IT1753 Principles of Computing PCMP

Topic 5A

Looping for Repetitive Tasks
Storing Data into Text Files

Topics

Objectives:

- Be able to use for and while constructs for repetitive tasks

- Storing data in files
- Reading data from text files
- Writing data to text files

Need for Repetition

- In computer programs, as in our every life, we encounter repetitive tasks.
- For example, we wake up, freshen ourselves, take break fast, go to school and back home… and
- .. It repeats the next day.

- It is the same for computer programs.
- Let us take a look at a simple example.

IT1753 Principles of Computing PCMP

Example of repetitive problem

- **Task** : Add the numbers 1,2,3,4,5. Display the sum
- In this case, the task to be accomplished repeatedly is the addition operation:

```
  int sum = 0;
  sum = sum + 1;
  sum = sum + 2;
  sum = sum + 3;
sum = sum + 4;
  sum = sum + 5;
  lblDisplay.Text = sum.ToString();
```

- Evidently, the above approach is not desired. What if we need to add 1 to 500? We can accomplish this task more efficiently using a ***for*** statement.

IT1753 Principles of Computing PCMP

Using for Statement

F

O

R

- The for statement has the following syntax:

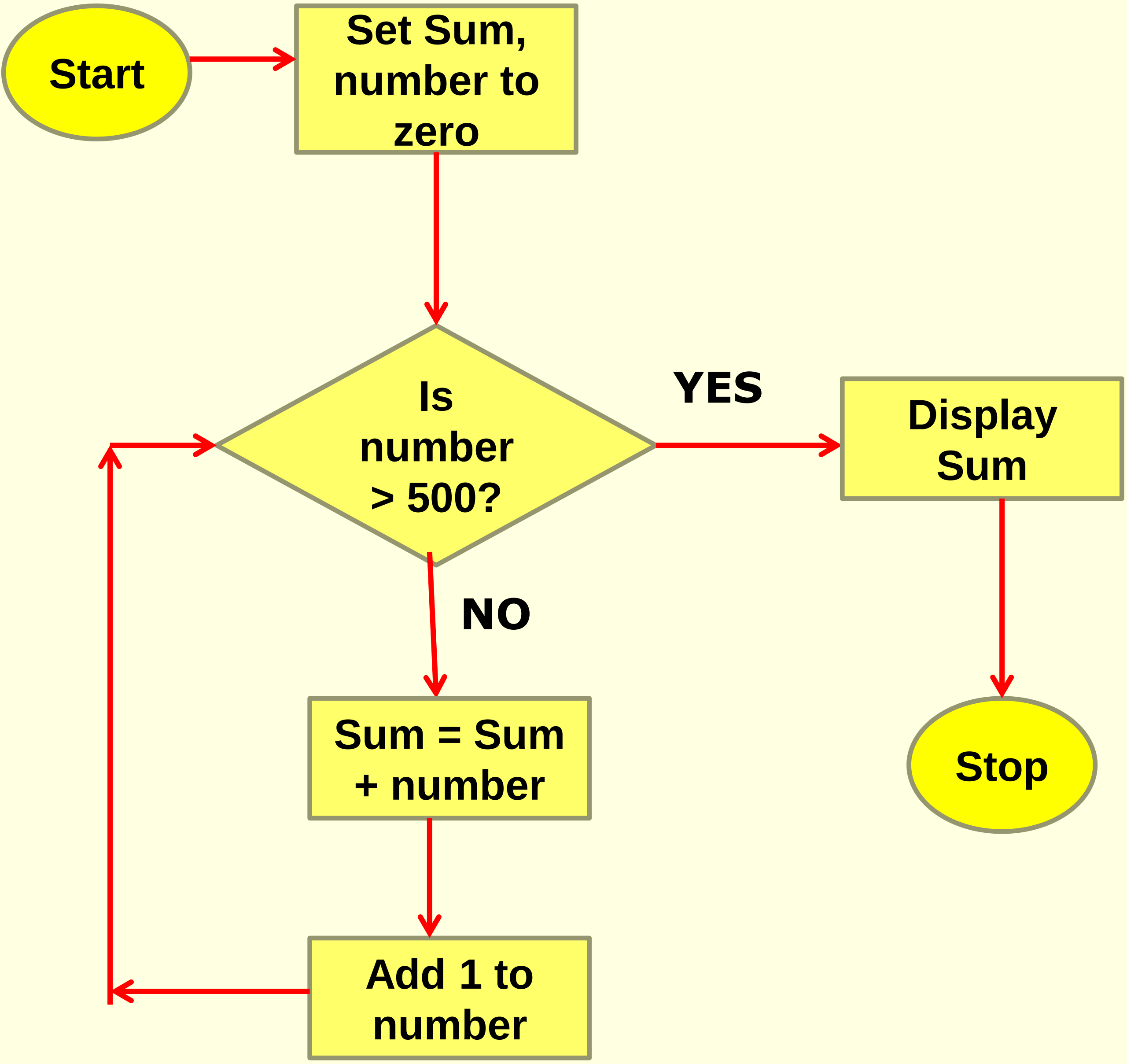
```
for ( initialise; condition; increment)
{
    statement1
    statement2 ....
}
```

where:

for	: compulsory keyword.
Intialise	: start value of loop counter.
Condition	: condition to test whether to loop.
increment	: value to increase loop counter.

