

Advance Programming Techniques (APT)

Lecture # 39

Ehtisham Rasheed

Department of Computer Science
University of Gurjat, Gurjat

Introduction to Multiple Forms

- Most real applications do not consist of a single form. They use multiple screens such as:
 - **Splash Screen**
 - **Login Form**
 - **Dashboard / Main Form**
 - **Settings Form**
 - **About Form**
- In C#, using WinForms, each screen is simply another **Form class**

Creating a New Form

- Follow the following steps to create a new form:
- Right-click your project → **Add** → **Windows Form**
- Give it a name:
 - SplashScreen.cs
 - LoginForm.cs
 - Dashboard.cs

Opening Another Form

```
private void btnOpen_Click(object sender, EventArgs e)
{
    Dashboard dashboard = new Dashboard();
    dashboard.Show();
}
```

```
LoginForm login = new LoginForm();
login.ShowDialog();
```

Prepare
HTML

Hiding the Current Form

```
this.Hide();  
Dashboard d = new Dashboard();  
d.Show();
```

- But hiding is **NOT** closing
- The application keeps running in the background unless you explicitly exit

Passing Data Between Forms

- Login Form:

```
private void btnLogin_Click(object sender, EventArgs e)
{
    string user = txtUsername.Text;

    Dashboard d = new Dashboard(user);
    d.Show();
    this.Hide();
}

public Dashboard(string username)
{
    InitializeComponent();
    lblWelcome.Text = "Welcome, " + username;
}
```

Prepared By: [REDACTED]

Closing Strategy (VERY IMPORTANT)

- If you hide forms without proper closing, the app stays running
- Dashboard form should handle exit:

```
private void Dashboard_FormClosing(object sender, FormClosingEventArgs e)
{
    Application.Exit();
}
```

Prep