



The Hashemite University

Prince Al-Hussein bin Abdullah II Faculty for Information Technology

Visual Programming Lab Manual



Prepared by: Ms. Haneen Hijazi

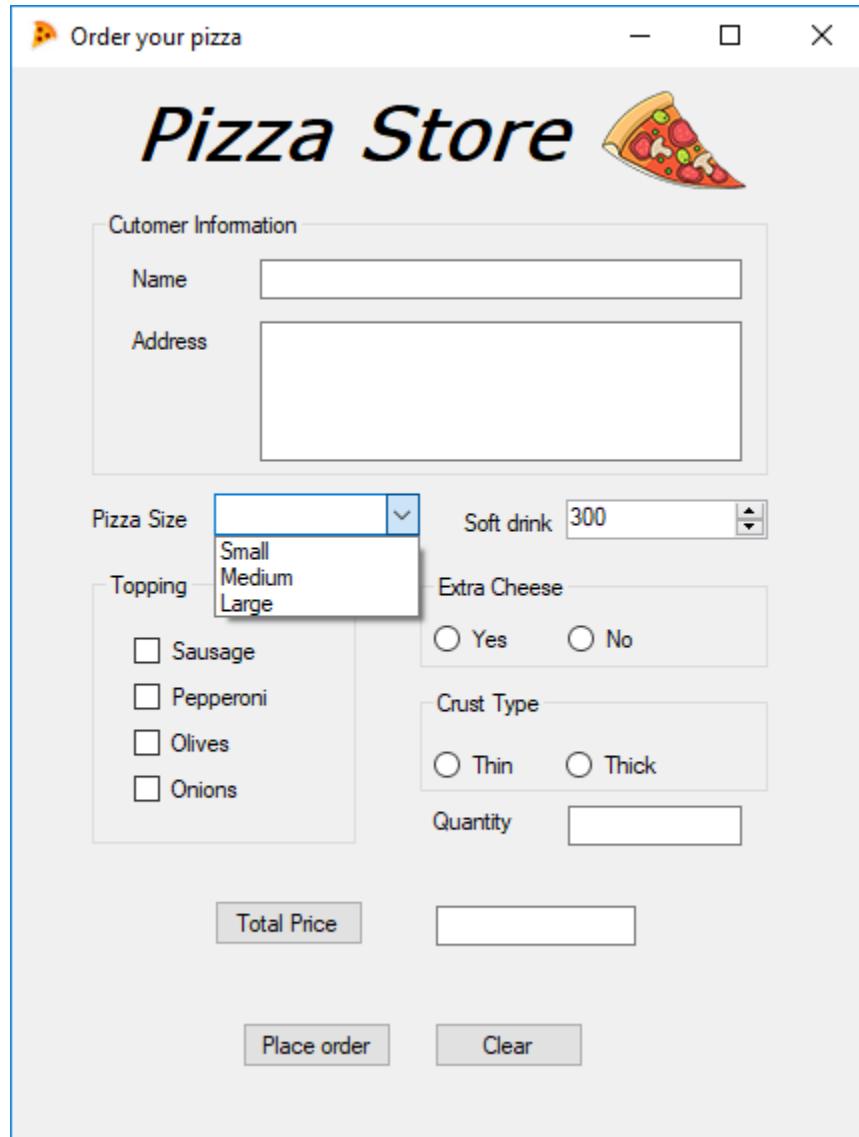
Fall, 2020

Ms. Haneen Hijazi

Working with Basic Controls

Project 1: (Pizza Order Interface)

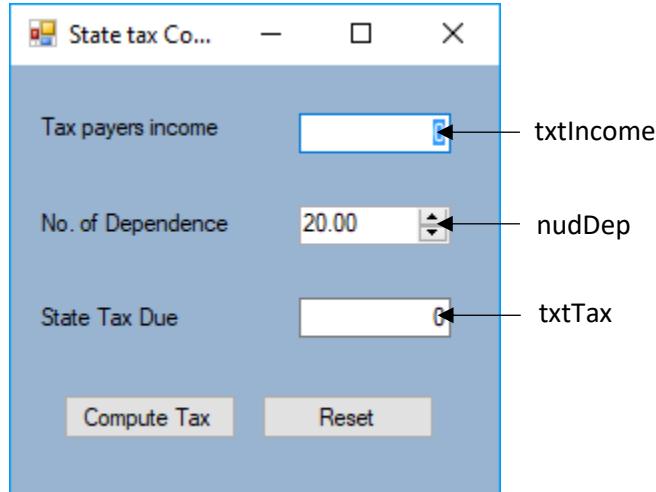
- 1) Design the following GUI Pizza Order application using VB.Net.



Working with Properties

Project 2: (Tax Calculator)

- 1) Design the following GUI for Tax Calculation application using VB.Net.



- 2) Change the properties of the controls as follows:

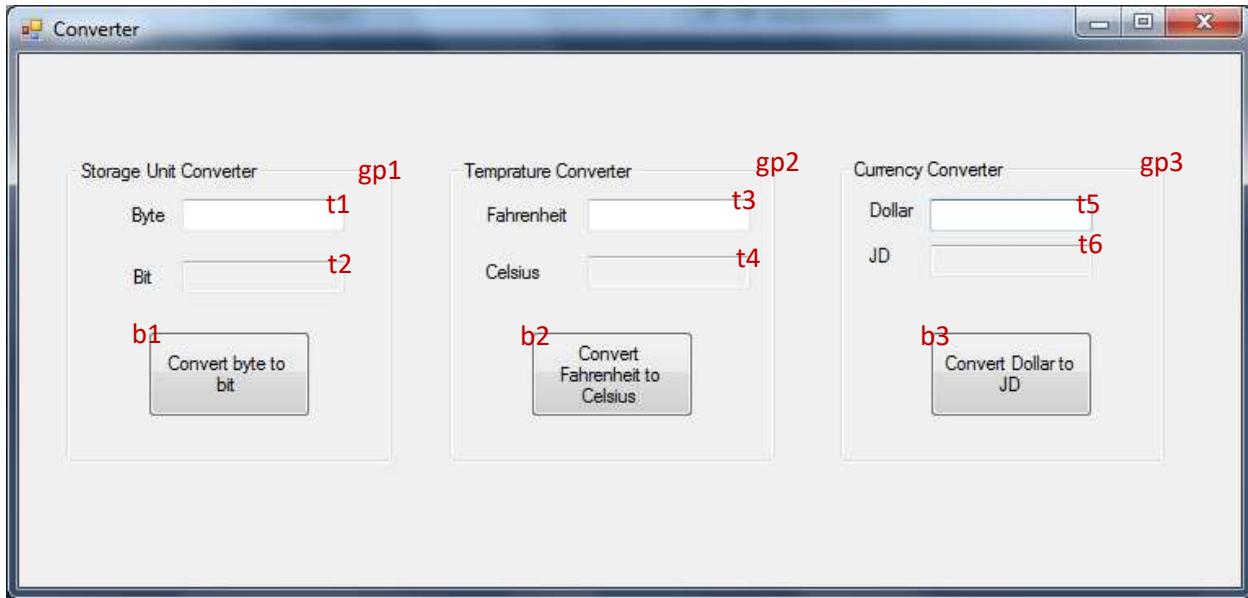
Form	TextBox	NumericUpDown	TextBox
Text: State Tax Computation Back Color: ActiveCaption Location: (144,24) Size: (360, 460)	Name: txtIncome Text: 0 Location: (144,24) Size: (76, 20) Text Align: right Fore color: red	Name: nudDep Location: (144,70) Size: (76, 20) Minimum: 2 Maximum: 30 Value: 20 Decimal places: 2 Increment: 1.03	Name: txtTax Text: 0 Location: (144,112) Size: (76, 20) Text Align: right Read only: true Tab Stop: False

- 3) Add a click event handler on button "Compute Tax" to calculate the tax based on the following formula:
$$\text{Tax} = 0.03 * (\text{income} - 600 * \text{number of dependence})$$
- 4) Add a click event handler on button "Reset" to reset the input values to their defaults.

Working with Events

Project 3: (Converter)

- 1) Design the following GUI for a converter application using vb.net



- 2) Add a click events handlers on buttons:

- B1: converts from bytes and displays the result in t2.
 - Hint: byte=8bits
- B2: converts from Fahrenheit to Celsius.
Hint: $\text{Celsius} = \frac{5}{9} * (\text{Fahrenheit} - 32)$
- B3: converts from dollars to JD.
 - Hint: $\text{JD} = 0.71 * \text{dollars}$

Hint: use val() method to get the numerical values of textboxes

- 3) Add mouse move events handlers on buttons:

- B1: changes the back color of the group box gp1 to “Rosy Brown”.
- B2: changes the back color of the group box gp2 to “Light Blue”.
- B3: changes the back color of the group box gp3 to “Light Pink”.

- 4) Add mouse leave events handlers on buttons:

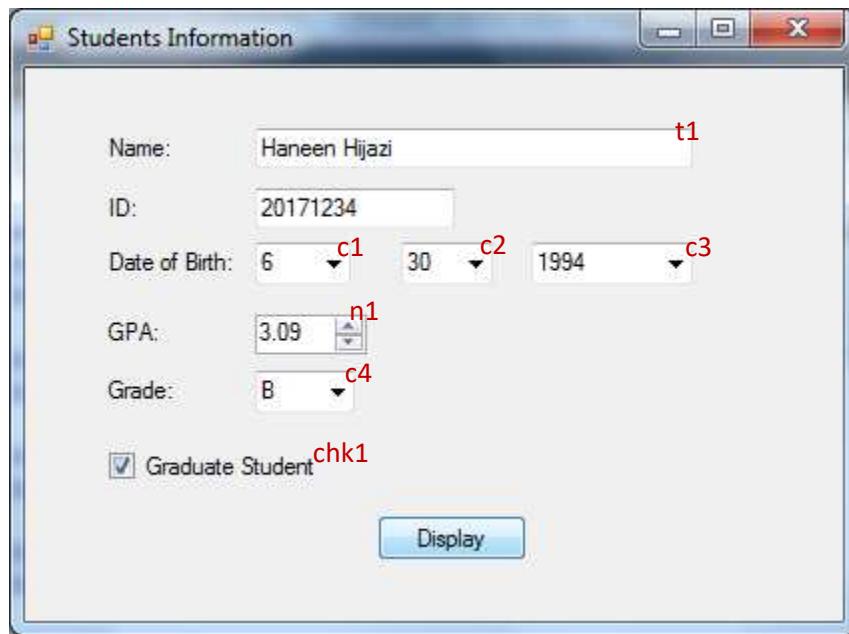
- B1: changes the back color of the group box gp1 to the form back color.
- B2: changes the back color of the group box gp2 to the form back color.
- B3: changes the back color of the group box gp3 to the form back color.

- 5) Add a form load event handler to display a message box with the text “Welcome to Converter Application”.

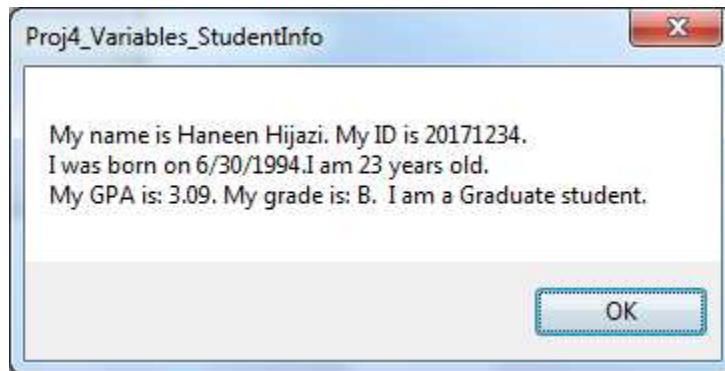
Working with Variables

Project 4: (Student Information)

