

Contents

Login Form	1
Basic form	2
Basic info	5
Death form	6
Death Info	8
Passport Form	10
Passport info	11
Birth form	12
Birth info	14

Login Form

```
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;

namespace IdRegistrationForm
{
    public partial class Login : Form
    {

        public Login()
        {
            InitializeComponent();
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            string[] username = new string[3] { "bob", "rob", "nob" };
            string[] password = new string[3] { "bob", "bor", "bon" };

            // if checks the array to see if the textfields match
            if (username.Contains(txtUserName.Text) &&
password.Contains(txtPassword.Text))
            {
                this.Close();
            }
        }
    }
}
```

```

        else
if((string.IsNullOrEmpty(txtUserName.Text))|| (string.IsNullOrEmpty(txtPassword.Text)))
        {
            // if the textboxes are empty it will give an error
            MessageBox.Show("please enter sonme data in the fields");
        }
        else {
            MessageBox.Show("error the username or password was incorrect
try again");
        }
    }

    private void Login_Load(object sender, EventArgs e)
    {

    }

    private void Register_Click(object sender, EventArgs e)
    {

if((string.IsNullOrEmpty(txtUserName.Text))|| (string.IsNullOrEmpty(txtPassword.Text)))
    {
        // if the textboxes are empty it will give an error
        MessageBox.Show("please enter some data in the fields");
    }
    else
    {
        MessageBox.Show("are you sure you are ok with this username and
password");
        this.Close();
    }
}

}
}
}

```

Basic form

```

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;

namespace IdRegistrationForm
{
    public partial class BasicForm : Form
    {
        PassportForm passFormObject = new PassportForm();
        BirthForm birthFormObject = new BirthForm();
    }
}

```

```

DeathForm deathFormObject = new DeathForm();
Login loginFormObject = new Login();

public BasicForm()
{
    //makes login show first

    InitializeComponent();

    cmbRegistration.SelectedIndex = 1;

    loginFormObject.ShowDialog();

}

// generate the Id
public static string GenerateIdNumber(string aDateOfBirth, string aSex, string
aCountryOfBirth)
{
    string idNumber;
    int genderNumber = 0;
    int regionNumber = 1;
    Random rnd = new Random();
    //if statements test radio buttons to change it to a readable string
    if (aSex.Equals("male"))
    {
        genderNumber = rnd.Next(5000, 9999);
    }
    else if (aSex.Equals("female"))
    {
        genderNumber = rnd.Next(1000, 5000);
    }

    //checks if user is from south africa
    if (aCountryOfBirth.Equals("South Africa") || aCountryOfBirth.Equals("south
africa"))
    {
        regionNumber = 0;
    }
    idNumber = aDateOfBirth + genderNumber + regionNumber + "8" + "2";

    return idNumber;
}

private void btnAccept_Click(object sender, EventArgs e)
{
    this.Hide();

    //get the sex of the person
    string sex = string.Empty;
    if (rdbMale.Checked)
    {
        sex = "male";
    }
    else if (rdbFemale.Checked)
    {

