```csharp
private void btnfind_click (object sender, eventArgs e)
{
try
{
    Rec = DS.Tables[0].Select ("Empid ="+txtempid.Text)[0];
    txtEName.Text = Rec[1].Tostring();
    txt Designation.Text = Rec[2].Tostring();
    txtDOJ.Text = Rec[3].Tostring();
    txt salary.Text = Rec[4].Tostring();
    txt Deptno.Text = Rec[5].Tostring();
    txtupdate.Text = Rec[6].Tostring();
    btnupdate.enable = true;
    btnDelete.Enabled = true;
}
catch (Indexout of range exception e)
{
    MessageBox.show ("Record not found");

}
private void btnupdate_click (object sender, EventArgs e)
{
    Rec[1] = txtename.Text;
    Rec[2] = txt Designation.Text;
    Rec[3] = txt DOJ.Text;
    Rec[4] = txtsalary.Text;
    Rec[5] = txtDeptno.Text;

    MessageBox.show ("Data is update in Dataset table");
    btn update.Enabled = false;
    btn Delete.Enabled = false;

}
private void btnDelete_click
{
```

```
Rec . Delete();
MessageBox. show ("Record is Deleted from Dataset table");
btn update .Enabled = false;
btn delete .Enabled = false;
}
```

Another code for find :-

```
Rec = DS. Tables [0]. Rows.find (txt empid .Text);

if (Rec !=Null)
{
txt Name .Text =Rec [1] ToString();
txtDesignation .Text =Rec[2] .Tostring();
txt DOJ .Text = Rec [3] ToString();
txt salary .Text= Rec[4] ToString();
txt DeptNo .Text = Rec [5] ToString();
}
else
    MessageBox .show ("Record not found");
```

update to Database

```
private void btnupdateDB -click (object sender, EventArgse)

{
int Rows = DA .update (DS.Tables[0]);
MessageBox . show (Rows+" Records) updated to Database");
}
private void btnclear -click (object sender, -eventArgse )

{
foreach (control c in this. controls
{
if (c is Textbox)
```

```
            C.Text = "";
        }
    }
```

Another Method update DB:-

```
    if (DS.Haschanges())
    {
        int Rows = Da.update(OS.Tables[0]);
        Message Box.show (Rows + " Record(s) updated to Database");
    }
    else
        MessageBox.show ("No Modifications in Dataset tables");
```

1) Component Development
2) MDI Forms
3) Extension Methods
4) SOLID Design Principles
5) Dependency Injection
5) N-Tier Architecture