- 1) Design the following GUI for a student information application using vb.net



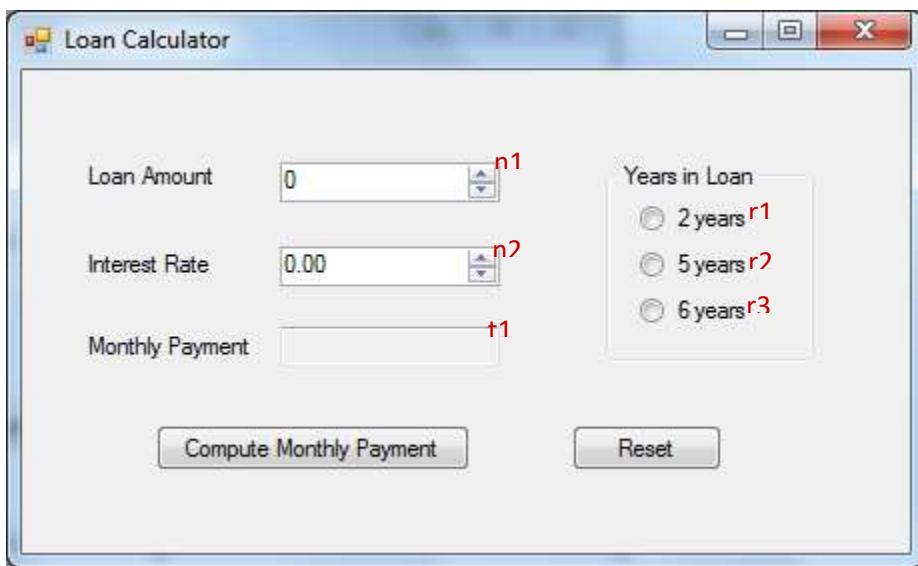
- 2) On button "Display", add a click event that saves all form data values in variables with proper data types. Then use these variables to display a paragraph in a message box as follows:
- 3) Hint: use val method with t2.



Working with Variables

Project 5: (Loan Calculator)

- 1) Design the following GUI for a Loan calculator application using vb.net



- 2) Modify the properties for the following controls as follows:
 - a. N1: minimum=0, maximum=20000, increment=1000
 - b. N2: minimum=0, maximum=100, increment=0.01
 - c. T1: read only
- 3) Declare a global double value called payment.
- 4) For each radio button add a checkChanged event handler to change the value of the payment variable as follows:
 - a. R1 checkChanged: payment = 2*12
 - b. R2 checkChanged: payment = 5*12
 - c. R3 checkChanged: payment = 6*12
- 5) On button "Compute Monthly Payment", add a click event handler to calculate Monthly Payment and display it on t1.

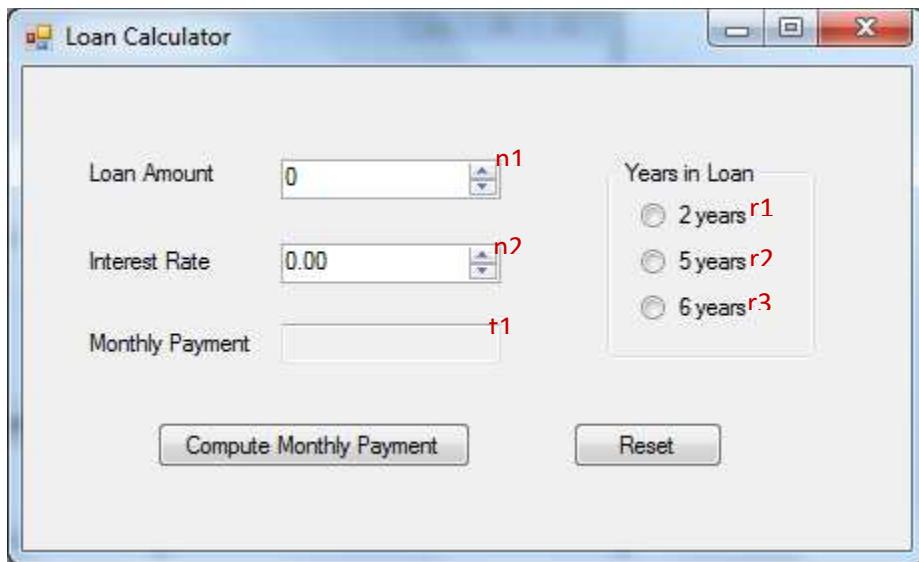
Monthly Payment=(Interest Rate*(1+interest rate)^payment)/((1+interest rate)^payment-1)*loan amount

- 6) On button "Reset", add a click event handler to reset all controls to its default values.

Selection Statement

Project 6: (Loan Calculator IF)

- 1) Rewrite the previous application using IF statement.



- 2) Modify the properties for the following controls as follows:
 - a. N1: minimum=0, maximum=20000, increment=1000
 - b. N2: minimum=0, maximum=100, increment=0.01
 - c. T1: read only
- 3) On button “Compute Monthly Payment”, add a click event handler to calculate Monthly Payment and display it on t1, as follows
 - a. If R1 is checked: payment = 2*12
 - b. If R2 is checked: payment = 5*12
 - c. If R3 is checked: payment = 6*12
- 4) Calculate monthly payment according to the formula:

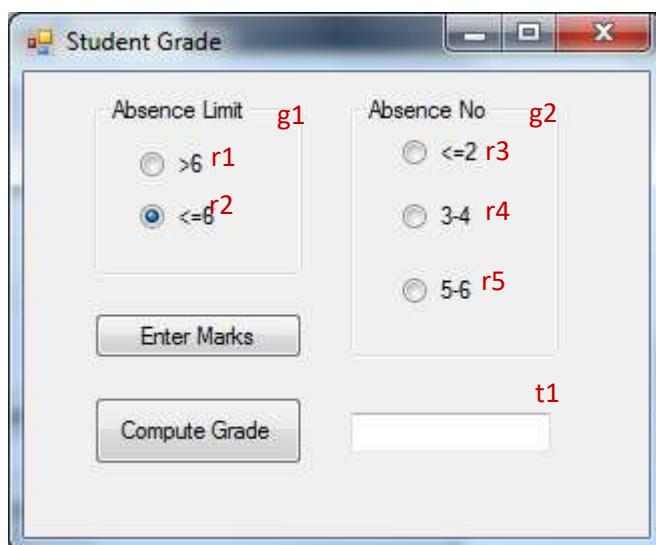
Monthly Payment=(Interest Rate*(1+interest rate)^payment)/((1+interest rate)^payment-1)*loan amount

- 5) On button “Reset”, add a click event handler to reset all controls to its default values.