```

```

        sex = "female";
    }
    //GenerateIdNumber(txtDateOfBirth.Text,sex,txtCountryOfBirth.Text)

    string registraionType = cmbRegistration.Text;

    //iff statement to determine the form opened
    if (registraionType.Equals("Passport"))
    {
        passFormObject.ShowDialog();
        //on form open creates object of the info class to get al the
information
        PassportInfo passInfo = new PassportInfo(txtName.Text,
txtSurname.Text, txtDateOfBirth.Text, txtCountryOfBirth.Text, sex,
GenerateIdNumber(txtDateOfBirth.Text, sex, txtCountryOfBirth.Text), txtTelNumber.Text,
txtWorkNumber.Text,
        passFormObject.GetPreviousSurname(),
passFormObject.GetPlaceOfBirth(), passFormObject.GetMaritalStatus());
        MessageBox.Show(passInfo.ToString());
    }
    else if (registraionType.Equals("Birth certificate"))
    {
        birthFormObject.ShowDialog();
        BirthInfo birthInfo = new BirthInfo(txtName.Text, txtSurname.Text,
txtDateOfBirth.Text, txtCountryOfBirth.Text, sex,
GenerateIdNumber(txtDateOfBirth.Text, sex, txtCountryOfBirth.Text), txtTelNumber.Text,
txtWorkNumber.Text,
        birthFormObject.GetBirthNumber(),
birthFormObject.GetFatherSurname(), birthFormObject.GetFatherForename(),
birthFormObject.GetMotherMaidenName(), birthFormObject.GetMotherForeName(),
birthFormObject.GetApplicantSurname(), birthFormObject.GetApplicantForename(),
birthFormObject.GetApplicantPostalAddress(),
birthFormObject.GetApplicantPostalCode());
        MessageBox.Show(birthInfo.ToString());
    }
    else if (registraionType.Equals("Death certificate"))
    {
        deathFormObject.ShowDialog();
        DeathInfo deathInfo = new DeathInfo(txtName.Text, txtSurname.Text,
txtDateOfBirth.Text, txtCountryOfBirth.Text, sex,
GenerateIdNumber(txtDateOfBirth.Text, sex, txtCountryOfBirth.Text), txtTelNumber.Text,
txtWorkNumber.Text,
        deathFormObject.GetDateOfDeath(), deathFormObject.GetDeathEntry(),
deathFormObject.GetTownOfDeath(), deathFormObject.GetProvinceOfDeath(),
deathFormObject.GetUndertaker(), deathFormObject.GetApplicantSurname(),
deathFormObject.GetInitials(), deathFormObject.GetIdNumber(),
deathFormObject.GetPostalAddress(), deathFormObject.GetPostalCode(),
deathFormObject.GetTelNumber(), deathFormObject.GetResidentialAddress());
        MessageBox.Show(deathInfo.ToString());
    }
}

}

private void BasicForm_Load(object sender, EventArgs e)
{
}

```

```
}  
}
```

Basic info

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
  
namespace IdRegistrationForm  
{  
    class BasicInfo  
    {  
        string firstName = string.Empty;  
        string surname = string.Empty;  
        string dateOfBirth = string.Empty;  
        string countryOfBirth = string.Empty;  
        string gender = string.Empty;  
        string idNumber = string.Empty;  
        string telNumber = string.Empty;  
        string workNumber = string.Empty;  
  
        public BasicInfo(string aFirstName, string aSurname , string aDateOfBirth,  
string aCountryOfBirth,string aGender, string aIdNumber,string aTelNumber,string  
aWorkNumber )  
        {  
            firstName=aFirstName;  
            surname= aSurname;  
            dateOfBirth= aDateOfBirth;  
            countryOfBirth= aCountryOfBirth;  
            gender = aGender;  
            idNumber= aIdNumber;  
            telNumber= aTelNumber;  
            workNumber = aWorkNumber;  
        }  
  
        public string TelNumber  
        {  
            get { return telNumber; }  
            set { telNumber = value; }  
        }  
  
        public string WorkNumber  
        {  
            get { return workNumber; }  
            set { workNumber = value; }  
        }  
  
        public string CountryOfBirth  
        {  
            get { return countryOfBirth; }  
            set { countryOfBirth = value; }  
        }  
    }  
}
```

```

    public string DateOfBirth
    {
        get { return dateOfBirth; }
        set { dateOfBirth = value; }
    }

    public string Surname
    {
        get { return surname; }
        set { surname = value; }
    }

    public string IdNumber
    {
        get { return idNumber; }
        set { idNumber = value; }
    }

    public string FirstName
    {
        get { return firstName; }
        set { firstName = value; }
    }

    //overloaded to string to get all the values into one form
    public override string ToString()
    {
        return "firstname: " + firstName + "\nsurname: " + surname + "\ndate of
        birth: " + dateOfBirth + "\ncountry of birth: " + countryOfBirth + "\nsex: " + gender
        + " \nID number: " + idNumber + "\nTel number: " + telNumber + "\nworkNumber: " +
        workNumber;
    }
}

```

Death form