Looping Explained with Flow Chart

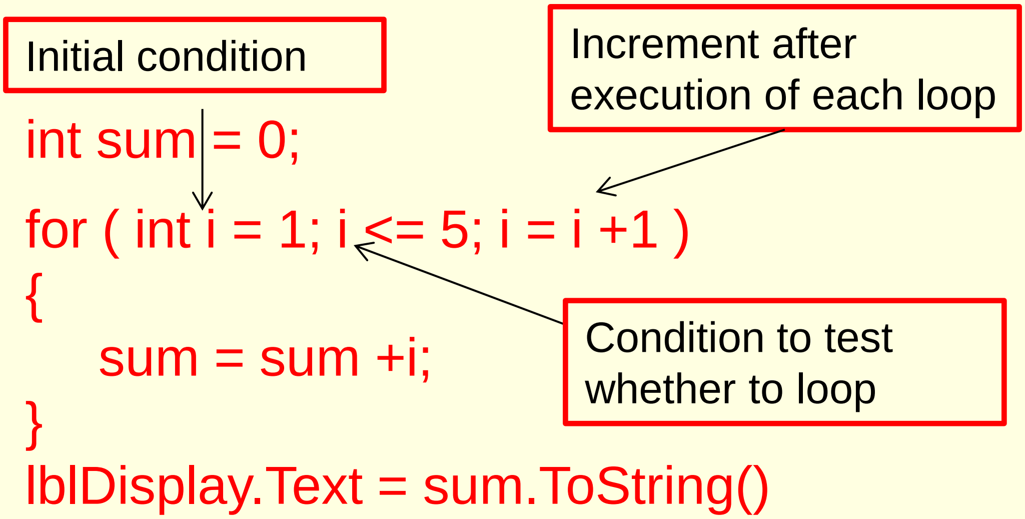
Add the numbers 1,2,3,4,5…. 500. Display the sum



Using for Statement

F
O
R

- Returning to the problem of adding 1,2,3,4,5, we can rewrite the code as follows:



Initial condition

Increment after execution of each loop

Condition to test whether to loop

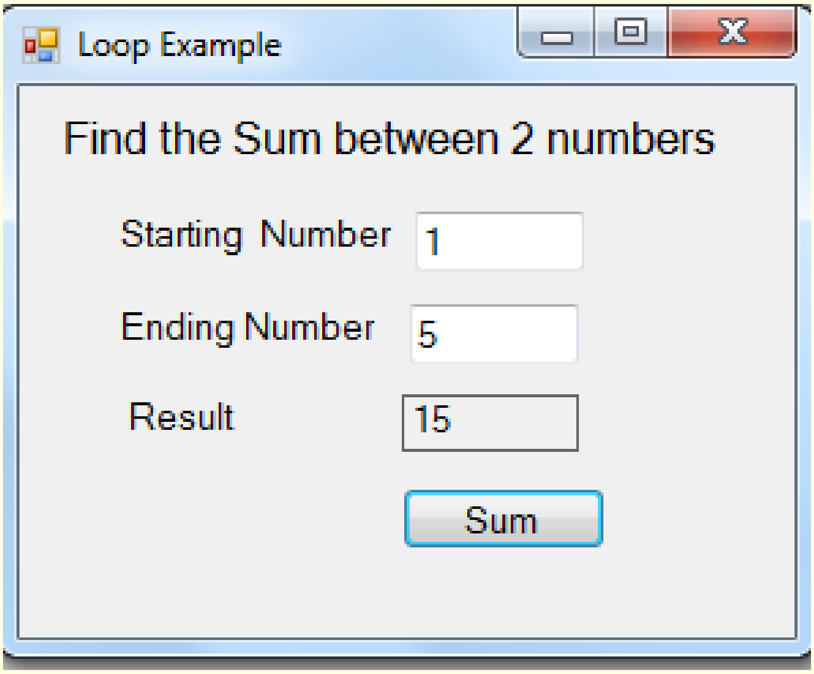
```
int sum = 0;
for ( int i = 1; i <= 5; i = i +1 )
{
    sum = sum +i;
}
lblDisplay.Text = sum.ToString()
```

Initial conditions i = 1

I	i<=5?	sum
1	True	1
2	True	3
3	True	6
4	True	10
5	True	15
6	False	

Example 1: Find the sum between 2 numbers

- Let us generalize the problem. Instead of adding numbers 1 through 5, we allow the user to enter a starting and ending number. The computer then computes the sum between these 2 numbers