Selection Statements

Project 7: (Student Grade)

- 1) Design the following GUI for a Student Grade application using vb.net



- 2) On button “Enter marks”, add a click event handler that allows user to enter three marks; first, second and final.
- 3) Add click event handlers on r1 and r2:
 - a. If r1 is checked, disable g2
 - b. If r2 is checked, enable g2
- 4) On button “Compute Grade”, add an event handler that compute the sum of first, second, final, and absence marks, and then find the grade and display it.
 - a. If user did not check absence limit a warning should appear
 - b. If absence limit >6, then grade = “F”
 - c. If absence limit <=6, then user should select absence number
 - i. If user did not check absence No. a warning should appear
 1. if <=2, absmarks=10
 2. if 3,4, absmarks=5
 3. if 5,6, absmarks=2
 - d. sum = first + second + final + absence
 - e. if sum=100, grade ="A"
 - f. if sum=80, grade ="B"
 - g. if sum=70, grade ="C"
 - h. if sum>=65, grade = d
 - i. otherwise: grade="F"

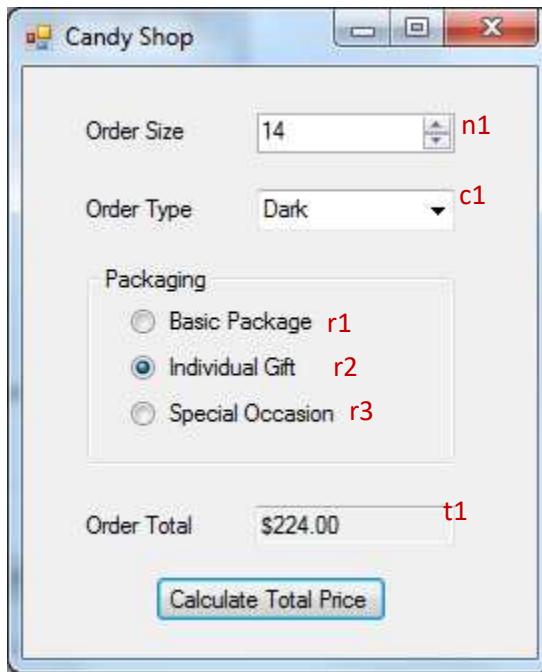
- 5) display grade result on textbox t1

Selection Statements

Combobox and RadioButtons

Project 8: (Candy Shop)

- 1) Design the following GUI for a Candy Shop application using vb.net



- 2) Candy shop has a minimum order size of 10 boxes, and maximum order size of 1000 boxes.
Make this initialization in the form load event handler.
- 3) Candy shop has five types of chocolate: Mint, White, Dark, Swiss and Cherry.
- 4) On button “Compute Total price”, add an event handler to calculate total price of an order as follows.
 - a. The prices per box for each type respectively are: 12.5, 12, 13.5, 19.4 and 14.5.
 - Hint: use select statement to assign prices and selectedIndex method
 - b. If the user did not choose any type, an error message should be displayed.
 - c. If the order is packaged an additional value will be added to the price as follows:
 - i. Basic package: no added value
 - ii. Individually gift: 2.5
 - iii. Special Occasion: 4.7
 - d. If the user did not choose any packaging option, an error message should be displayed.
 - e. Calculate and display the total price in Order Total text box in currency format according the following formula:

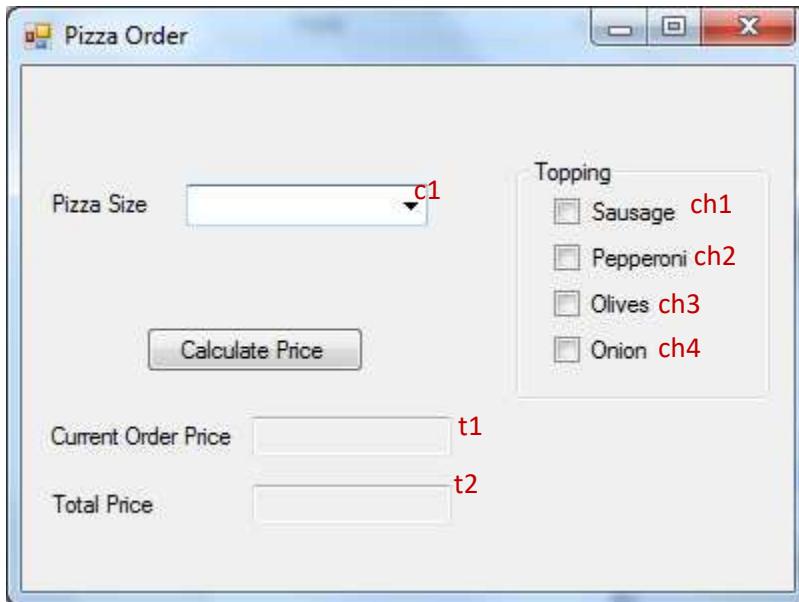
$$\text{Total price} = (\text{price per type box} + \text{packaging price}) * \text{order size}$$

Selection Statements

Check Boxes

Project 9: (Pizza Order Pricing)

- 5) Design the following GUI for a Pizza Order application using vb.net



Q1) On form load, add the items “Small”, “Medium” and “Large” to combobox c1

Q2) On button “Calculate Price”, write an event handler that allows the customer to make several pizza orders, calculate and display the price for each order and the total price, based on the following:

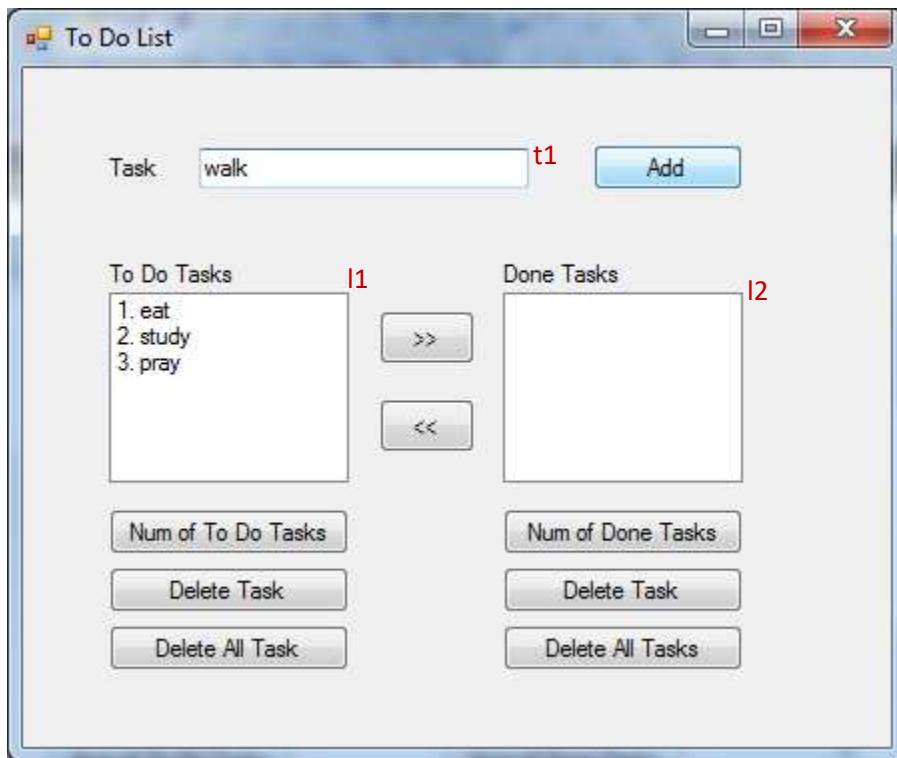
(Hint: use selectedItem method)

- The price of small sized pizza = 2
- The price of medium sized pizza = 4
- The price of large sized pizza = 6
- If the customer selects Sausage, 0.75 will be added to the price.
- If the customer selects Pepperoni, 0.75 will be added to the price.
- If the customer selects Olives, 0.5 will be added to the price.
- If the customer selects Onions, 0.25 will be added to the price.
- After each order a messagebox should be displayed to inform the customer that he can make another order.

List Boxes

Project 10: (To Do List)

- 1) Design the following GUI for a To Do List application using vb.net



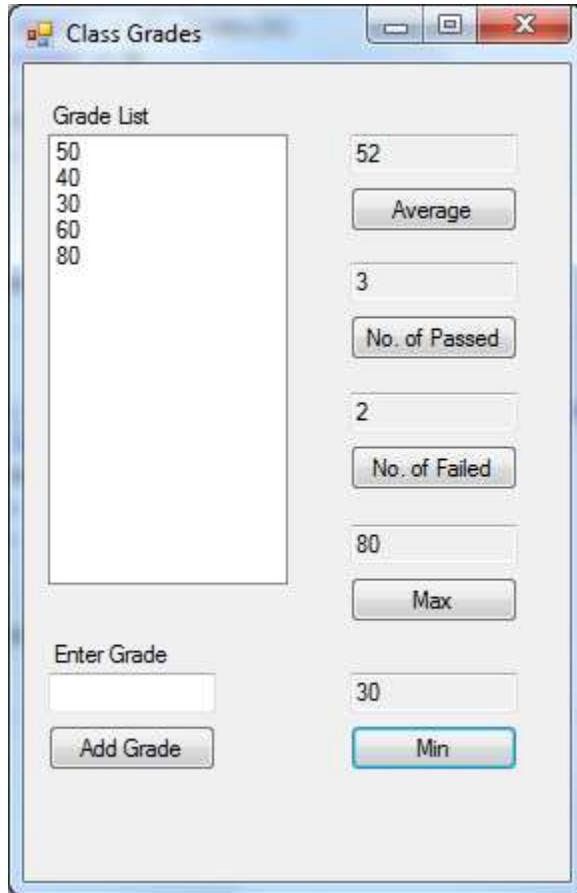
Add the following event handlers:

- 2) On clicking on “Add” button, the task inserted in task text box will be added to “To Do Tasks” list box in numbered format.
- 3) On clicking on “>>” button, the selected to do task will be moved to done tasks list.
- 4) On clicking on “<<” button, the selected done task will be returned back to “To Do Tasks” list.
- 5) On clicking on “Num of To Do Tasks” button, the number of tasks in “To Do Tasks” list will be displayed in a message box.
- 6) On clicking on “Num of Done Tasks” button, the number of tasks in “Done Tasks” list will be displayed in a message box.
- 7) On clicking on “Delete Task” button (on the left), the selected to do task will be deleted. (Hint: use remove procedure)
- 8) On clicking on “Delete Task” button (on the right), the selected done task will be deleted. (Hint: use removeAt procedure)
- 9) On clicking on “Delete All Task” button (on the left), all to do tasks will be deleted.
- 10) On clicking on “Delete All Task” button (on the right), the all done tasks will be deleted.

Repetition Statements

Project 11: (Class Grades)

- 1) Design the following GUI for a Class Grades application using vb.net

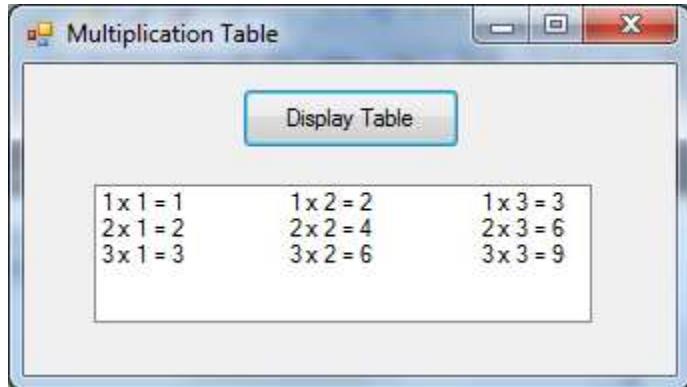


- 2) On clicking on “Add Grade”, the grade inserted in “Enter grade” text box will be added to the grade list.
- 3) On clicking on “Average” button, the average of grades in the list will be displayed in the above text box. (Hint: use for statement)
- 4) On clicking on “No. of Passed” button, the number of passed students will be displayed in the above text box. (Hint: use while statement)
- 5) On clicking on “No of Failed” button, the number of failed students will be displayed in the above text box. (Hint: use Do while statement)
- 6) On clicking on “Max” button, the maximum grade will be displayed in the above text box. (Hint: use Do until statement)
- 7) On clicking on “Min” button, the minimum grade will be displayed in the above text box. (Hint: use for each statement)

Nested Loops

Project 12: (Nested Loops)

- 1) Design the following GUI for a Multiplication Table application using vb.net



- 2) On clicking on “Display Table” button, the ,multiplication table for integers from 1 to 3 will be displayed in the list box lb1
- 3) The entries should not be duplicated each time you click the button.

