

Laboratory Exercise

Creating Text File

Objectives:

At the end of the exercise, the students should be able to:

- Specify options in handling files; and
- Use predefined classes in writing to and reading from a file.

Software Requirement:

- Visual Studio IDE 2015 or higher

Procedures:

1. Create a program that generates a basic text file. This program should have two (2) windows form named **FrmLab1** and **FrmFileName**. See *Figures 1 and 2*.

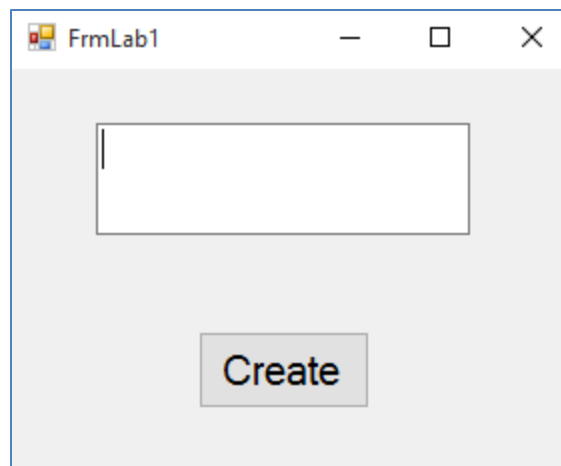


Figure 1. FrmLab1

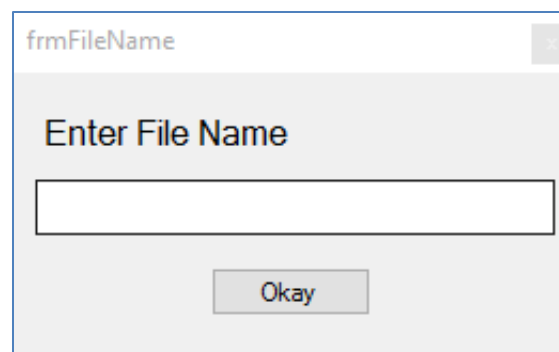


Figure 2. FrmFileName

2. Follow the given design in *Figures 1 and 2*. See *Table 1* for the names of the control that contains function.

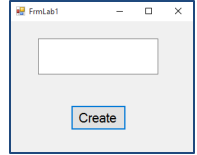
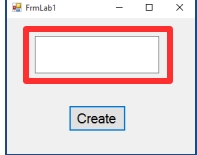
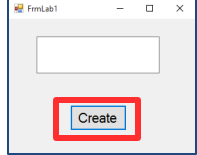
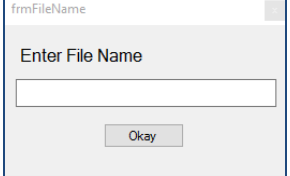
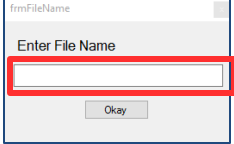
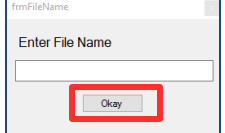
Forms/Controls	Name	
Form	FrmLab1	
TextBox	txtInput	
Button	btnCreate	
Form	FrmFileName	
TextBox	txtFileName	
Button	btnOkay	

Table 1. Forms and controls

3. In FrmLab1, double click the **Create** button and write the following syntax that are needed:
 - a. Declare an instance **FrmFileName** and call **ShowDialog()** method.
 - b. Declare a string variable named **getInput** and get the value of the **TextBox** named **txtInput**.

```

string docPath =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

using (StreamWriter outputFile = new StreamWriter(Path.Combine(docPath,
FrmFileName.SetFileName)))
{
    outputFile.WriteLine(getInput);
    Console.WriteLine(getInput);
}

```

4. Go to **FrmFileName** and click **Okay** to generate an event click.
5. Inside the event, click the **Okay** button. Set the value of the static string **SetFileName** by calling the value of **txtFileName**, and concatenate it with ".txt".
6. Call the **Close()** method afterward.

Note: You need to create a public static string above the constructor for **SetFileName** of **FrmFileName**.

7. Run the program to check the output.

Note: You might encounter some errors, so analyze and understand the code.

Challenge Exercise:

1. In this challenge exercise, you will create another form named **FrmRegistration**. See Figure 3.

The screenshot shows a Windows form titled 'FrmRegistration'. The form has a title bar with standard window controls. The main area is titled 'Registration'. It contains the following fields: 'Student No.' with text '10000242589', 'Program' with a dropdown menu showing 'BS Information Technology', 'Last Name' with text 'Cruz', 'First Name' with text 'Mary', 'M.I.' with text 'M.', 'Age' with text '21', 'Gender' with a dropdown menu showing 'Female', 'Birthday' with a date picker showing 'Thursday, 15 July 1999', and 'Contact No.' with text '09090998524'. At the bottom center is a 'Register' button.

Figure 3. FrmRegistration

2. This program should generate a text file same as above. The file name of the text should be set according to its student number.

The screenshot shows a Notepad window titled '10000242589 - Notepad'. The text inside is: Student No.:10000242589, Full Name:Cruz, Mary, M., Program: BS Information Technology, Gender: Female, Age: 21, Birthday: 1998-07-04, Contact No.9090998524.

Figure 4. RegistrationTextFile

3. Run the program to check the output. See Figure 4 for the sample output of text file.
Hint: Create an array of string and use foreach to display all the content inside the writeLine method.
4. Inform your instructor once you're done with the Challenge Exercise as this will be recorded.
5. Save your file in a folder with the filename **LastName_FirstName_LabStream** (ex. Cruz_Christian_LabStream). This will be collected by your instructor.

Grading Rubric:

CRITERIA	PERFORMANCE INDICATORS	POINTS
Correctness	The code produces the expected result.	30
Logic	The code meets the specifications of the problem.	30
Efficiency	The code is concise without sacrificing correctness and logic.	20
Syntax	The code adheres to the rules of the programming language.	20
Total		100