```

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;

namespace IdRegistrationForm
{
    public partial class DeathForm : Form
    {
        //gets for all the values to populate the toString method
        public string GetDateOfDeath()
        {
            return txtDeathDate.Text;
        }

        public string GetDeathEntry()
        {

```

```

        return txtDeathNumber.Text;
    }

    public string GetTownOfDeath()
    {
        return txtCityOfDeath.Text;
    }

    public string GetProvinceOfDeath()
    {
        return txtProvinceOfDeath.Text;
    }

    public string GetUndertaker()
    {
        return txtUndertakerName.Text;
    }
    public string GetApplicantSurname()
    {
        return txtSurname.Text;
    }

    public string GetInitials()
    {
        return txtInitials.Text;
    }

    public string GetIdNumber()
    {
        return txtIdNumber.Text;
    }

    public string GetPostalAddress()
    {
        return txtPostalAddress.Text;
    }

    public string GetPostalCode()
    {
        return txtPostalCode.Text;
    }

    public string GetTelNumber()
    {
        return txtTelNumber.Text;
    }
    public string GetResidentialAddress()
    {
        return txtResidential.Text;
    }

    public DeathForm()
    {
        InitializeComponent();
    }

    private void btnAccept_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void DeathForm_Load(object sender, EventArgs e)

```

```

    {
    }
}

```

Death Info

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace IdRegistrationForm
{
    class DeathInfo :BasicInfo
    {
        string dateOfDeath = string.Empty;
        string deathEntryNumber = string.Empty;
        string townCityOfDeath = string.Empty;
        string provinceOfDeath = string.Empty;
        string nameOfUndertaker = string.Empty;
        string applicantSurname = string.Empty;
        string applicantInitials = string.Empty;
        string applicantIdNumber = string.Empty;
        string applicantPostalAddress = string.Empty;
        string applicantPostalCode = string.Empty;
        string applicantTelNumber = string.Empty;
        string applicantResidentialAddress = string.Empty;

        //overloaded constructor to hold all the information for passport form
        public DeathInfo(string aFirstName, string aSurname, string aDateOfBirth,
            string aCountryOfBirth, string aGender, string aIdNumber, string aTelNumber, string
            aWorkNumber,
                string aDateOfDeath ,string aDeathEntryNumber,string
            aTownCityOfDeath,string aProvinceOfDeath,string aNameOfUndertaker,string
            aApplicantSurname ,string aApplicantInitials ,
                string aApplicantIdNumber,string aApplicantPostalAddress ,string
            aApplicantPostalCode,string aApplicantTelNumber ,string aApplicantResidentialAddress)
            : base(aFirstName, aSurname, aDateOfBirth, aCountryOfBirth, aGender,
            aIdNumber, aTelNumber, aWorkNumber)
        {
            dateOfDeath = aDateOfDeath;
            deathEntryNumber = aDeathEntryNumber;
            townCityOfDeath = aTownCityOfDeath;
            provinceOfDeath = aProvinceOfDeath;
            nameOfUndertaker = aNameOfUndertaker;
            applicantSurname = aApplicantSurname;
            applicantInitials = aApplicantInitials;
            applicantIdNumber = aApplicantIdNumber;
            applicantPostalAddress = aApplicantPostalAddress;
            applicantPostalCode = aApplicantPostalCode;
            applicantTelNumber = aApplicantTelNumber;
            applicantResidentialAddress = aApplicantResidentialAddress;
        }

        public string DateOfDeath