Multiple-Forms and List View

Project 13: (Employees Salaries)

- 1) Create a project with the following three forms:

frmemployees

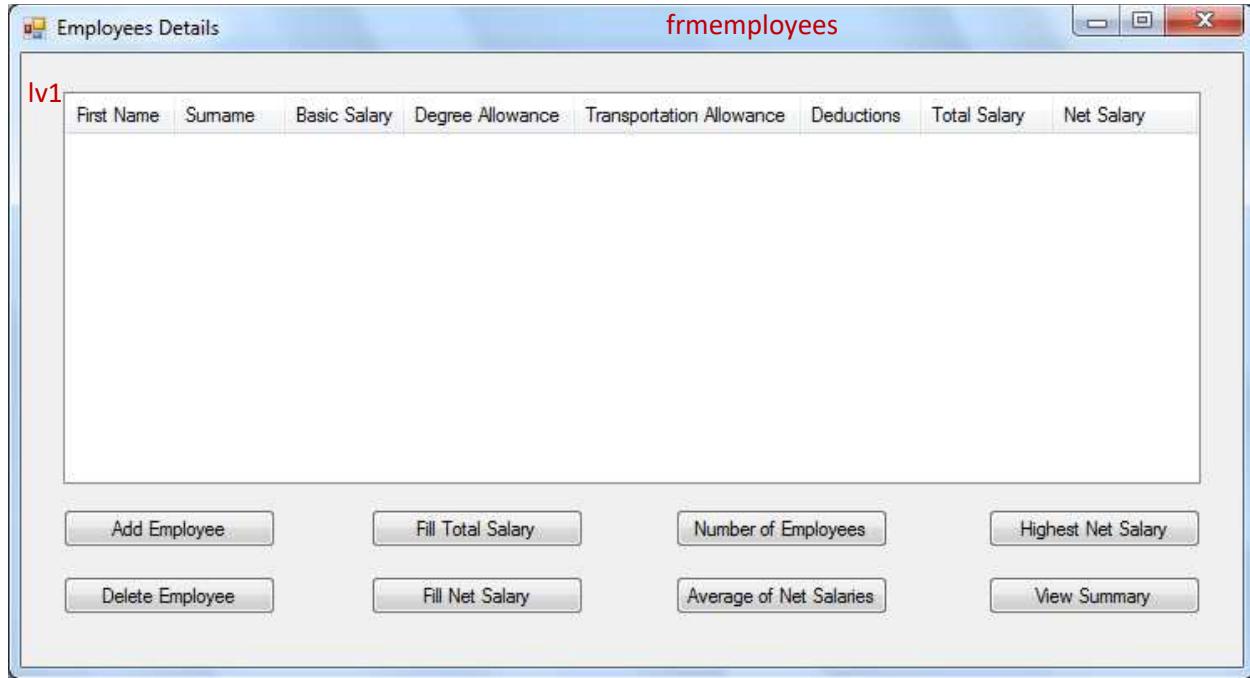
Employees Details

lv1

First Name	Surname	Basic Salary	Degree Allowance	Transportation Allowance	Deductions	Total Salary	Net Salary

Add Employee Fill Total Salary Number of Employees Highest Net Salary

Delete Employee Fill Net Salary Average of Net Salaries View Summary



frmaddemployee

Add Employee

First Name	t1	<input type="text"/>
Last Name	t2	<input type="text"/>
Basic Salary	t3	<input type="text"/>
Degree Allowance	t4	<input type="text"/>
Transportation Allowance	t5	<input type="text"/>
Deductions	t6	<input type="text"/>

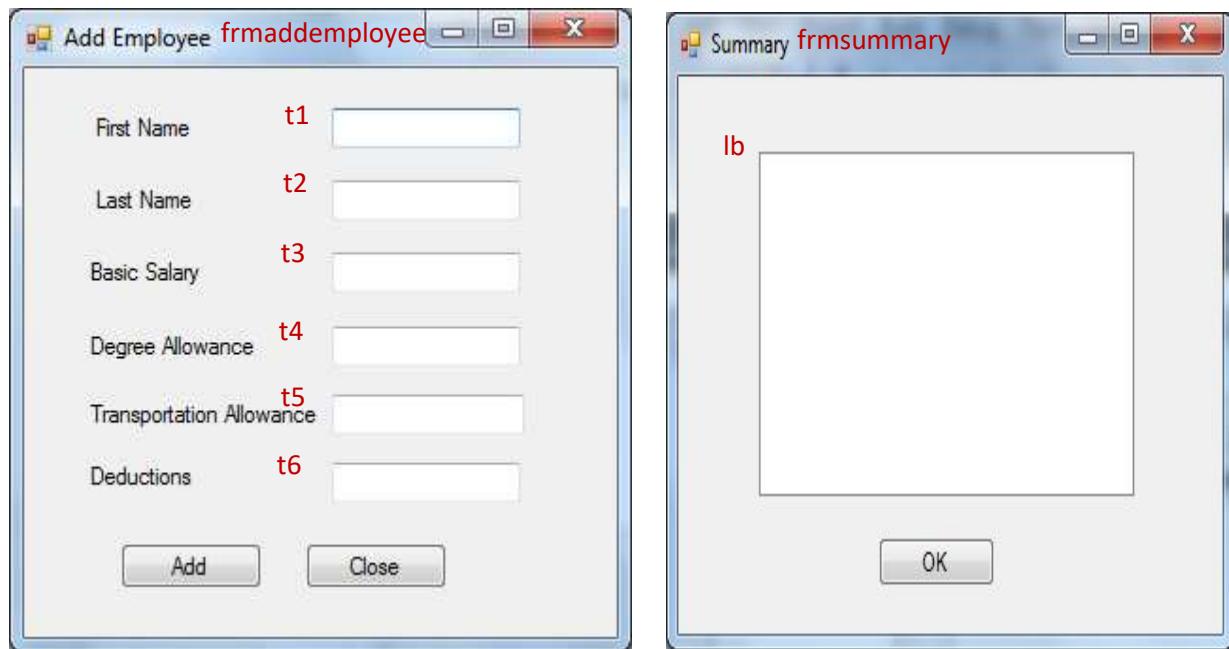
Add Close

frmsummary

Summary

lb

OK



- 2) On clicking “Add Employee” button, the form “frmaddemployee” should be displayed.
- 3) On clicking “Add” button, a new employee should be created and added to “lv1”. All text boxes should be filled before adding the employee.
- 4) On clicking “Close” button, the form “frmaddemployee” should close.
- 5) On clicking “Delete Employee” button, the selected employee will be deleted.
- 6) On pressing “Delete Key”, the selected employee should be deleted.
- 7) On clicking “Fill Total Salary” button, the total salary will be calculated for each employee and displayed in the column “Total Salary”.

