**Question 1:**

(**2 points**)

You are asked to write a console application that includes a **Student class**. This class contains:

* Three private properties **(0.5 point):**
* **int Id**
* **string Name**
* **DateTime DOB**
* Necessary constructors (0.5 point).
* Two public methods
* **int GetAge()** return the age of the Student **(0.5 point)**
* **string ToString()** return the string contains student’s information by the format **(0.5 point):**

“**Student’s Information: ID: xxx – Name: yyy – Age: zzz**”

with xxx is the Id, yyy is the Name and zzz is the age of the Student.

Hint:

Your Student class must match with main function below:

static void Main(string[] args)

{

Student st = new Student(1, "Trung", new DateTime(1990, 11, 10));

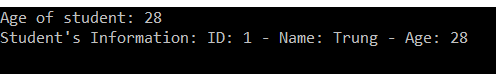
Console.WriteLine("Age of student: " + st.GetAge().ToString());

Console.WriteLine(st.ToString());

Console.ReadLine();

}

You application should display like following:



**Question 2:**

(**2 points**)

You are asked to write a console application that includes a **MyCollection<T> generic class**. This class contains:

* One ArrayList or List or array to store list of object of type T. T can be any built-in data type or any class.
* Nessary constructors **(0.5 point)**
* Three public methods
* **void Add(T obj)** to add the object obj into the above list at last position. **(0.5 point)**
* **void Add(T obj, int k)** to add the object **obj** into the position indexed k in the above list **(0.5 point).**
* **void DisplayItems()** to write to the screen the information of items in the list. You no need to worry about formatting. **(0.5 point).**

Note:

Your class needs to match for the following main function:

The main function of your application should look like following:

static void Main(string[] args)

{

MyCollection<int> IntCollection = new MyCollection<int>();

IntCollection.Add(1);

IntCollection.Add(2);

IntCollection.Add(3);

IntCollection.Add(4, 1);

Console.WriteLine("Display integer list:");

IntCollection.DisplayItems();

MyCollection<string> StringCollection = new MyCollection<string>();

StringCollection.Add("aa");

StringCollection.Add("bb");

StringCollection.Add("cc");

StringCollection.Add("dd", 1);

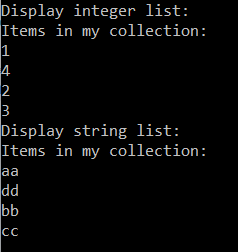
Console.WriteLine("Display string list:");

StringCollection.DisplayItems();

Console.ReadLine();

}

You application should display like following:



**Question 3:**

(**3 points**)

You are given a table in database. Figure 1 describes the table structure.

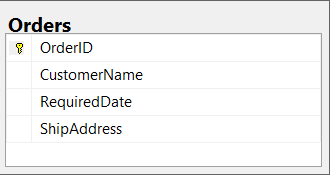


Figure 1: Order Table Schema

You are asked to write a window form application which allow user add order. The UI should look like below picture **(0.5 point).**

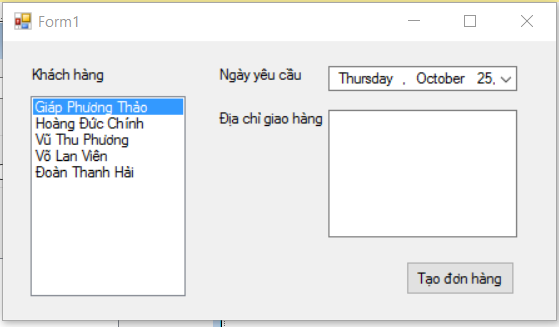


Figure 2: Add Order Screen

Your tasks include several of following steps:

* Customers are listed in a listbox for chosen. Add five items into listbox **like Figure 1**. Set the **first item was selected** when form loaded. **(0.5 point).**
* The listbox must be disable mutiple choices option. Address fieldis a textbox that enable multilines. **(0.5 point)**
* When user click in the button
* The ship address cannot blank. Check if it is blank, show the message box as below (**0.5 point**):

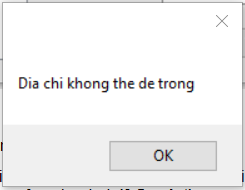


Figure 3: Build-in Message Box to display error message when address is blank

* If address field is not blank, add new order information into the database **(1 point)**.
* When add student successfully, show the message box to note like below (if not, **reduce 0.5 point**):

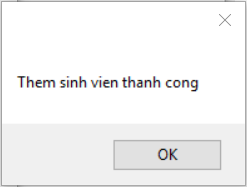


Figure 4: Build-in Message Box to display adding success message

**Note:** (**0 will be given to the work that not using database connection string in App.config)**

**Question 4:**

(**3 points**)

You are given a table in database. Figure 1 describes the table structure.

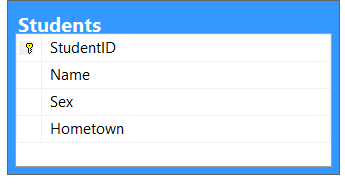


Figure 1: Student Table Schema

You are asked to write a Webform application which allow user filter the students. (**0 will be given to the work that not using database connection string in Web.config)**

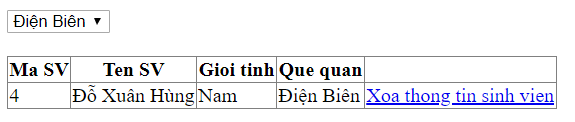


Figure 2: Search Screen

* The page has one Dropdown list, and its data will be loaded from distinct hometown values in Students table. (**0.5 point**)
* When the page is loaded for the first time, the first item of dropdown list was selected. Data of students from this hometown is filled into the Gridview. **(1 point)**
* You must add a link or button into the last column of the Gridview. When user click on these links or buttons, corresponding student info is deleted from database **(0.5 point).**
* When user select one item in Dropdown list, student list in the Gridview changes accordingly **(1 point).**