```



```
{
    get { return dateOfDeath; }
    set { dateOfDeath = value; }
}

public string DeathEntryNumber
{
    get { return deathEntryNumber; }
    set { deathEntryNumber = value; }
}

public string TownCityOfDeath
{
    get { return townCityOfDeath; }
    set { townCityOfDeath = value; }
}

public string ProvinceOfDeath
{
    get { return provinceOfDeath; }
    set { provinceOfDeath = value; }
}

public string NameOfUndertaker
{
    get { return nameOfUndertaker; }
    set { nameOfUndertaker = value; }
}

public string ApplicantSurname
{
    get { return applicantSurname; }
    set { applicantSurname = value; }
}

public string ApplicantInitials
{
    get { return applicantInitials; }
    set { applicantInitials = value; }
}

public string ApplicantIdNumber
{
    get { return applicantIdNumber; }
    set { applicantIdNumber = value; }
}

public string ApplicantPostalAddress
{
    get { return applicantPostalAddress; }
    set { applicantPostalAddress = value; }
}

public string ApplicantPostalCode
{
    get { return applicantPostalCode; }
    set { applicantPostalCode = value; }
}

public string ApplicantTelNumber
{
    get { return applicantTelNumber; }
}
```

```

        set { applicantTelNumber = value; }
    }

    public string ApplicantResidentialAddress
    {
        get { return applicantResidentialAddress; }
        set { applicantResidentialAddress = value; }
    }
    //overloaded toString() to get all the values printed
    public string ToString()
    {
        return "Particulars of the deceased\n"+"-----\n" +
base.ToString() + "\ndate of death: " + dateOfDeath + "\ndeath entry number: " +
deathEntryNumber + "\ntown or city of death: " + townCityOfDeath + "\nprovince of
death: " + provinceOfDeath + "\nname of undertaker: " + nameOfUndertaker + "\n\n
Partuclars of applicant \n" + "-----" + "\nSurname: " +
applicantSurname + "\nInitials: " + applicantInitials + "\nID number: " +
applicantIdNumber + "\npostalAddress: " + applicantPostalAddress + "\npostalCode: " +
applicantPostalCode + "\ntel number: " + applicantTelNumber + "\nResidentail Address:
" + applicantResidentialAddress;
    }
}
}

```

Passport Form

```

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;

namespace IdRegistrationForm
{
    public partial class PassportForm : Form
    {

        public string GetPreviousSurname()
        {
            string previousSurname = txtPreviousSurname.Text;

            return previousSurname;
        }

        public string GetPlaceOfBirth()
        {
            string placeOfBirth = txtPlaceOfBirth.Text;

            return placeOfBirth;
        }

        //get method will check which radiobutton is selected
        public string GetMaritalStatus()
        {

```

```

        string maritalStatus = string.Empty;
        if(rdbDevorced.Checked)
        {
            maritalStatus = "Devorced";
        }
        else if (rdbMarried.Checked)
        {
            maritalStatus = "Married";
        }
        else if (rdbUnmarried.Checked)
        {
            maritalStatus = "Unmarried";
        }
        else if (rdbWidow.Checked)
        {
            maritalStatus = "Widow";
        }
        else if (rdbWidower.Checked)
        {
            maritalStatus = "Widower";
        }

        return maritalStatus;
    }

    public PassportForm()
    {
        InitializeComponent();
    }

    private void btnAccept_Click(object sender, EventArgs e)
    {
        //will close after accept is pressed

        this.Close();
    }

    private void PassportForm_Load(object sender, EventArgs e)
    {
    }
}

```

Passport info

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace IdRegistrationForm
{
    class PassportInfo : BasicInfo
    {
        string previousSurname = string.Empty;
        string placeOfBirth = string.Empty;
    }
}

```

```

        string maritalStatus = string.Empty;

        //overloaded constructor for the passport
        public PassportInfo(string aFirstName, string aSurname, string aDateOfBirth,
            string aCountryOfBirth, string aGender, string aIdNumber, string aTelNumber, string
            aWorkNumber, string aPreviousSurname, string aPlaceOfBirth, string aMaritalStatus)
            : base(aFirstName, aSurname, aDateOfBirth, aCountryOfBirth, aGender,
            aIdNumber, aTelNumber, aWorkNumber)
        {
            previousSurname = aPreviousSurname;
            placeOfBirth = aPlaceOfBirth;
            maritalStatus = aMaritalStatus;
        }

        public string PreviousSurname
        {
            get { return previousSurname; }
            set { previousSurname = value; }
        }

        public string PlaceOfBirth
        {
            get { return placeOfBirth; }
            set { placeOfBirth = value; }
        }

        public string MaritalStatus
        {
            get { return maritalStatus; }
            set { maritalStatus = value; }
        }

        //toString method concatenating all the fields into useful information
        public string ToString()
        {
            return "Particulars of applicant\n" + "-----" + base.ToString()
            + "\nprevious surname: " + previousSurname + "\nplace of birth: " + placeOfBirth
            + "\nMarital status: " + maritalStatus;
        }
    }
}

```

Birth form

```

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;

namespace IdRegistrationForm
{
    public partial class BirthForm : Form
    {

```