Total Salary = Basic Salary + Degree Allowance + Transportation Allowance

- 8) On clicking “Fill Net Salary” button, the net salary will be calculated for each employee and displayed in the column “Net Salary”.

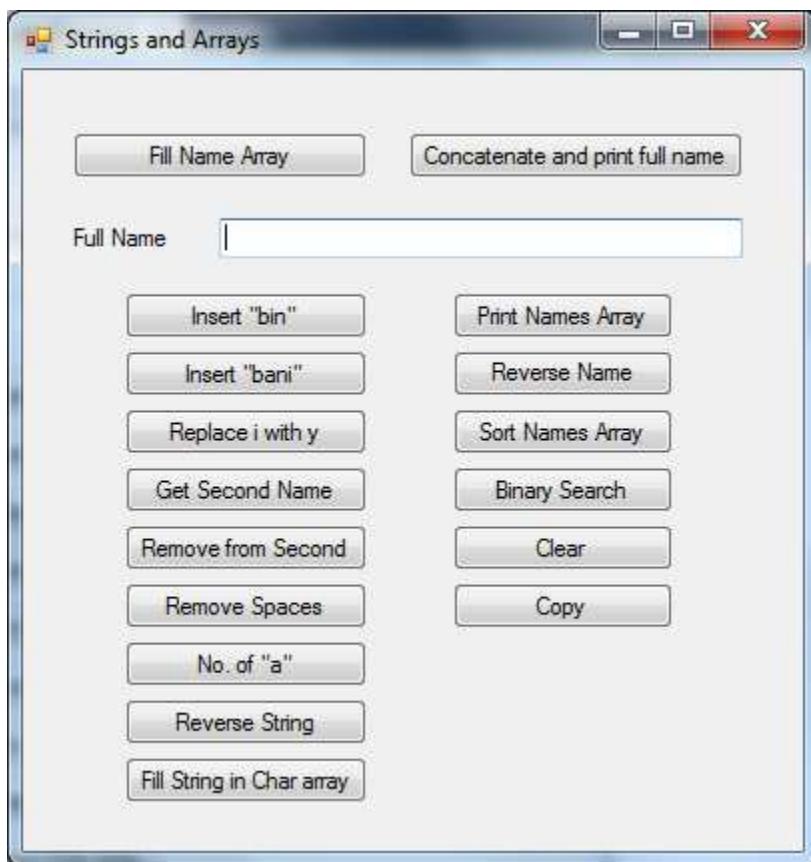
Net Salary = Total Salary - Deductions

- 9) On clicking “Number of Employees” button, the number of employees will be displayed in a message box.
- 10) On clicking on “Average Net Salaries” button, the average of net salaries will be calculated and displayed in a message box.
- 11) On clicking on “Highest Net Salaries” button, the highest net salary will be calculated and displayed in a message box.
- 12) On clicking on “View Summary” button, the form “frmsummary” should be displayed. The list box in this form should contain the first Name, Surname, and net salary for each employee.
- 13) On clicking “OK” button, the form “frmsummary” should close.

Strings and Arrays

Project 14: (Full Name Manipulation)

- 1) Design the following GUI.



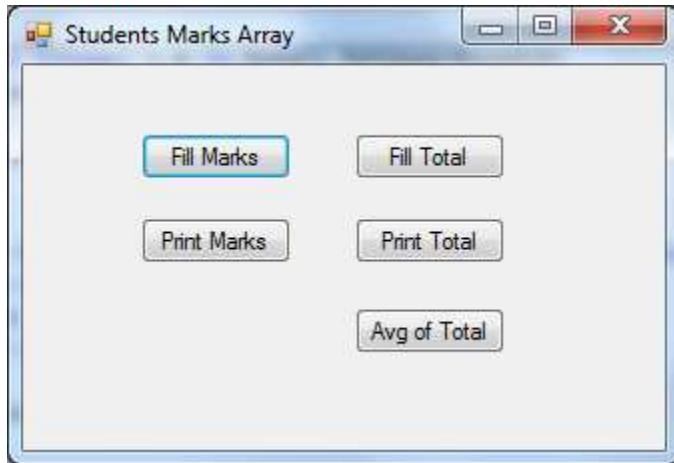
- 2) Declare a global one-dimensional string array “namesarr” with size =3.
- 3) On clicking “Fill Name array” button, “namesarr” will be filled with first name, second name, and family name consequently through 3 input boxes.
- 4) Declare a global string variable “fullname”
- 5) On clicking “Concatenate and print full name” button, each array element will be concatenated to the fullname string and finally displayed on the text box t1.
- 6) On clicking “Insert bin” button, the string “bin” will be inserted after the first name. This insertion **should be reflected on the original fullname string**.
- 7) On clicking “Insert bani” button, the string “bani” will be inserted before the last name. This insertion **should be reflected on the original fullname string**.
- 8) On clicking “Replace i with y” button, replace each occurrence of “i” withy
- 9) On clicking “Get bin” button, the substring “bin” will be displayed in a message box.
- 10) On clicking “Remove bin” button, the fullname will be displayed in a message box without “bin”.

