, the Request and Response objects are set. If the request is an old request or post back, the IsPostBack property of the page is set to true. The UICulture property of the page is also set.

Page initialization .at this stage, the controls on the page are assigned unique ID by setting the UniqueID property and themes are applied. For a new request postback data is loaded and the control properties are restored to the view-state values.

Page load .at this stage, control properties are set using the view state and control state values**Validation.** Validate method of the validation control is called and if it runs successfully, the IsValid property of the page is set to true.

Postback event handling .if the request is a postback (old request), the related event handler is called.

Page rendering .at this stage, view state for the page and all controls are saved. The page calls the Render method for each control and the output of rendering is written to the OutputStream class of the Page's Response property.

Unload .the rendered page is sent to the client and page properties, such as Response and Request are unloaded and all cleanup done.

4. (a) Attempt any THREE of the following: 12

(i) Explain Transaction database design.

(Explanation - 4 Marks)

Ans: Transaction Database design

For applications transaction

1. CurTrans Rec

2. For trans Rec

Table requires fields like

Transaction Id

Ttype

TDate

TAmount

Authorized by

For What

Toaccount Number

Tansaction Program

Example Or steps

In ASP the ADO Connection Object is used to create and open connection to a data source rather than ADO.NET

If you want to access a database multiple times, you should establish a connection using the connection object. You can also make a connection to a database by passing a connection string via a command or Recordset object. However, this type of connection is only good for single query. SqlTransaction class is used to perform the transaction in ASP.NET SqlTransaction class represents a transact SQL transaction to be made in an SQL SERVER database. SQLTransaction class maintains transaction state information performs, commit and rollback functionality. It catches objects and arrays within the transaction boundary in order to reduce database I/O. It also manages database specific connection and transaction objects within the transaction boundary

(ii) Write a program to get file name and file extension.

(Program – 4 Marks)

Ans:

```
<%@PageLanguage="vb"AutoEventWireup="false"CodeBehind="Default.aspx.vb"Inherits="fileextension._Default"%>
<!DOCTYPEhtmlPUBLIC"-//W3C//DTD XHTML 1.0
Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<htmlxmlns="http://www.w3.org/1999/xhtml">
<headrunat="server">
<title>Untitled Page</title>
</head>
<scriptrunat="server">
SubGet_fileext(ByVal sender AsObject, ByVal e AsSystem.EventArgs)
```

```
Dimfs = Server.CreateObject("Scripting.FileSystemObject")
Response.Write("The file extension is:")
Response.Write(fs.getextensionName("c:\text1.txt"))
fs = Nothing
EndSub
SubGet_filename(ByVal sender AsObject, ByVal e AsSystem.EventArgs)
Dimfs = Server.CreateObject("scripting.FileSystemObject")
Response.Write(fs.GetFileName("c:\hello\text1.txt"))
fs = Nothing
EndSub
</script>
<body>
<formid="form1"runat="server">
<asp:ButtonID="Button1"runat="server"OnClick="Get_fileext"Text="SHOW FILE
EXTENSION"/><br/>
<asp:ButtonID="Button2"runat="server"OnClick="Get_filename"Text="SHOW FILE
NAME"/><br/>
</form>
</body>
</html>
```

(iii)Explain how to fill dataset with multiple tables.

(Explanation – 4 Marks)

Ans: Dataset can hold multiple tables.

Open window form

Drag sqldataadapter1& sqldataadapter2 on the form.

If this control is not present then right click on toolbox, select Choose item…. Option.

When you drag and drop the OleDbDataAdapter control on the form it will open screen

By selecting QueryBuilder option you can build query & check its result.

This will add OleDbConnection1 control on component tray . Repeat the same procedure for another OleDbDataAdapter control, which will also create OleDbConnection2 control on form.

Now right click on OleDbDataAdapter1 control to Generate DataSet

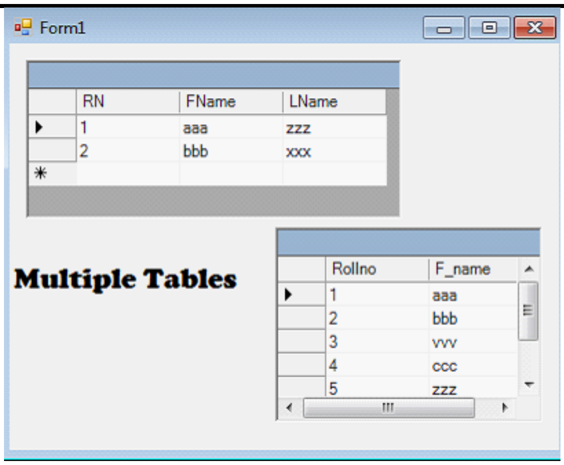
After this step, the DataSet with name DataSet11 control is added on your component tray.