Loop Example

Find the Sum between 2 numbers

Starting Number: 1

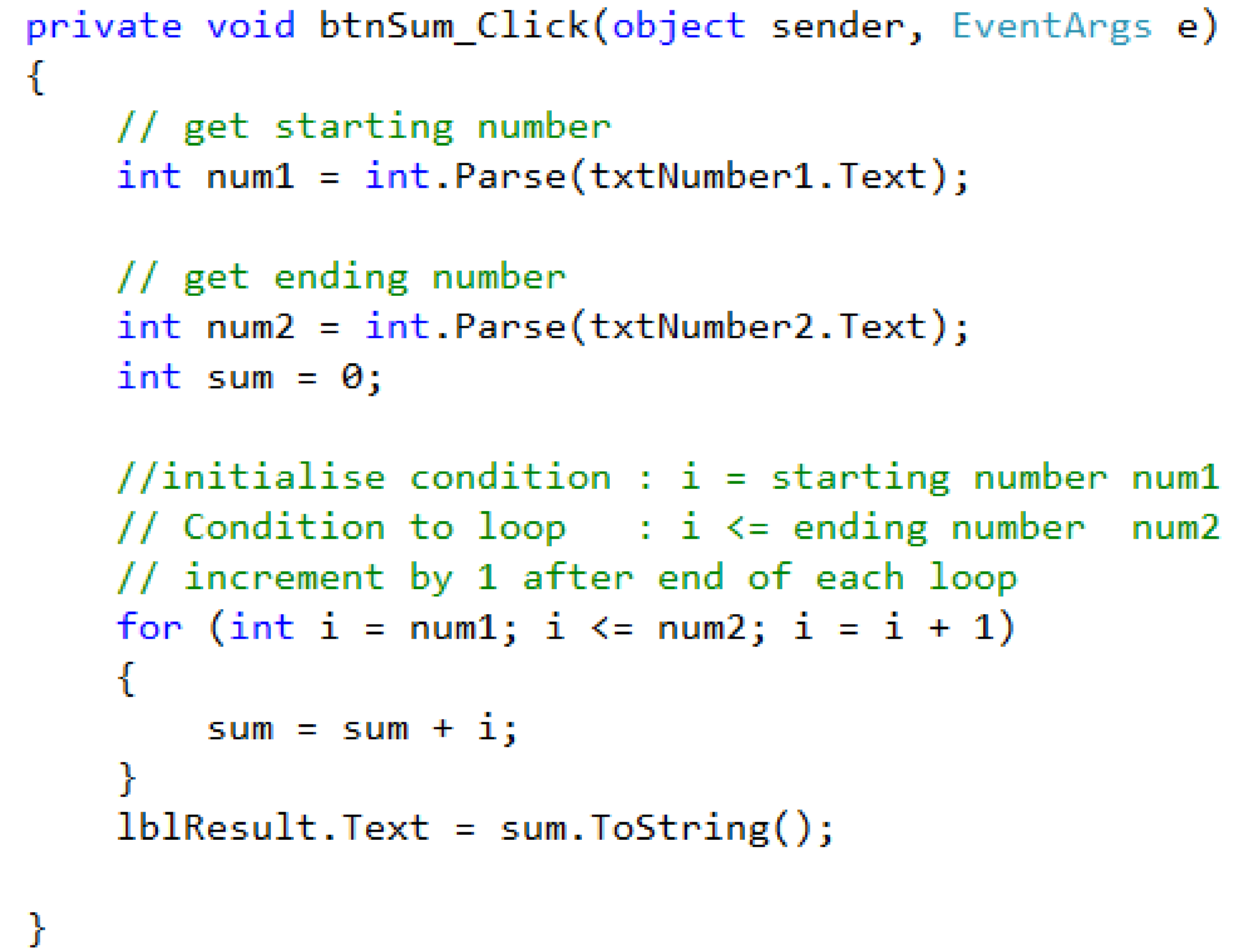
Ending Number: 5

Result: 15

Try writing the Algorithm or Pseudo code.

Example 1: Find the sum between 2 numbers

- Codes:



```
private void btnSum_Click(object sender, EventArgs e)
{
    // get starting number
    int num1 = int.Parse(txtNumber1.Text);

    // get ending number
    int num2 = int.Parse(txtNumber2.Text);
    int sum = 0;

    //initialise condition : i = starting number num1
    // Condition to loop   : i <= ending number  num2
    // increment by 1 after end of each loop
    for (int i = num1; i <= num2; i = i + 1)
    {
        sum = sum + i;
    }
    lblResult.Text = sum.ToString();

}
```

IT1753 Principles of Computing PCMP

Using while statement

While

- ❑ Besides the for statement, C# also has a while statement to implement looping.
- ❑ The syntax is as follows:

```
while (condition is true)
      statement/s
      update
end_while
```

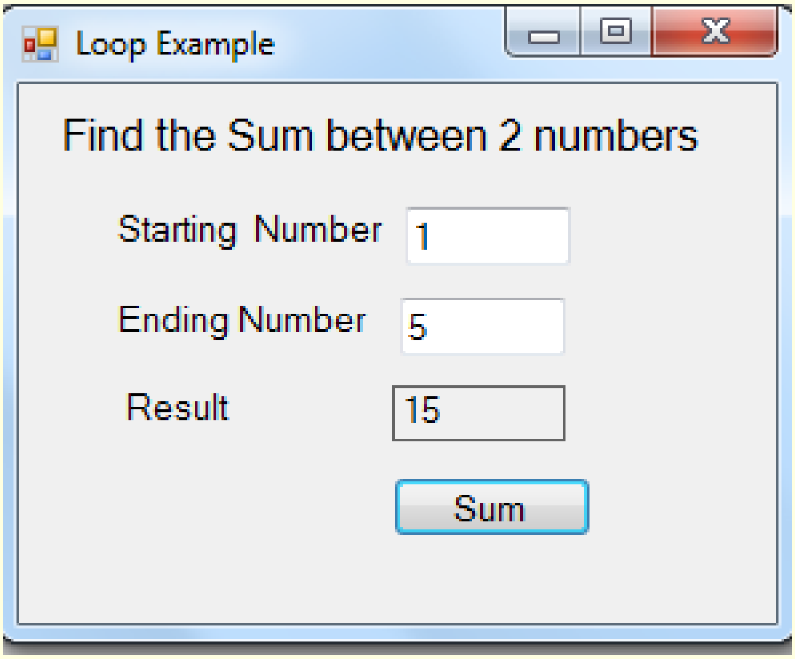
{ Evaluate condition first

} Statements will be executed as long as condition is TRUE

✋ If result on 1st evaluation is FALSE, statements will NOT be executed

Using while statement

- Let us see how we can write a program to solve Example 1:



Loop Example

Find the Sum between 2 numbers

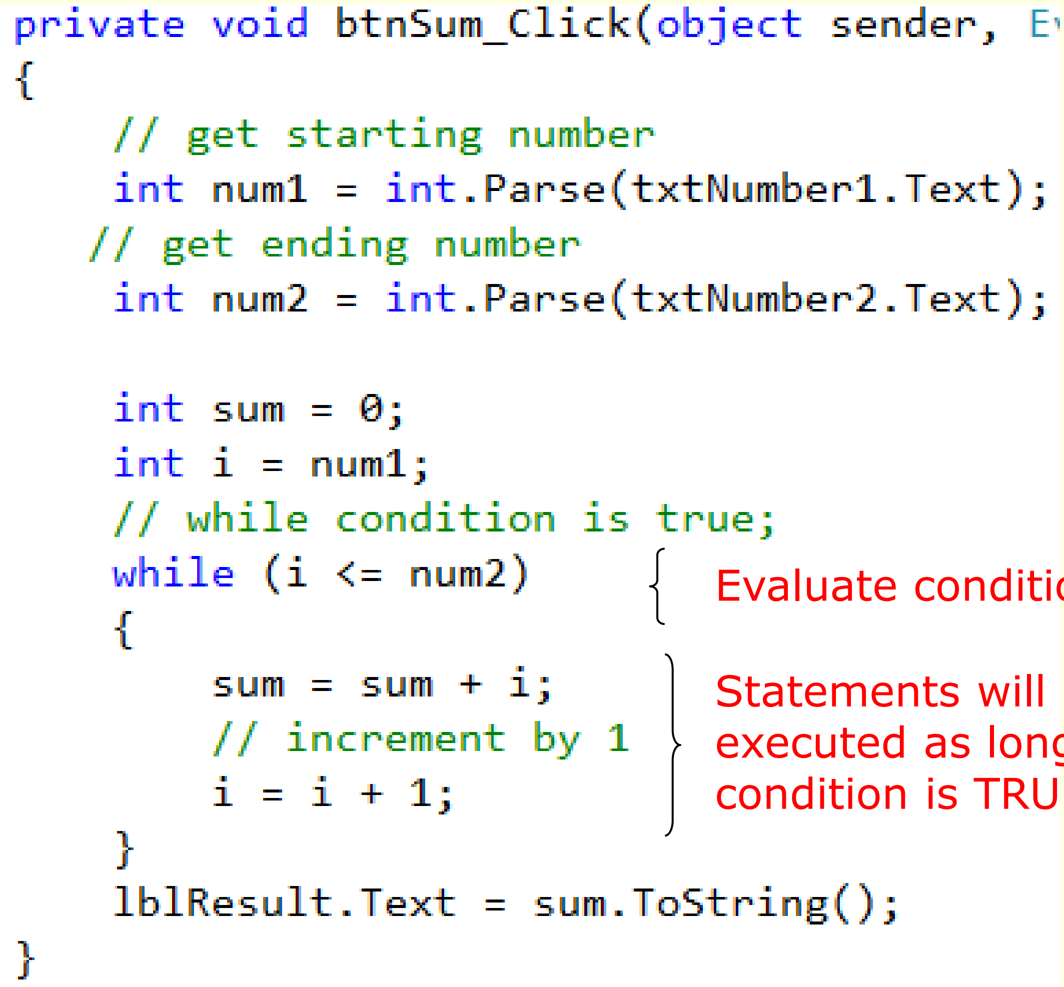
Starting Number 1

Ending Number 5

Result 15

Using while statement

- Code:



```
private void btnSum_Click(object sender, E
{
    // get starting number
    int num1 = int.Parse(txtNumber1.Text);
   // get ending number
    int num2 = int.Parse(txtNumber2.Text);

    int sum = 0;
    int i = num1;
    // while condition is true;
    while (i <= num2)
    {
        sum = sum + i;
        // increment by 1
        i = i + 1;
    }
    lblResult.Text = sum.ToString();
}
```

Evaluate condition first

Statements will be executed as long as condition is TRUE

IT1753 Principles of Computing PCMP

Iteration or Loop

- Another operation performed by a computer is:
  - **A Computer can Repeat a Group of Actions**

- The pseudo code for while loop:

```
while (condition is true)
      statement/s
      update
end_while
```

Evaluate condition first

Statements will be executed as long as condition is TRUE

✋ If result on 1st evaluation is FALSE, statements will NOT be executed

IT1753 Principles of Computing PCMP

FOR vs WHILE

- Performing repetitive tasks can be accomplished using for and while statements

F

O

R

Vs While

Should I use **FOR** or **WHILE** to repeat tasks? Which is better?

Applying Looping

- Apply **looping** :
  - Storing data using Text File
  - Use **loop** to read data till end of file
    - Which loop will you use **WHILE or FOR loop**?

  - Not possible to implement it without using **loop**

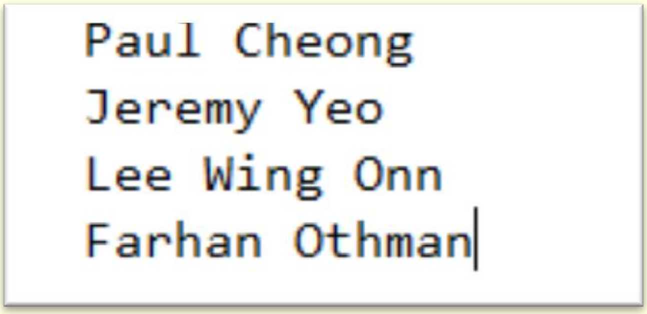
Storing data in file

- Variables stores data only temporary as long as a program is running. The data is lost when a program terminates.
- Files can be used for long term storage of records sequentially
- For large amount of data where relationship exists between them, using databases is more suitable.