- 11) On clicking “Remove Spaces” button, all spaces in the string will be removed.
- 12) On clicking “No of a” button will display the number of occurrences of the character “a” in fullname.
- 13) On clicking “Reverse” button, the string fullname will be reversed and displayed in a message box.
- 14) On clicking “fill String in char array” button, the characters of the string “fullname” will be filled in a character array (carr) and their ascii code will be filled in an integer array “intarr”.
- 15) On clicking “Print name array” button, the content of the namesarr will be displayed in a single message box separated by newlines.
- 16) On clicking “Reverse Name” button, the content of “namesarr” will be reversed.
- 17) On clicking “Sort Names” button, the content of “namesarr” will be sorted ascending.
- 18) On clicking “Binary Search” button, the index of a specific character in a sorted array will be displayed in a message box.
- 19) On clicking “Clear” button, the content of “namesarr” will be reset to its default values.
- 20) On clicking “Copy” button,
 - a. Define a Stringarray newarr and initialize it with “a”, “b”, “c”
 - b. The content of “newarr” will be copied into namesarr starting from position 0.

Two Dimensional Arrays

Project 15: (Students Marks Array)

- 1) Design the following GUI.



- 2) On application startup, an input box should be displayed to enter the number of students
- 3) Define a global two-dimensional array “marks” to contain first, second and final marks for each student.
- 4) Define a global one-dimensional array “totals” to contain the total of first, second and final marks.

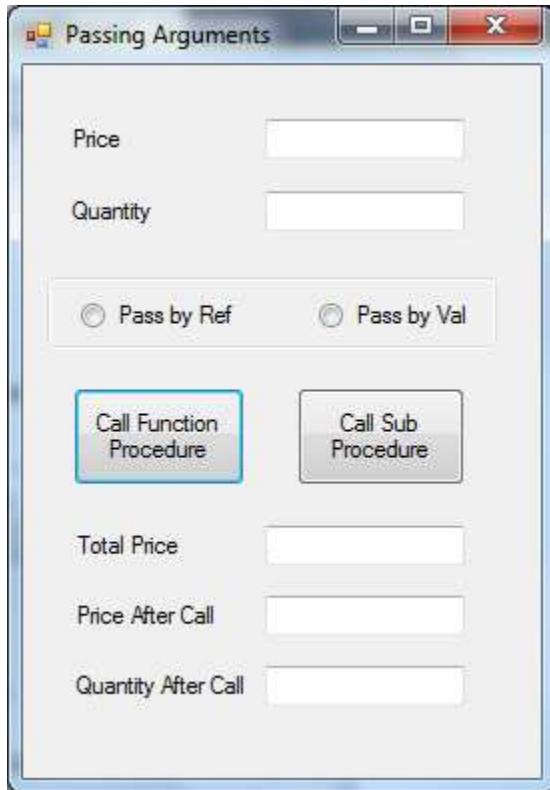
	1 st	2 nd	final	Totals
St0				
St1				
St2				

- 5) On clicking “Fill Marks” button, the user would enter first, second and final marks for each student using input boxes.
- 6) On clicking “Print Marks”, the content of the marks array will be printed in a message box in a tabular format.
- 7) Write a sub procedure fillTotals that finds the summation of the first, second and final marks for each student saves it in the corresponding position in array “totals”.
- 8) On clicking on “Fill Totals” button, the sub procedure fillTotals will be called.
- 9) Write a sub procedure printTotals that prints the content of the totals array in one message box.
- 10) On clicking “Fill Total” button, the sub procedure printTotals will be called.
- 11) Write a function getAvg, that calculates the average of the values in the totals array.
- 12) On clicking on “Average of Total” button, the function getAvg will be called and the average value will be displayed in a message box.

Procedures and Passing Arguments

Project 16: (Passing)

- 1) Design the following GUI.



- 2) Write a function procedure **calculateTotal_1** that accepts two arguments (price and quantity) by reference and returns the total price as double. Total price = $\text{price} * \text{quantity}$. After calculating total price the procedure should set the values of price and quantity to 0.
- 3) Write a function procedure **calculateTotal_2** that accepts two arguments (price and quantity) by value and returns the total price as double. Total price = $\text{price} * \text{quantity}$. After calculating total price the procedure should set the values of price and quantity to 0.
- 4) Write a sub procedure **calculateTotal_3** that accepts three arguments (price, quantity and total) by reference and calculates the total price. Total price = $\text{price} * \text{quantity}$. After calculating total price the procedure should set the values of price and quantity to 0.
- 5) Write a sub procedure **calculateTotal_4** that accepts three arguments (price, quantity and total) by value and calculates the total price. Total price = $\text{price} * \text{quantity}$. After calculating total price the procedure should set the values of price and quantity to 0.
- 6) On clicking on “**Call Function Procedure**” button,
 - a. if **Pass by Ref** was selected, the procedure **calculateTotal_1** will be called
 - b. if **Pass by Val** was selected, the procedure **calculateTotal_2** will be called
 - c. The values of total price, price and quantity after calling the function will be displayed in the corresponding textboxes.
- 7) On clicking on “**Call Sub Procedure**” button,

- a. if **Pass by Ref** was selected, the procedure **calculateTotal_3** will be called
- b. if **Pass by Val** was selected, the procedure **calculateTotal_4** will be called
- c. The values of total price, price and quantity after calling the sub procedure will be displayed in the corresponding textboxes.

Connecting with Database

Project 17: (AddressBook)

- 1) Create an access DB named (AddressBook) with one table named “contacts”
- 2) In the “contacts” table create the following datafields:

Field Name	Data Type
ID	AutoNumber
FirstName	Short Text
LastName	Short Text
email	Short Text

- 3) Create the following GUI for an Address Book Application

ID	FirstName	LastName	email
17	suha	ahmad	s@hotmail.com
18	haneen	hijazi	haneen@hu.edu
19	ola	ali	o@yahoo.com

First: suha Last: ahmad Email: s@hotmail.com

Next: 17 Previous: Insert Delete

- 4) On form load,
 - a. All records in the table will be displayed in the DataGridView.
 - b. The values of the first record will be displayed in the textboxes.
- 5) On clicking on “Next” button, the values of the next record in the table will be displayed
- 6) On clicking on “Previous” button, the values of the previous record in the table will be displayed
- 7) On clicking on “Insert” button, the values in the texboxes will be added to the table as a new record.
- 8) On clicking on “Delete” button, the record with the ID displayed in the textbox will be deleted.