Now write following code on Button1_Click( ) & Button2_Click( ) event

```
Private Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click
DataGrid1.DataSource=DataSet11
DataGrid1.DataMember=”Table1”
OleDbDataAdapter1.Fill(DataSet11)
End Sub
Private Sub Button2_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click
DataGrid2.DataSource=DataSet11
DataGrid2.DataMember=”Table1”
OleDbDataAdapter2.Fill(DataSet11)
End Sub
```

OR

```
PublicClass Form1
PrivateSub Form1_Load(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs) HandlesMyBase.Load
Me.Table1TableAdapter.Fill(Me.Database1DataSet.Table1)
Me.StudentTableAdapter.Fill(Me.StudentDataSet.student)
EndSub
EndClass
```



RN	FName	LName
1	aaa	zzz
2	bbb	xxx

Multiple Tables

Rollno	F_name
1	aaa
2	bbb
3	vvv
4	ccc
5	zzz

(iv) Describe if and select case statement with example.

(if statement with example – 2 Marks, select case statement with example – 2 Marks)

Ans: An **If** statement can be followed by an optional **Else** statement, which executes when the Boolean expression is false.

Syntax:

The syntax of an If...Then... Else statement in VB.Net is as follows:

```
If(boolean_expression)Then
Else
End If
```

If the Boolean expression evaluates to **true**, then the if block of code will be executed, otherwise else block of code will be executed.

Example:

```
Module decisions
Sub Main()
Dim a As Integer = 100
If (a <20) Then
Console.WriteLine("a is less than 20")
Else
Console.WriteLine("a is not less than 20")
End If
Console.WriteLine("value of a is : {0}", a)
Console.ReadLine()
```

```
End Sub
End Module
```

Select Case

A **Select Case** statement allows a variable to be tested for equality against a list of values.

Each value is called a case, and the variable being switched on is checked for each selectcase.

Syntax:

```
Select [ Case] expression
[ Caseexpressionlist
[ statements] ]
[ Case Else
[ else statements ] ]
End Select
```

Where,

Expression: is an expression that must evaluate to any of the elementary data type in VB.Net, i.e., Boolean, Byte, Char, Date, Double, Decimal, Integer, Long, Object, SByte, Short, Single,String, UInteger, ULong, and UShort.

expressionlist: List of expression clauses representing match values for expression. Multiple expression clauses are separated by commas.

Statements: statements following Case that run if the select expression matches any clause inexpressionlist.

Else statements: statements following Case Else that run if the select expression does not match any clause in the *expressionlist*of any of the Case statements.

```
Module decisions
Sub Main() 'local variable definition
Dim grade As Char
grade= "B"
Select grade
Case "A"
Console.WriteLine("Excellent!")
Case "B", "C"
```

```
Console.WriteLine("Well done")
Case "D"
Console.WriteLine("You passed")
Case "F"
Console.WriteLine("Better try again")
Case Else
Console.WriteLine("Invalid grade")
End Select
Console.WriteLine("Your grade is {0}", grade)
Console.ReadLine()
End Sub
End Module
```

(b) Attempt any ONE of the following: 6

(i) Write a ASP.NET form to display information of C: drive.

(Code - 6 Marks)

Ans:

```
<%@PageLanguage="vb"CodeFile="Default.aspx.vb"Inherits="_Default"%>
<htmlxmlns="http://www.w3.org/1999/xhtml">
<headrunat="server">
<title>Untitled Page</title>
</head>
<body>
<formid="form1"runat="server">
<div>
<%
Dimfs = Server.CreateObject("Scripting.FileSystemObject")
DimrootFolder = fs.getfolder(fs.getdrive("c:").rootfolder)
DimsubFolders = rootFolder.subFolders
Dim folder AsObject
Response.Write("<h3> Folders in c:\</h3>")
ForEach folder InsubFolders
```

```
Response.Write(folder.name)
Response.Write("<br>")
Next
Dimsubfile = rootFolder.files
Dim f1 AsObject
Response.Write("<h3> file in c:\</h3>")
ForEach f1 Insubfile
Response.Write(f1.name)
Response.Write("<br>")
Next
%>
</div>
</form></body></html>
```

(ii) Write the steps for binding datagrid control step by step.