IT1753 Principles of Computing PCMP

Working with Text File
Basics of File Input/Output

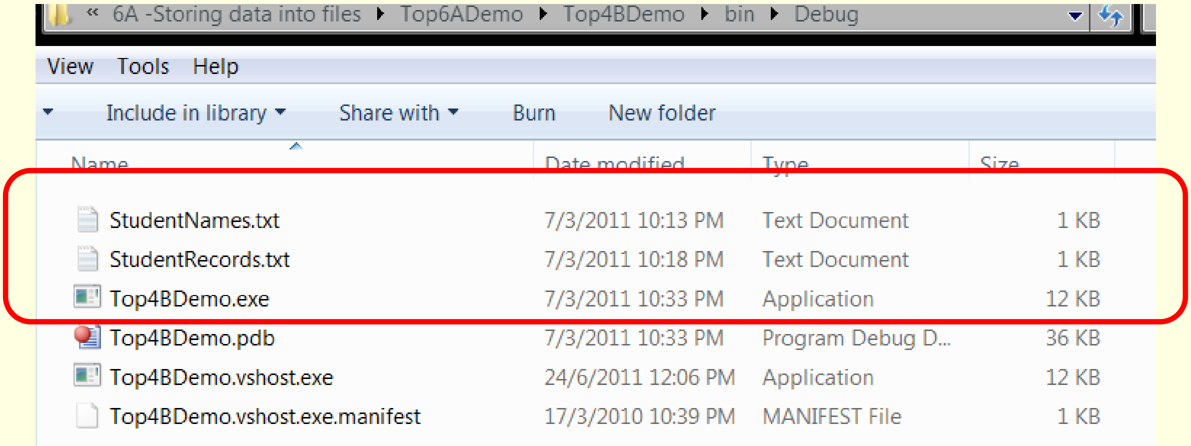
- ❑ A data file, for our purpose, is a text file.
- ❑ Each line in a file is a record
- ❑ A record consists of 1 or more field



```
Paul Cheong
Jeremy Yeo
Lee Wing Onn
Farhan Othman
```

A text file StudentNames.txt with 4 records

StudentNames.txt file is located at the folder where the executable application (XXX.exe) file is stored.



« 6A -Storing data into files ▸ Top6ADemo ▸ Top4BDemo ▸ bin ▸ Debug

Name	Date modified	Type	Size
StudentNames.txt	7/3/2011 10:13 PM	Text Document	1 KB
StudentRecords.txt	7/3/2011 10:18 PM	Text Document	1 KB
Top4BDemo.exe	7/3/2011 10:33 PM	Application	12 KB
Top4BDemo.pdb	7/3/2011 10:33 PM	Program Debug D...	36 KB
Top4BDemo.vshost.exe	24/6/2011 12:06 PM	Application	12 KB
Top4BDemo.vshost.exe.manifest	17/3/2010 10:39 PM	MANIFEST File	1 KB

