MOCK EXAM

<u>Question 1</u> (20 Marks)

a) Briefly why **BOTH** server-side and scripting over client-side scripting often needed to build one website. (8 marks)

- Client-side scripting usually involves the user-interface of the website including buttons and etc. Client-side scripting usually involves HTML, CSS and JavaScript which is to deliver a design to a website. But, even so apart from the user-interface the website is static which means contents are always the same unless re-coded. For a dynamic website where contents change or when you want a website to pull data from a database and much more, server-side scripting is required so that these webpages can pull data in and out from the server, making it dynamic.

b) Provide brief explanation on the purpose of cookies. (4 marks)

- Cookies are small files stored in a user's computer memory. The purpose of cookies is to store information for the website to use again in the future and etc. It is used for long term memory unlike sessions which last as long as the user stays on the website, but cookies can last virtually forever until the user decided to delete it. It can be used to stored information such as personal customization and save data for online games and etc.

c) Demonstrate the method to pass a data value using Form Post via a Hidden Field. (8 marks)

MyStudent.aspx

```
<form id="form1" runat="server" method="post" action="Result.aspx">
        <asp:Label  runat="server" id="lblText" Text="Hello World"><asp:Label>
        <asp:Button runat="server" ID="btnProcess" Text="Process"
        OnClick="btnProcess_Click" PostBackUrl="~/MyStudent.aspx"/>
        <asp:Button runat="server" ID="btnSubmit" Text="Submit"/><br /><br />
        <asp:HiddenField runat="server" ID="hdfText" />
</form>
```

MyStudent.aspx.cs

```
protected void btnProcess_Click(object sender, EventArgs e)
 {
         hdfText.Value = lblText.Text;
 }
```

<u>Question 2</u> (20 Marks)

a) Briefly explain **<u>THREE (3)</u>** advantages of ASP.NET. (9 marks)

- The framework reduces coding time, with full functionality in .NET, users can code less for the same output. Utilizing Visual Studio toolbox and the fact the Visual Studio does the server hosting. With server side and common functions already handled by Visual Studio, user don't have to reinvent the wheel by developing everything from scratch.

- ASP.NET has a separate code for its HTML layout and Server-side code, this means that both the backend and user-interface designer can do their own job individually.

- ASP.NET pages are compiled, in which enables faster reading and better performance when compared to the previous ASP technology.

b) Provide purposes and show example of the following ASP.NET Page Structure

(i) Page Directives

(ii) ASP.NET Scripting Block

(6 marks)

c) Discuss is the function of view state controls. (5 marks)

<u>Question 3</u> (20 Marks)

Basic CRUD operation is imperative to maintain persistence storage. Based on the table in Figure 1, provide ASP.NET C# code in the buttons' events to perform Delete and Update Operations assuming a valid connection string in Web.config. (20 marks)

Table Name : TStudent

SID (varchar)	**Name (varchar)**	**Age (integer)**
S001	Mamat	20
S002	Joyah	25
S003	Minah	19

Figure 1 – Table TStudent

CREATE

```
protected void btnAddNew_Click(object sender, EventArgs e)
      {
         SqlConnection curConn = new SqlConnection(connString);
         SqlCommand cmd = new SqlCommand();
         cmd.Connection = curConn;
         cmd.CommandType = System.Data.CommandType.Text;
         cmd.CommandText = "insert into TStudent values ('" + txtSID.Text + "','"
                                             + txtName.Text + "','’
                                             + txtAge.Text"')";

         try
         {
            curConn.Open();
            cmd.ExecuteNonQuery();
            lblStatus.Text = "New record " + txtSID.Text + "added!";

         }
         catch (Exception ex)
         {
            Response.Write(ex.Message);
         }
         finally
         {
            curConn.Close();
         }
      }
```

UPDATE

```
protected void btnUpdate_Click(object sender, EventArgs e)
      {
         SqlConnection curConn = new SqlConnection(connString);
         SqlCommand cmd = new SqlCommand();
         cmd.Connection = curConn;
         cmd.CommandType = System.Data.CommandType.Text;
         cmd.CommandText = "update TStudent set Name = '" +
                         txtName.Text + "', Age = " +
                         txtAge.Text +"', SID = '" + txtSID.Text + " ' ";

         try
         {
            curConn.Open();
            cmd.ExecuteNonQuery();
            lblStatus.Text = "Updated record " + txtSID.Text + "!";

         }
         catch (Exception ex)
         {
            Response.Write(ex.Message);
         }
         finally
         {
            curConn.Close();
         }
      }
```

DELETE

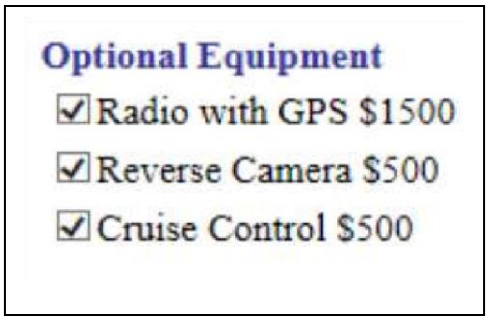
```
protected void btnDelete_Click(object sender, EventArgs e)
        {
            SqlConnection curConn = new SqlConnection(connString);
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = curConn;
            cmd.CommandType = System.Data.CommandType.Text;
            cmd.CommandText = "delete TStudent where SID = '" + txtSID.Text + "'";

            try
            {
                curConn.Open();
                cmd.ExecuteNonQuery();
                lblStatus.Text = "Deleted record " + txtSID.Text + "!";

            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
            finally
            {
                curConn.Close();
            }
        }
```