(Correct steps -6 Marks)

Ans: DataGridView is very powerful and flexible control for displaying records in a tabular (row-column) form. Here I am describing a different way of databinding with a DataGridView control. Take a windows Form Application -> take a DataGridView control.

Follow the given steps.

Step 1: Select DataGridView control and click at smart property.

Step 2: After clicking, a pop-up window will be open.

Step 3: Click ComboBox.

Step 4: Click at Add Project Data Source. A new window will be opened to choose Data Source Type.

Step 5 : Choose Database (By default it is selected)

Step 6: Select DataSet (By default it is selected)

Step 7: Write Server name, User name and Password of your SQL server and select Database name.

Step 08: Click on Table to explore all tables of your Database.

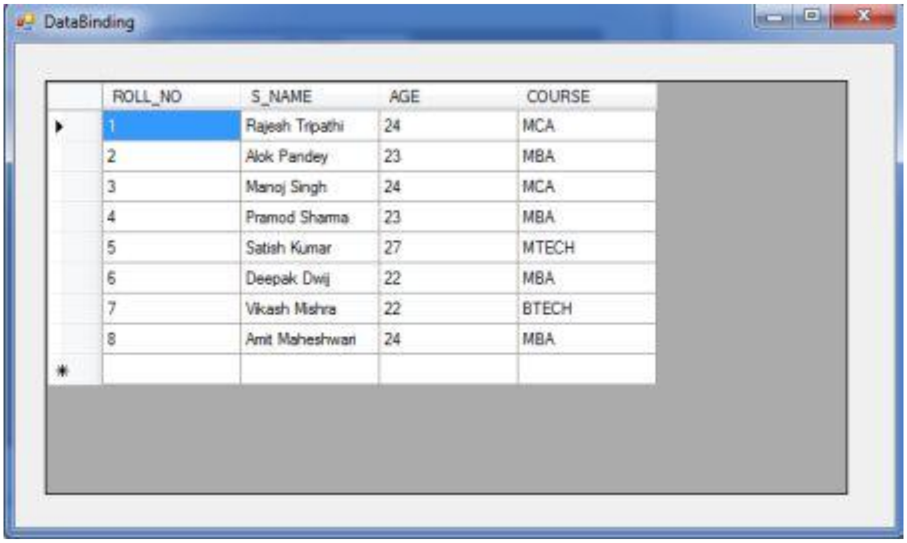
Step 09: Click on the selected Database table to explore all columns.

Step 10: Check the CheckBox to select columns.

Step 15: DataGridView will show all columns of the table (Here, "Student_detail").

Run the application.

Output



	ROLL_NO	S_NAME	AGE	COURSE
▸	1	Rajesh Tripathi	24	MCA
	2	Alok Pandey	23	MBA
	3	Manoj Singh	24	MCA
	4	Pramod Sharma	23	MBA
	5	Satish Kumar	27	MTECH
	6	Deepak Dwij	22	MBA
	7	Vikash Mishra	22	BTECH
	8	Amit Maheshwari	24	MBA
*				

Now we bind the DataGridView with database by code. Take another DataGridView control and write the following code on the form load event.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DatabindingWithdataGridView
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        SqlDataAdapter dadapter;
        DataSet dset;
        string connstring = "server=.;database=student;user=sa;password=wintellect";
        private void Form1_Load(object sender, EventArgs e)
        {
            dadapter = new SqlDataAdapter("select * from student_detail", connstring);
            dset = new System.Data.DataSet();
            dadapter.Fill(dset);
            dataGridView1.DataSource = dset.Tables[0].DefaultView;
        }
    }
}
```

5. Attempt any FOUR of the following: 16

(a) Explain interface with example.

(Explanation – 2 Marks, Example – 2 Marks)

Ans: Interfaces provide a way of implementing polymorphism, throughinterfaces; you can specify methods that a component must implementwithout actually specifying how the method is implemented. Interfaces include abstract methods, meaning you don’t need to provide code for a method, only declare the method and code it wherever it is required. VB.net does not support multiple inheritances directly so interface is a solution for that. ‘Interface’ keyword can be used to create interface and ‘Implements’ keyword is to implement the interface.

Syntax:

```
Interface <name>
Body of the interface
End Interface
```

Example:

```
Module Module1
Sub Main()
    Dim a as new Test
    a.add()
Console.ReadLine()
End Sub
Interface Addnum
    Sub add()
End Interface
Public class test
     Implements Addnum
Dim x as Integer=5
Dim y as Integer=50
Public sub add() Implements Addnum.add
    Console.WriteLine(“Sum=   “ & (i+j))
End sub
```

(b) Write VB.NET application using command button and textbox to read number & calculate factorial of the number.

(Correct Logic – 2 Marks, Correct Syntaxes – 2 Marks)

Ans:

```
Public Class Form1
    Dim f As Integer = 1
    Dim i As Integer = 1
    Dim n As Integer = 0
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        f = 1
        n = CInt(TextBox1.Text)
        For i = 1 To n
            f = f * i
        Next
        TextBox2.Text = f
    End Sub
End Class
```

(c) Explain the purpose of data adapter control. How does a data adapter differ from a connection?

(Data adapter explanation – 3 Marks, Use of Connection and how it differs from data adapter – 1 Mark)

Ans: The DataAdapter is a part of ADO.NET Data Provider & it is the class at the core of ADO.NET's disconnected data access. DataAdapter provides the communication between the Dataset and Datasource.

Adapters are used to exchange data between a data source and a dataset. In many applications, this means reading data from a database into a dataset, and then writing changed data from the dataset back to the database. However, a data adapter can move data between any source and a dataset.

DataAdapter uses select statement to fill a dataset and uses other three sqlstatements(insert, update, delete) to transmit changes back to the database.

Visual Studio makes these data adapters available for use with databases:

•The OleDbDataAdapter object is suitable for use with any data source exposed by an OLE DB provider.

•The SqlDataAdapter object is specific to SQL Server. Because it does not have to go through an OLE DB layer, it is faster than the OleDbDataAdapter class.

Many a times you don't require to create explicit connection to get connected to the database, but sometimes you require to create a connection object explicitly.

Difference between Connection and Data Adapter is as follows:

Connection object is necessary to interact with a database. The connection helps to identify the database server, the database name, user name, password and other parameters are necessary for connecting to a database.

Whereas Data Adapters are used to fill dataset and to read and write data to databases

(d) Create a web project using ASP.NET to display 10 pages on web document using content linker. Also describe content linking list file.

(Content linking explanation – 2 Marks, Program – 2 Marks)

Ans: The content linking component manages a list of URLs so that you can treat the pages in your web site like the pages in a book. The content Linking Component can be used to automatically generate and update tables of contents and navigational links to previous and subsequent web pages. This is ideal for application such as online newspapers and forum messages listings. Structure Of Content Linking Component: - The Content linker has two parts:

The Index file: An ASCII text file that holds a list of .asp pages in the order that they should be presented to the viewer. The index file can easily be changed to add new pages, revise the order of pages or substitute newer pages for older pages

The ASPpages: - Each .asp file contains two parts. The first part consists of the normal content of the page, such as the text and graphics. The second part is the ASP code that uses the content Linker to give the user options for the next, first, previous and last page.

Content Linking Index File: - The content linking component depends on an external file kept on the server for its list of navigable content. An index file will be text file that looks like as follows

```
 HTMLPage.html       HTMLPage1              Intro
 HTMLPage2.html      HTMLPage2              Variables
 HTMLPage3.html      HTMLPage3              Procedures
```

And so on for 10 pages.

Each line in the above file includes three things The relative URL of the content page The Display text A Comment describing the content

Code in Default.aspx file