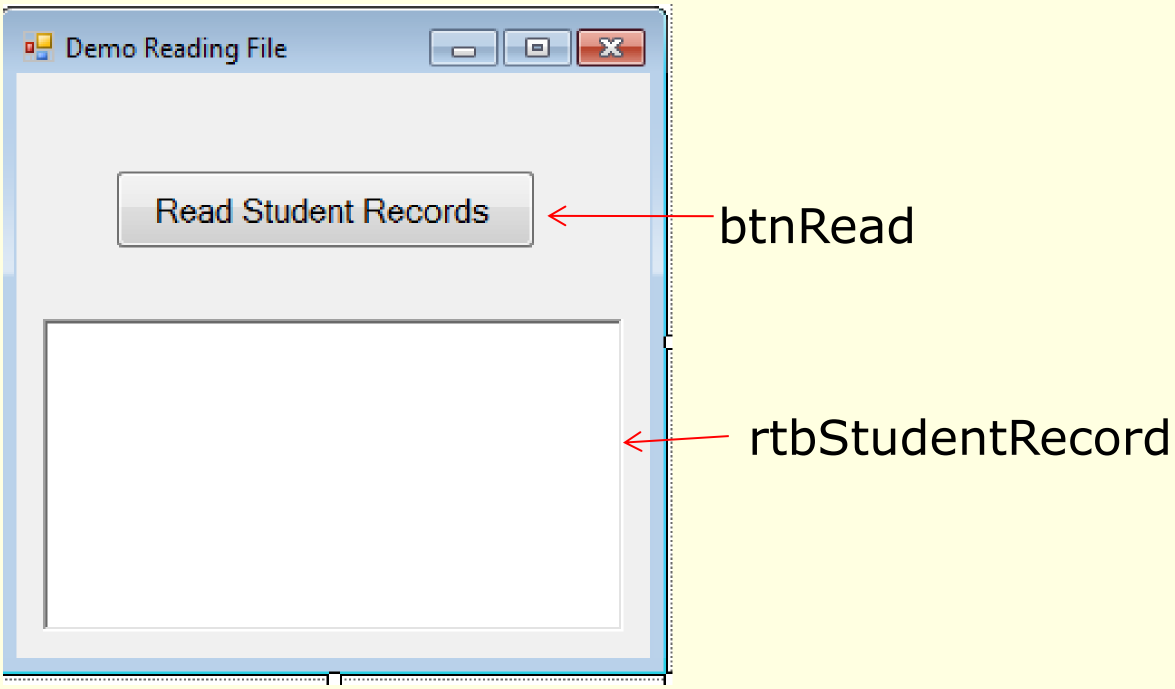
Working with Text File

Basics of File Input/Output

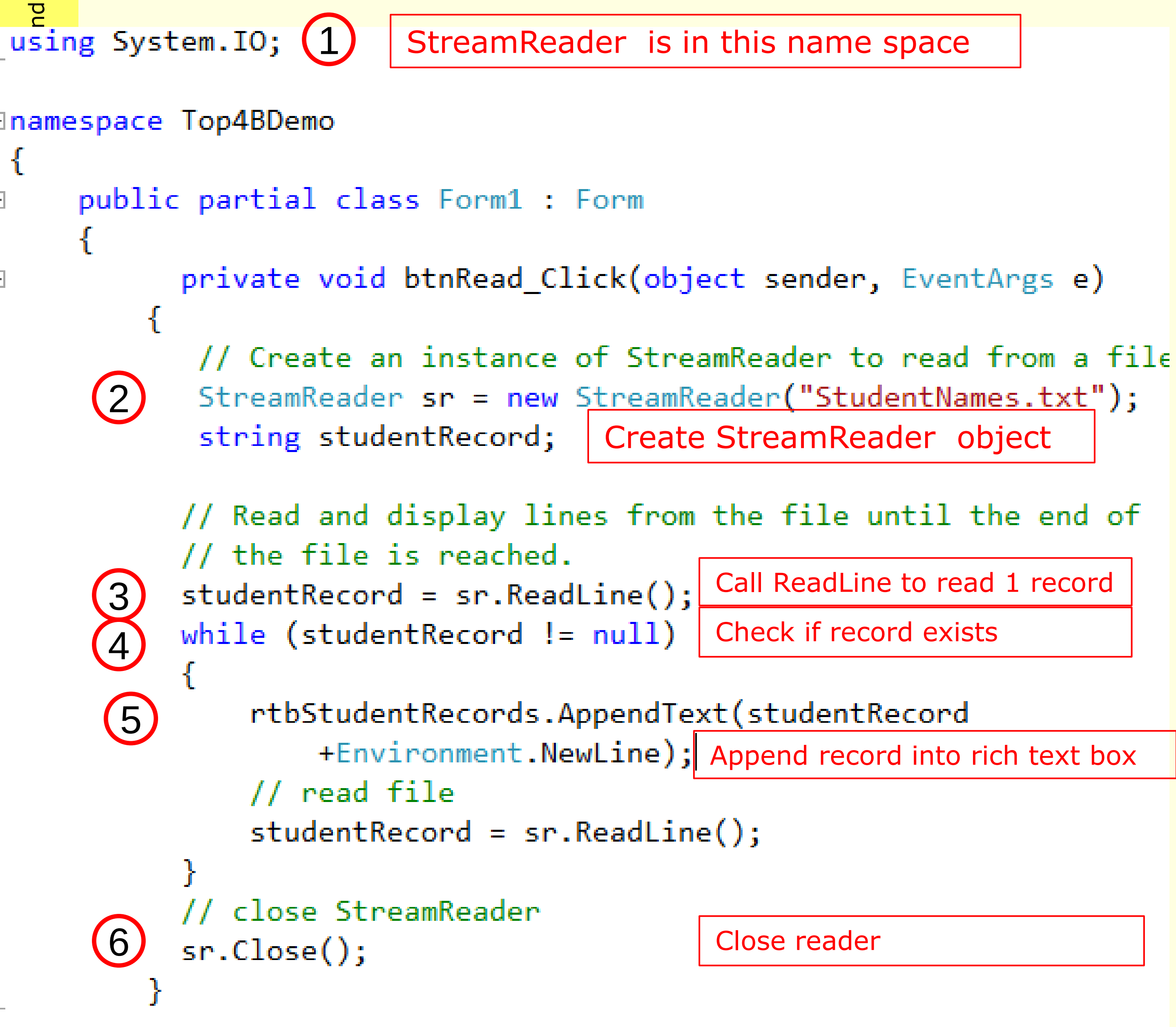
- Records are read sequentially, that is, read from the file in order from start to finish.
- This input is called a **data stream**
- In C#, we make use of the **StreamReader** class to read files

Example 2
Reading from a text data file

- In this example, when the button is clicked, we want to read a file studentRecords.txt and display them in the Rich Text Box



Example 2
Reading from a text data file



```
using System.IO;

namespace Top4BDemo
{
    public partial class Form1 : Form
    {
        private void btnRead_Click(object sender, EventArgs e)
        {
          // Create an instance of StreamReader to read from a file
          StreamReader sr = new StreamReader("StudentNames.txt");
          string studentRecord;

         // Read and display lines from the file until the end of
         // the file is reached.
         studentRecord = sr.ReadLine();
         while (studentRecord != null)
         {
             rtbStudentRecords.AppendText(studentRecord
                 +Environment.NewLine);
             // read file
             studentRecord = sr.ReadLine();
         }
         // close StreamReader
         sr.Close();
        }
```

1 StreamReader is in this name space

2 Create StreamReader object

3 Call ReadLine to read 1 record

4 Check if record exists

5 Append record into rich text box

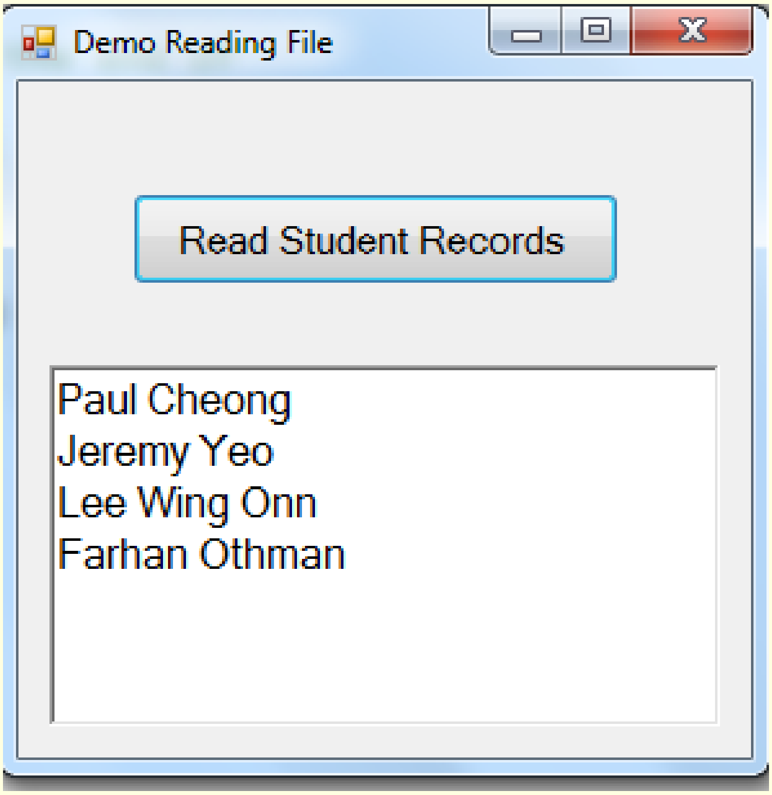
6 Close reader

Can you spot the **looping** statement?

