



C# BASICS

Training Assignments

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	20/11/2012

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1.	01/Oct/2018	Create new	Draft		
2.	01/Jun/2019	Update template	Fsoft template	DieuNT1	

Contents

Day 4: Collection & Generic	4
Objectives:	4
Prerequisites:	4
Technical Requirements:	4
Problem Requirements:	4
Estimated Time: 180 minutes.....	7
Mark Scale:	7



CODE:	NCTR.M.A04
TYPE:	MEDIUM
LOC:	190
DURATION:	180 MINUTES

Day 4: Collection & Generic

Objectives:

- » Understand and practice with Collection and Generic in C#.

Prerequisites:

- » Working environment: Visual Studio 2013 or higher.
- » Delivery: Source code packaged in a compress archive.

Technical Requirements:

Create a solution named **NCTR.M.A04**.

Within the solution, create a console project corresponding to the following exercise:

- **NCTR.M.A04**

Problem Requirements:

You are tasked with designing and implementing an e-commerce platform. The platform should support management of customers, products, and orders. Additionally, it should provide user interfaces for adding, updating, and removing customers, products, and orders. Furthermore, it should allow for displaying all customers, products, and orders.

Model Classes:

1. Customer:

- Properties:
 - Id (int): Unique identifier (auto-generated)
 - Name (string, required): Customer's full name
 - PhoneNumber (string, required): Valid phone number (implement validation logic)
 - Address (string, optional): Customer's delivery address
- ToString() method:
 - Should return a formatted string representation of the customer object, including all properties.

2. Product:

- Properties:
 - Id (int): Unique identifier (auto-generated)
 - Name (string, required): Product name
 - Price (double, required): Product price
 - Description (string, optional): Detailed product description
- ToString() method:
 - Should return a formatted string representation of the product object, including all properties.

3. Order:

- Properties:
 - Id (int): Unique identifier (auto-generated)
 - Customer (Customer object, required): Reference to the customer who placed the order
 - Products (List<Product>, required): List of products included in the order
 - OrderDate (DateTime, required): Date and time the order was placed
 - Note (string, optional): Any additional notes associated with the order
- ToString() method:
 - Should return a formatted string representation of the order object, including all properties.

Management Class:

- **Abstract Class (Base Class): GenericManagement<T> (where T is the model class)**
 - **Properties:**
 - List<T> items: Internal list to store objects of the specified type (Customer, Product, or Order).
 - **Methods:**
 - GetList(): Returns a copy of the internal list.
 - Add(T item): Adds a new item to the internal list. Allow override
 - Remove(T item): Removes an item from the internal list based on a matching property (e.g., Id). Allow override
 - Display(): Formats and displays all items in the list to the console. Allow override
 - FindById(int id): Abstract method.
 - Update(T item): Abstract method.

- **Customer Management:**

This class inherits from **GenericManagement<Customer>** and provides specific functionalities for managing customers.

- **Methods:**
 - AddCustomer(): Prompts user for customer details and calls Add on the CustomerManagement object.
 - UpdateCustomer(): Allows user to search for a customer by Id, modify details, and calls Update on the CustomerManagement object.
 - RemoveCustomer(): Allows user to search for a customer by Id and calls Remove on the CustomerManagement object.
 - DisplayCustomers(): Calls Display on the CustomerManagement object to show a list of all customers.
- **Product Management:**
 - Similar functionalities to Customer Management for adding, updating, removing, and displaying products.
- **Order Management:**
 - Similar functionalities to Customer Management for adding, updating, removing, and displaying orders.
- **ECommerceManagement:**

Provides access to CustomerManagement, ProductManagement, and OrderManagement functionalities through public properties.

- **Properties:**

```
public readonly CustomerManagement customerManagement;
```

```
public readonly ProductManagement productManagement;
```

```
public readonly OrderManagement orderManagement;
```

- **Main Menu:**

- Presents a user-friendly menu with options for Customer Management, Product Management, Order Management, and Exit.
- Validates user input and loops until the user chooses to exit.

```
===== E-commerce Platform =====
1. Customer Management
2. Product Management
3. Order Management
4. Exit
Enter your choice: 1
===== Customer Management =====
1. Add new customer
2. Display all customers
3. Update customer
4. Remove customer
5. Main menu
Enter your choice: 1
===== Add new customer =====
Enter customer name: Cong
Enter customer phone: 0987654321
Enter customer address: Nam Dinh
===== Customer Management =====
1. Add new customer
2. Display all customers
3. Update customer
4. Remove customer
5. Main menu
Enter your choice: 1
===== Add new customer =====
Enter customer name: Van
Enter customer phone: +84943218374
Enter customer address: Ha Noi
===== Customer Management =====
1. Add new customer
2. Display all customers
3. Update customer
4. Remove customer
5. Main menu
Enter your choice: 5
```

```
===== E-commerce Platform =====
1. Customer Management
2. Product Management
3. Order Management
4. Exit
Enter your choice: 2
===== Product Management =====
1. Add new product
2. Display all products
3. Update product
4. Remove product
5. Main menu
Enter your choice: 1
===== Add new product =====
Enter product name: Samsung
Enter product price: 999
Enter product description: samsung
===== Product Management =====
1. Add new product
2. Display all products
3. Update product
4. Remove product
5. Main menu
Enter your choice: 1
===== Add new product =====
Enter product name: iPhone
Enter product price: 899
Enter product description: apple
===== Product Management =====
1. Add new product
2. Display all products
3. Update product
4. Remove product
5. Main menu
Enter your choice: 5
```

```

===== E-commerce Platform =====
1. Customer Management
2. Product Management
3. Order Management
4. Exit
Enter your choice: 3
===== Order Management =====
1. Add new order
2. Display all orders
3. Remove order
4. Main menu
Enter your choice: 1
===== Add new order =====
Id          Name          Phone          Address
1           Cong          0987654321     Nam Dinh
2           Van           +84943218374   Ha Noi
Enter customer id: 1
Id          Name          Price          Description
1           Samsung       999            samsung
2           iPhone       899            apple
Enter product id to add to order (0 to exit): 2
Enter order date (dd/MM/yyyy): 22/04/2024
Enter order note: Cong Order
===== Order Management =====
1. Add new order
2. Display all orders
3. Remove order
4. Main menu
Enter your choice: 2
Id          Customer  Products  OrderDate  Note
1           Cong      1          4/22/2024  12:00:00 AM Cong Order

```

Hints:

- Enhance the user interface with a menu-driven system for better navigation.
- Implement comprehensive validation for user input (e.g., phone number format, product quantity).
- Remember that the Management classes inherit from the abstract base class `Management<T>`. This base class defines the core `Display()` method. Each specific Management class (`CustomerManagement`, `ProductManagement`, `OrderManagement`) can override the inherited `Display()` method to customize the output format based on the model class properties.

Estimated Time: 180 minutes.

Mark Scale:

OOP design	10%	Function requirements	60%
Business rules	15%	Main function	15%