```
<%@Page Language=”VB” AutoEventWireup=”false” CodeFile= “Default.aspx.vb”
Indherits=”_Default” %><html xmlns=”http://www.w3.org/1999/xhtml”>
<head runat=”server”><title>Untitled Page</title></head><script runat=”server”>
<body>
        <Form id=”forma”runat=”server”>
                <h3>Contents List :</h3>
        <ul>
<%Dim nl = Server.createObject(“MSWC.NextLink”)
        Dim intCount = nl.GetListCount(“textFile.txt”)
        Dim intLoop
        For intLoop = 1 To intCount%>
<li>
        <a href =”<% = nl.GetNthURL(“TextFile.txt” , intLoop)%>”>
        <% =n=nl.GetNthDeascription(“TextFile.txt”,intLoop)%>
        </a>
</li>
<% Next%>
</ul>
</form></body></html>
```

(e) Describe form view and data repeater control.

(Form View Control – 2 Marks, Data Repeater – 2 Marks)

Ans: 1) Form view control:

Formview is a data bound user interface control that render a single record at a time from its associated data source, optionally providing paging buttons to navigate between records.

Binding to data source control, such as SqlDataSource and ObjectDataSource.

The Features of FormView are as follows:

1. Buit-in inserting,updating and deleting capabilities.

2. Buit-in paging capabilities.

3. Customizable appearance through user-defined templates, themes and styles.

Example:

Default.aspx

```
<%@Page Language=”VB” CodeFile=”Default.aspx.vb”Inherits=”_Default”%>

<html xmlns=http://www.w3.org/1999/xhtml>

<head runat=”server”><title>Unititled page</titlex/head>

<body>

<form id=”Form1”runat=”server”>

<asp:Lable ID =”lable1”runat=”server”text=”FormViewConrol”/xbr/><br/>

<asp:FormViewID=”FormView1”runnat=”server” Allowpaging=”True”BorderColor=”Black”

BorderWidth=”1px” dataKeyNames=”Student_no”DtaSourceID=”sqlDataSource1”>

<insertItemTemplate.

        Student_no:>asp:TextBoxID=”TextBoxID”runat=”server”Text=’<%#Eval(Employee_ID”
        )%’/><br/>

        Student_name:<asp:TextBoxID=”TextBox2”runat=”server
        text=’<%#Bind(“Emplyee_name”)%>’/><br/>

<insertitemtemplate>
```

```
<itemTemplate>
        Student_no:<asp:Lable ID=”Lable3 runat=”server” text=’<%# Eval (“Student_no”)%>’/><br/>
        Student_name:<asp:Lable ID=”Lable4”runat =server” text=’<%# Bind (“Student_no”)%>’/><br/>
</Item Template>
</asp:FromView>
<asp:sqlDataSource ID=”sqldataSource1” runat=”Server” ConnectionString=”dtaSource=.;intial catalog =northwind;user ID=sa;password=sa” SelectCommand=”SELECT *FROM Student”providername=”System.Data.Squlclient”/>
</from></body></html>
```

2) Repeater control

- The repeater control displays data items in a repeating list. Similar to DataList, the content and layout of list items in repeater is defined using templates.
- At a minimum, every repeater must define an item Template: Unlike DataList, reapeater has no built-in layout or styles.
- You must explicitly declare all HTML layout, formatting and style tags within the templates of the control.

Example:

```
<html>
<body>
<form id=”Form1”runat=”server”>
<asp:Repeater ID=”rptStudents”Runat=”Server”>
<Item Template><%#Container.DataItem(“Student_name”)%></Item Template>
</asp:Reapeater>
</form></body></html>
```

(f) What is multithreading? Also describe the synchronization of threads.

(Multithreading – 1 Mark, Synchronization – 3 Marks)

Ans: Multithreading

It is a feature provided by the operating system that enables your application to have more than one execution path at the same time. We are all used to Windows' multitasking abilities, which allow us to execute more than one application at the same time. While writing in Microsoft Word, listening to favorite songs in WinAmp and downloading a new song using the Internet Download Manager and document is sent for printing. In a similar manner, we may use multithreading to run different methods of our program at the same time. The .NET Framework, and thus VB.NET provides full support for multiple execution threads in a program. You can add threading functionality to your application by using the System.Threading namespace. A thread in .NET is represented by the System.Threading.Thread class. We can create multiple threads in our program by creating multiple instances (objects) of this class. A thread starts its execution by calling the specified method and terminates when the execution of that method gets completed. We can specify the method name that the thread will call when it starts by passing a delegate of the ThreadStart type in the Thread class constructor. The delegate System.Threading.ThreadStart may reference any method which has the void return type and which takes no arguments.

Synchronization of Threads:

- There may be a situation when two threads are working with the same data and you might not want the second thread to work with that data until the first thread is finished with it.
- So in this situation synchronization is necessary. Synchronizationcan be achieved by, SyncLock statement and join method. SyncLock gains an exclusive lock to be an object reference that is passed to it.
- By gaining this exclusive lock you can ensure that multiple threads are not accessing shared data or that the code is executing on multiple threads.
- This code will provide the locking functionality:

Program:

```
Imports System

Imports System.Threading

Imports system.data
Imports system.text
Module module1
Dim text as new stringbuilder()
Public sub main()
Dim firstThread As New Thread(New ThreadStart(AddressOf Fun1))
 Dim secondThread As New Thread(New ThreadStart(AddressOf Fun2))
firstThread.Start()
secondThread.Start()
firstThread.join()
secondThread.join()
console.writeline("text is:{0}{1}",vbCrlf,text.tostring()))
consolr.read()
End Sub
Public Sub Fun1()
 Dim i As Integer
Synclock text
 For i = 1 To 20
Thread.sleep(10)
Text.append(i.tostring()+" ")
Next
End Synclock
 End Sub
Public Sub Fun2()
 Dim i As Integer
Synclock text
```

```
 For i =2 1 To 30

Thread.sleep(2)

Text.append(i.tostring()+" ")

Next

End Synclock

 End Sub

End Module
```

6. Attempt any TWO of the following: 16

(a) Describe with example, how database connectivity can be achieved in VB.NET using SQL and MS-Access with Data Set and Data Reader. Write code to display contents of database.

(Explanation – 4 Marks, Example using Data set – 2 Marks, Using Data reader – 2 Marks)

Ans: The Sql Connection Object is handling the part of physical communication between the application and the SQL Server Database. An instance of the SqlConnection class in .NET Framework is supported the Data Provider for SQL Server Database The four Objects from the .Net Framework provide the functionality of Data Providers in ADO.NET. They are Connection Object, Command Object ,DataReader Object and DataAdapter Object.

Connection Object The Connection Object provides physical connection to the Data Source. Connection Object connect to the specified Database and open a connection between the application and the Database . When the connection is established , SQL Commands may be executed, with the help of the Connection Object, to retrieve or manipulate data in the Database

Data Reader The DataReader Object is a stream-based , forward-only, read-only retrieval of query results from the Data Source, which do not update the data. DataReader Object in ADO.NET is a stream-based , forward-only, read-only retrieval of query results from the Data Source, which do not update the data. The DataReader cannot be created directly from code, they created only by calling the ExecuteReader method of a Command Object. **DataReader =Command.ExecuteReader*()*** DataReader Object provides a connection oriented data access to the data Sources. A Connection Object can contain only one DataReader at a time and the connection in the DataReader remains open and cannot be used for any other purpose while data is being accessed. When started to read from a DataReader it should always be open and positioned prior to the first record. The Read() method in the DataReader is used to read the rows from DataReader and it always moves forward to a new valid row, if any row exist. **DataReader.Read()** There are two types of DataReader in ADO.NET. They are SqlDataReader and the OleDbDataReader. The System.Data.SqlClient and System.Data.OleDb are containing these DataReaders respectively

DataSetstores many DataTables in VB.NET programs. A DataSet is conceptually a set of DataTables and other information about those tables. It is an abstraction that makes programs simpler to develop. This is a container for multiple DataTables. You can use it to create XML. It is a useful abstraction for simplifying program.

[Note: same example can be considered for MS-access database, in that case instead of sqlclient, oledb can be used]

<u>Example using Dataset</u>

```
Imports System.Data.SqlClient
Public Class Form1
Dim con As SqlConnection
 Dim da As SqlDataAdapter
 Dim ds AsDataSet
Dim dv As DataView
 Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
con = New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Documents and Settings\vaishali\My Documents\mydb.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")
con.Open()
da = New SqlDataAdapter("select * from student", con)
ds = New DataSet
da.Fill(ds, "student")
dv = New DataView(ds.Tables("student"))
 DataGridView1.DataSource = dv
End Sub
End Class
```

Example using DataReader