IT1753 Principles of Computing PCMP

Example 2
Reading from a text data file

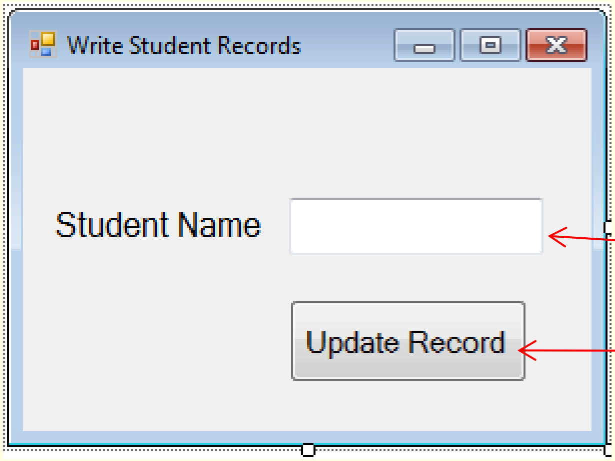
- Sample Output:



```
Paul Cheong
Jeremy Yeo
Lee Wing Onn
Farhan Othman
```

Example 3
Writing to a text data file

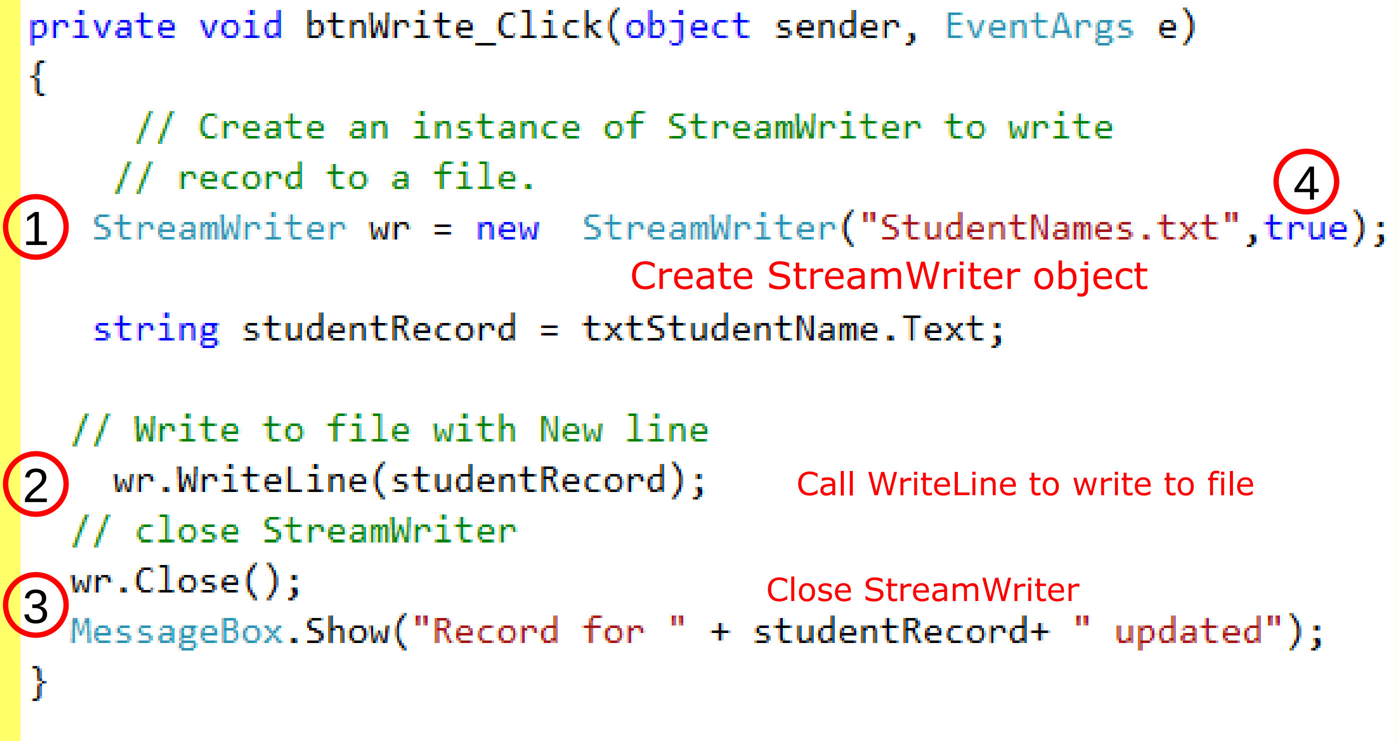
- In this example, when the button is clicked, we want to write a record to studentRecords2.txt



txtStudentRecord

Write

Example 3
Writing to a text data file



```
private void btnWrite_Click(object sender, EventArgs e)
{
     // Create an instance of StreamWriter to write
    // record to a file.
   StreamWriter wr = new  StreamWriter("StudentNames.txt",true);

   string studentRecord = txtStudentName.Text;

  // Write to file with New line
    wr.WriteLine(studentRecord);
  // close StreamWriter
  wr.Close();
  MessageBox.Show("Record for " + studentRecord+ " updated");
}
```

