Tristan Izlar

COP2360 C# Programming I

**Module 6 ASSIGNMENT Chapter 9 - Problem 4: Employee Class**

SCREENSHOTS:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

CODE FORM1:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Employee\_Class

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

// create list to house employee objects

List<Employee> employeeList = new List<Employee>();

private void addButton\_Click(object sender, EventArgs e)

{

// create a new instance of the employee class

Employee employee = new Employee();

// variable for testing if the ID is an integer

int idNum;

//get the employee name

employee.Name = inputName.Text;

// get the employee department

employee.Department = inputDept.Text;

// get the employee positon

employee.Position = inputPos.Text;

// get the the employee id

if (int.TryParse(inputID.Text, out idNum))

{

employee.IDNumber = idNum;

}

else

{

MessageBox.Show("Invalid ID Number. Default applied.");

}

//add employee to the employeeList

employeeList.Add(employee);

//add entry to list box

employeeListBox.Items.Add(employee.Name);

//clear out input boxes

inputName.Clear();

inputID.Clear();

inputDept.Clear();

inputPos.Clear();

// clear the output boxes

outputName.Text = "";

outputID.Text = "";

outputDept.Text = "";

outputPos.Text = "";

}

// when a list box item is selected the following takes place

private void employeeListBox\_SelectedIndexChanged(object sender, EventArgs e)

{

// find index of currently selected list box item

int index = employeeListBox.SelectedIndex;

// send that objects data to the output boxes to be viewed by the user

outputName.Text = employeeList[index].Name;

outputID.Text = employeeList[index].IDNumber.ToString();

outputDept.Text = employeeList[index].Department;

outputPos.Text = employeeList[index].Position;

}

private void exitButton\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

//collaboration statement: I worked alone.

CODE Employees Class:

using System;

using System.Collections.Generic;

using System.Text;

namespace Employee\_Class

{

class Employee

{

// backing fields

private string \_name;

private int \_idnumber;

private string \_department;

private string \_position;

// establishing constructors to default our values if nothing is entered

public Employee()

{

\_name = "";

\_idnumber = 0;

\_department = "";

\_position = "";

}

public Employee(string Name, int IDNumber)

{

\_name = Name;

\_idnumber = IDNumber;

\_department = "";

\_position = "";

}

public Employee(string Name, int IDNumber, string Department, string Position)

{

\_name = Name;

\_idnumber = IDNumber;

\_department = Department;

\_position = Position;

}

// creating properties for each field

public string Name

{

get { return \_name; }

set { \_name = value; }

}

public int IDNumber

{

get { return \_idnumber; }

set { \_idnumber = value; }

}

public string Department

{

get { return \_department; }

set { \_department = value; }

}

public string Position

{

get { return \_position; }

set { \_position = value; }

}

}

}

//collaboration statement: I worked alone.