```

//get methods that will be called in basic form to grab the variables
public string GetBirthNumber()
{
    return txtBirthEntryNum.Text ;
}

public string GetFatherSurname()
{
    return txtFatherSurname.Text;
}

public string GetFatherForename()
{
    return txtFatherForename.Text;
}

public string GetMotherMaidenName()
{
    return txtMotherMaidenName.Text;
}

public string GetMotherForeName()
{
    return txtMotherForename.Text;
}

public string GetApplicantSurname()
{
    return txtApplicantSurname.Text;
}

public string GetApplicantForename()
{
    return txtApplicantForename.Text;
}

public string GetApplicantPostalAddress()
{
    return txtApplicantPostalAddress.Text;
}

public string GetApplicantPostalCode()
{
    return txtApplicantPostalCode.Text;
}

public BirthForm()
{
    InitializeComponent();
}

private void BirthForm_Load(object sender, EventArgs e)
{
}

private void btnAccept_Click(object sender, EventArgs e)
{
    this.Close();
}
}
}

```

Birth info

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace IdRegistrationForm
{
    class BirthInfo : BasicInfo
    {
        string birthEntryNumber = string.Empty;
        string applicantPostalCode = string.Empty;
        string fatherSurname = string.Empty;
        string fatherForename = string.Empty;
        string motherMaidenName = string.Empty;
        string motherForename = string.Empty;
        string applicantSurname = string.Empty;
        string applicantPostalAddress = string.Empty;
        string applicantForename = string.Empty;

        //constructor that will hold all the variables
        public BirthInfo(string aFirstName, string aSurname, string aDateOfBirth,
            string aCountryOfBirth, string aGender, string aIdNumber, string aTelNumber, string
            aWorkNumber,
            string aBirthentryNumber, string aFatherSurname, string aFatherForename,
            string aMotherMaidenName, string aMotherForename, string aApplicantSurname, string
            aApplicantForename, string aApplicantPostalAddress, string aApplicantPostalCode)
            : base(aFirstName, aSurname, aDateOfBirth, aCountryOfBirth, aGender,
            aIdNumber, aTelNumber, aWorkNumber)
        {
            birthEntryNumber = aBirthentryNumber;
            fatherForename = aFatherForename;
            fatherSurname = aFatherSurname;
            motherForename = aMotherForename;
            motherMaidenName = aMotherMaidenName;
            applicantSurname = aApplicantSurname;
            applicantForename = aApplicantForename;
            applicantPostalAddress = aApplicantPostalAddress;
            applicantPostalCode = aApplicantPostalCode;
        }

        public string BirthEntryNumber
        {
            get { return birthEntryNumber; }
            set { birthEntryNumber = value; }
        }

        public string FatherSurname
        {
            get { return fatherSurname; }
            set { fatherSurname = value; }
        }

        public string FatherForename
        {
            get { return fatherForename; }
            set { fatherForename = value; }
        }
    }
}
```

```

    }

    public string MotherMaidenName
    {
        get { return motherMaidenName; }
        set { motherMaidenName = value; }
    }

    public string MotherForename
    {
        get { return motherForename; }
        set { motherForename = value; }
    }

    public string ApplicantSurname
    {
        get { return applicantSurname; }
        set { applicantSurname = value; }
    }

    public string ApplicantForename
    {
        get { return applicantForename; }
        set { applicantForename = value; }
    }

    public string ApplicantPostalAddress
    {
        get { return applicantPostalAddress; }
        set { applicantPostalAddress = value; }
    }

    public string ApplicantPostalCode
    {
        get { return applicantPostalCode; }
        set { applicantPostalCode = value; }
    }

    //overloaded toString method that will get all the values into one form
    public string ToString()
    {
        return "Particulars of born\n" + "-----\n" + base.ToString()
+ "\nbirth entry number: " + birthEntryNumber + "\nParticulars of father\n" + "-----"
+ "\nfather forename: " + fatherForename + "\nfather surname: " +
fatherSurname + "\nParticulars of mother\n" + "-----" + "\nmother
forename: " + motherForename + "\nmother maiden name: " + motherMaidenName
+ "\nParticulars of applicant\n" + "-----" + "\napplicant surname: " +
applicantSurname + "\napplicant forename: " + applicantForename + "\napplicant postal
address: " + applicantPostalAddress + "\napplicant postal code:" + applicantPostalCode
;
    }

}
}

```