```
Imports System.Data.SqlClient
Public Class Form2
Dim con As SqlConnection
Dim cmd As SqlCommand
Dim drAsSqlDataReader
Private Sub buttonshow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles buttonshow.Click
con = New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Documents and Settings\vaishali\My Documents\mydb.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")
con.Open()
cmd = New SqlCommand("select * from student", con)
Dim b As Boolean
dr = cmd.ExecuteReader()
b = dr.Read()
If b Then txtrollno.Text = dr(0).ToString
txtname.Text = dr(1)
txtyear.Text = dr(2)
txtcourse.Text = dr(3)
End If
End Sub
End class
```

(b) What is “Composite Data Bound” control? Explain Detail view and Grid view control with example.

(Composite Data Bound explanation – 2 Marks, Detail view Explanation – 2 Marks and example – 1 Mark, Grid View Control Explanation – 2 Marks and example – 1 Mark)

Ans: Composite Data Bound control are separate classes which address the UI based needs and it defines the common foundation for all composite data bound controls.

Composite data bound control is an abstract class which declares and implements the Controls property. In ASP.NET the developer has to only call the PerformDataBinding method and all tasks are performed and the implementation of the CreateChildContros is called to implement the building control tree.

DetailView control

It is a data-bound user interface control that renders a single record at a time from its associated data source,optionally providing paging buttons to navigate between records.

It is similar to the form view of an Access database and is that DetailsView has a buit-in tabular rendering, whereas formView requires a user-defined template for its rendering.

Example:

```
<%@ Page
Language="VB"AutoEventWireup="false"CodeFile="Default.aspx.vb"Inherits="_Default"%>
<html xmlns-"http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
</head>
<body.
<form id=Form1"runat="server">
<asp:DetailsView ID="DetailsView1"runat="server"AllowPaging="true"AutoGenerate
Rows="false"dataSourceID="sqlDataSource1"Header Text="RollnoSortExpression="Rollno"/>
</Field>
</asp:DetailView>
</asp:DetailView>
<asp:sqlDtasource ID="SqlDataSource1"runat="server"ConnectionString="data source=lab-311
\SQLexpress;Initial catalog=tempdb;Integrated Security=true"
ProviderName="System.Data.Sqlclient"SelectCommand="SELECT* FROM[student]">
</asp:sqlDataSource>
</body>
</html>
```

GridView Control

- The GridView control is used to display the values of a data source in a table.

Each Column represents a field, while each row represents a records. It is a successor of datagride control.

- The GridView control supports the Following features:
  1. Binding to data source control, such as sqldataSource.
  2. Built-in sorting, updating, deleting, paging capabilities.
  3. Built –in row selecting capabilities.
  4. Multiple key fields.
  5. Multiple data field for the hyperlink columns.
  6. Customizable appearance through themes and styles.

Example:

```
<%@Page Language=”VB”
AutoEventWireup=’false”codeFile=”Default.aspx.vb”Inherits=”_Default”%>
<!DOCTYPE html PUBLIC”-//W3c//DTDXHTML 1.0 Transitional//EN”
        http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns=http://www.w3.org/1999/xhtml>
<head runat=”server”><title>Unititled page</titlex/head>
<body>
<form id=”Form1”runat=”server”>
<asp:GridView ID=”GridView1”dataSourceID=“sqldataSource1”runat=”server”/>
<asp:aqlDatasourceID=”sqlDataSource1”Runat=”Server”ConnectionString=”Data
Source=.;Initial Catalog=northwind;UID=sa;Password=sa”SelectCommand=”SELECT*
FROM Employee”ProviderName=”System.Data.Aqlclient”/>
</form></body></html>
```

(c) Explain built in ASP.NET object in details.

(Any four ASP.NET objects – 2 Marks each)

Ans: Server Object:

- It is used to provide a set of essential properties and methods that enables a user to perform action on a server.
- Set the timeout so that long tasks can be completed.encode text so that characters
  Not allowed in a browser parameter can be used without incident.

- Transforms text that is html tags into the correct codes to display properly.

Response Object:

- Response object is used to send output to the user from the server.
- Halt the page being served in mid stream if required.
- Transfer the page being served.send cookies to the browser.

Request Object:

- Retrieve any form values that a page submits to the script
- When a browser asks for a page from a server, it is called a request. The Request object is used to get information from a visitor.
- Retrieve any form values that a page submits to the script.
- Retrieve some vital server and script .

Application Object:

- Use to store information that many scripts need to access globally.
- Can be used to store objects that many scripts will access.
- Application variables might be affected adversely if task switching occurs.It offers a LOCK and UNLOCK capability to force such operations to not be uninterrupted.

Session Object:

- Used to store data attached to a specific user/browser with minimal effort on the programmer's part.
- Can be used to store data on the server that can only be accessed by scripts but never is stored on the client's cookies.