READ

```
protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection curConn = new SqlConnection(connString);
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = curConn;
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = "Select * from TStudent ";
        try
        {
            curConn.Open();
            SqlDataReader sdr = cmd.ExecuteReader();
            Response.Write("<table border=1>");
            Response.Write("<tr>");
            Response.Write("<th>SID</th> <th>Name</th> <th>Age</th>);
            Response.Write("</tr>");
            while (sdr.Read())
            {
                Response.Write("<tr>");
                Response.Write("<td>" + sdr.GetString(0) + "</td>");
                Response.Write("<td>" + sdr.GetString(1) + "</td>");
                Response.Write("<td>" + sdr.GetInt32l(2) + "</td>");
                Response.Write("</tr>");
            }
            Response.Write("</table>");
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
        }
        finally
        {       curConn.Close();      }
}
```

<u>Question 4</u> (20 Marks)

a) Write C# code to accumulate the sum of all checked items in a control named chkEquipment as shown in Figure 2. (8 marks)



Optional Equipment

☑ Radio with GPS $1500

☑ Reverse Camera $500

☑ Cruise Control $500

Figure 2 – A CheckBoxList control

```
protected void btnProcess_Click(object sender, EventArgs e)
{
   double totCost = 0;
   foreach (ListItem i in chkEquipment.Items)
   {
      if (i.Selected)
         totCost = totCost + Convert.ToDouble(i.Value);
   }

   labTotCost.Text = totCost.ToString("0.00");

}
```

b) Provide proper ASPX validation code and error messages for the following scenario:

(i) A TextBox named txtStudentNo that must be filled.

```
<form id="form1" runat="server">

      Student No: <br /> <asp:TextBox ID=" txtStudentNo" runat="server" />

      <asp:RequiredFieldValidator  CssClass="error" ID="rfvUserID"
      ControlToValidate=" txtStudentNo" ErrorMessage="Student No is
      required" runat="server" />

 </form>
```

(ii) TextBox named txtSalary that needs a value greater than zero.

```
<form id="form1" runat="server">

     Salary :<asp:TextBox runat="server" ID=" txtSalary"></asp:TextBox>
     <asp:CompareValidator ID="compareAge" ControlToValidate=" txtSalary"
     ValueToCompare="0" Operator="GreaterThan" Type="Integer"
     ErrorMessage="Salary must be above 0!" runat="server"/>
     <asp:RequiredFieldValidator ID="salaryValidator" ControlToValidate="
     txtSalary" ErrorMessage="Salary required!" runat="server" />

</form>
```

(ii) A TextBox named txtYear that limits valid year from 1990 to 2015. [DATE NOT YEAR]

```
<form id="form1" runat="server">

        Date Graduated: <asp:TextBox runat="server" ID=" txtYear
        "></asp:TextBox>

                <asp:CompareValidator ID="compareDate" ControlToValidate="
                txtYear " Operator="DataTypeCheck" Type="Date"
                ErrorMessage="Enter a valid date format!"
                runat="server" />

                <asp:RangeValidator runat="server" Type="Date"
                ControlToValidate="txtDate"
                MinimumValue="01/01/1990" MaximumValue="31/12/2015"
                ErrorMessage="Graduation Date between 2012 till June
                2017"></asp:RangeValidator>
                <br /><br />

</form>
```

(iii) A TextBox named txtUserID and txtVerifyUID that must be identical.

```
<form id="form1" runat="server">

            User ID: <br /> <asp:TextBox ID=" txtUserID " runat="server"
            TextMode="Password"/>

            VERIFY User ID: <br /> <asp:TextBox ID=" txtVerifyUID "
            runat="server" TextMode="Password"/>

            <asp:CompareValidator runat="server" ID="cvPassword"
            CssClass="error" ControlToCompare=" txtUserID " ControlToValidate="
            txtVerifyUID " ErrorMessage="User ID Mismatch! />

 </form>
```

(12 marks)

<u>Question 5</u> (20 Marks)

a) Given a list of colors, write C# code to bind the list to an ASP.NET RadioButtonList.

(6 marks)

```
<asp:RadioButtonList ID="RadioButtonList1" runat="server">
            <asp:ListItem>Red</asp:ListItem>
            <asp:ListItem>Green</asp:ListItem>
            <asp:ListItem>Blue</asp:ListItem>
</asp:RadioButtonList>
```

b) What are the purpose of Rendering stage in ASP.NET page life cycle?

(3 marks)

c) Briefly explain the two types of SQL queries often use in database connected website.

(5 marks)

d) Briefly explain the following core objects in ADO.NET:

(i) Data Provider

(ii) DataSet

(6 marks)

- END -