**K. Vamsi Krishna Prasad**

**- J00744483**

**Code:**

**Customer.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Chapter13OOPAssignment

{

public class Customer

{

//fields with automatic properties (allow updating and retrieving field values)

private int id { get; set; }

private string name { get; set; }

private string phone { get; set; }

private string creditCard;

//list of services for the customer

public List<Services> serviceList { get; set; }

//property for credit card

public void setCreditCard(string theCreditCard)

{

this.creditCard = theCreditCard;

}

//create constructor

//default

public Customer()

{

this.id = 0;

this.name = string.Empty;

this.phone = string.Empty;

this.creditCard = string.Empty;

this.serviceList = new List<Services>();

}

//overloaded constructor allows us to add new customers

public Customer(int theId,string theName,string thePhone,string theCreditCard)

{

this.id = theId;

this.name = theName;

this.phone = thePhone;

this.creditCard = theCreditCard;

this.serviceList = new List<Services>();

}

//to string allows us to output a Customer

public override string ToString()

{

return ("Customer's Name: " + this.name + " and phone number : " + this.phone);

}

// any other methods as needed for a customer

// add a service

public void addCusomerService(Services theServices)

{

this.serviceList.Add(theServices);

}

// remove a service

public void removeCustomerService(int theIndex)

{

this.serviceList.RemoveAt(theIndex);

}

}

}

**frmSalon.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Chapter13OOPAssignment

{

public partial class frmSalon : Form

{

// create a list of customers - needs to be static (uses customer class)

public static List<Customer> customerList;

public frmSalon()

{

InitializeComponent();

}

private void frmSalon\_Load(object sender, EventArgs e)

{ // initialize the list of customers

customerList = new List<Customer>();

}

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

{ //hide existing form

this.Hide();

//open the customer form

frmCustomer CustomerForm = new frmCustomer();

CustomerForm.ShowDialog();

this.Show();

}

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

{ //hide this form

this.Hide();

//open the services form

frmAdd AddForm = new frmAdd();

AddForm.ShowDialog();

//reopen this form

this.Show();

}

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

{

Environment.Exit(0);//exit the applicaiton

}

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

{

//hide this form

this.Hide();

//open the services form

frmAdd AddForm = new frmAdd();

AddForm.ShowDialog();

//reopen this form

this.Show();

}

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

{

//hide existing form

this.Hide();

//open the customer form

frmCustomer CustomerForm = new frmCustomer();

CustomerForm.ShowDialog();

this.Show();

}

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

{

Environment.Exit(0);//exit the applicaiton

}

}

}

**frmAdd.cs**

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 Chapter13OOPAssignment

{

public partial class frmAdd : Form

{

public frmAdd()

{

InitializeComponent();

}

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

{

ClearForm();

}

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

{ //close this form, return to menu form

this.Close();

}

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

{

try

{

//Add a customer

int customerID = int.Parse(txtCustomerNumber.Text);

Customer newCustomer = new Customer(customerID, txtName.Text, txtPhone.Text, txtCreditCard.Text);

//Add the newCustomer to customer listbox. and display it by tostring method

frmSalon.customerList.Add(newCustomer);

MessageBox.Show("Customer Added.");

ClearForm();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void ClearForm()

{

txtCustomerNumber.Text = string.Empty;

txtName.Text = string.Empty;

txtPhone.Text = string.Empty;

txtCreditCard.Text = string.Empty;

txtCustomerNumber.Focus();

}

}

}

**frmCustomer**

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 Chapter13OOPAssignment

{

public partial class frmCustomer : Form

{

public frmCustomer()

{

InitializeComponent();

}

public static int SelectedIndex;

private void frmCustomer\_Load(object sender, EventArgs e)

{

PopulateListBox();

}

private void PopulateListBox()

{

//if any exist clear them out

lstCustomers.Items.Clear();

//add customers to listbox

//make sure the list isn't empty

if(frmSalon.customerList.Count != 0)

{

foreach(Customer theCustomer in frmSalon.customerList)

{

lstCustomers.Items.Add(theCustomer.ToString());

}

}

else

{

lstCustomers.Items.Add("No Customers Exist in the List.");

}

}

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

{

this.Close();

}

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

{

try

{

//add services for a customer

lstServices.Items.Clear();

SelectedIndex = lstCustomers.SelectedIndex;

if (SelectedIndex != -1)

{

Customer selectedCustomer = frmSalon.customerList[SelectedIndex];

foreach (Services theservice in selectedCustomer.serviceList)

{

lstServices.Items.Add(theservice.ToString());

}

}

else

{

lstServices.Items.Add("No service exist in the List.");

}

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

//display the AddService Form

SelectedIndex = lstCustomers.SelectedIndex;

this.Hide();

frmServices addService = new frmServices();

addService.ShowDialog();

this.Show();

}

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

{

try

{

//delete a customer

if (lstCustomers.SelectedIndex != -1)

{

frmSalon.customerList.RemoveAt(SelectedIndex);

lstCustomers.Items.RemoveAt(lstCustomers.SelectedIndex);

MessageBox.Show("Customer Deleted.");

}

else

{

MessageBox.Show("Please Select an item to delete.");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

try

{

if (lstServices.SelectedIndex != -1)

{

frmSalon.customerList[SelectedIndex].removeCustomerService(lstServices.SelectedIndex);

MessageBox.Show("Service Deleted.");

}

else

{

MessageBox.Show("Please Select an item to delete.");

}

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

**frmServices.cs**

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 Chapter13OOPAssignment

{

public partial class frmServices : Form

{

public frmServices()

{

InitializeComponent();

}

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

{

this.Close();

}

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

{

ClearForm();

}

private void ClearForm()

{

txtService.Clear();

txtPrice.Clear();

txtDescription.Clear();

txtService.Focus();

}

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

{

try

{

//add service

decimal price = decimal.Parse(txtPrice.Text);

Services newService = new Services(txtService.Text, txtDescription.Text, price, dtpService.Value);

int index = frmCustomer.SelectedIndex;

//then add to selected customer

frmSalon.customerList[index].addCusomerService(newService);

MessageBox.Show("New Service Added.");

ClearForm();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

**Services.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Chapter13OOPAssignment

{

public class Services

{

//fields with automatic properties

private string service { get; set; }

private string description { get; set; }

private decimal cost { get; set; }

private DateTime date { get; set; }

// properties - access to our fields

//read

public string getService()

{

return this.service;

}

//write

public void writeService(string theService)

{

this.service = theService;

}

//default constructor

public Services()

{

this.service = String.Empty;

this.description = String.Empty;

this.cost = 0;

this.date = DateTime.Today;

}

//overloaded constructor with data values

public Services(string theService, string theDescription, decimal theCost,DateTime theDate)

{

this.service= theService;

this.description = theDescription;

this.cost = theCost;

this.date = theDate;

}

//methods

// toString method

public override string ToString()

{

return ("Services: " + this.service + " has a cost of" + this.cost.ToString("c"));

}

}

}

**Output:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer

Description automatically generated