Create StreamWriter object

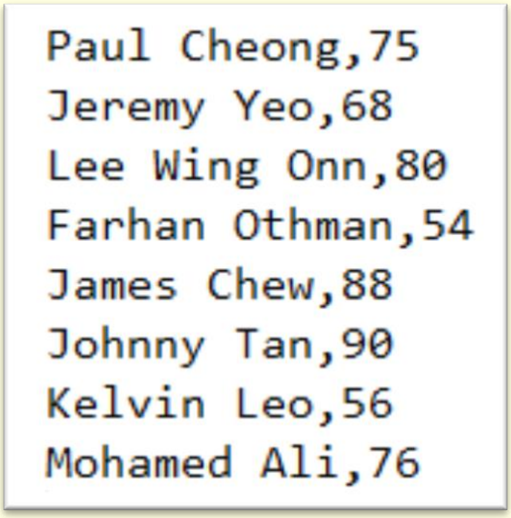
Call WriteLine to write to file

Close StreamWriter

Status	File Exists?	Result
true	Yes	Records will be appended to existing record
false	Yes	Existing records will be overwritten
-	No	New file created

Records with multiple fields

- ❑ So far, we have seen only files with single field records
- ❑ In the real whole, each record will have multiple fields.
- ❑ Example:

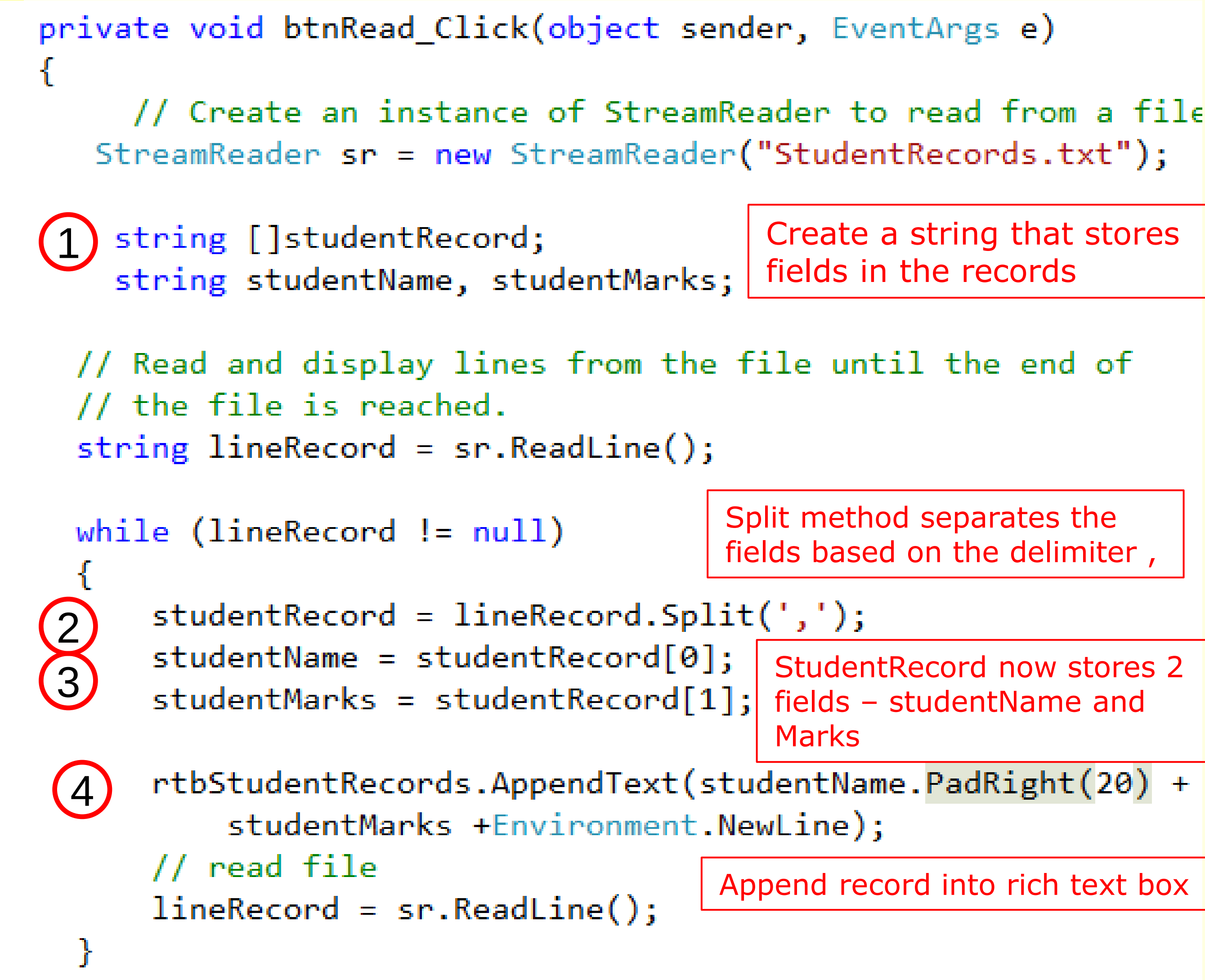


```
Paul Cheong,75
Jeremy Yeo,68
Lee Wing Onn,80
Farhan Othman,54
James Chew,88
Johnny Tan,90
Kelvin Leo,56
Mohamed Ali,76
```

- ❑ Let us modify Example 1 Reading from a text file to read records with multiple fields.

Example 4
Reading from a data file with multiple field records

- ❑ We will focus here only the modifications required:



```
private void btnRead_Click(object sender, EventArgs e)
{
     // Create an instance of StreamReader to read from a file
   StreamReader sr = new StreamReader("StudentRecords.txt");

    string []studentRecord;
    string studentName, studentMarks;

  // Read and display lines from the file until the end of
  // the file is reached.
  string lineRecord = sr.ReadLine();

  while (lineRecord != null)
  {
      studentRecord = lineRecord.Split(',');
      studentName = studentRecord[0];
      studentMarks = studentRecord[1];

      rtbStudentRecords.AppendText(studentName.PadRight(20) +
          studentMarks +Environment.NewLine);
      // read file
      lineRecord = sr.ReadLine();
  }
```

1

Create a string that stores fields in the records

Split method separates the fields based on the delimiter ,

2

3

StudentRecord now stores 2 fields – studentName and Marks

4

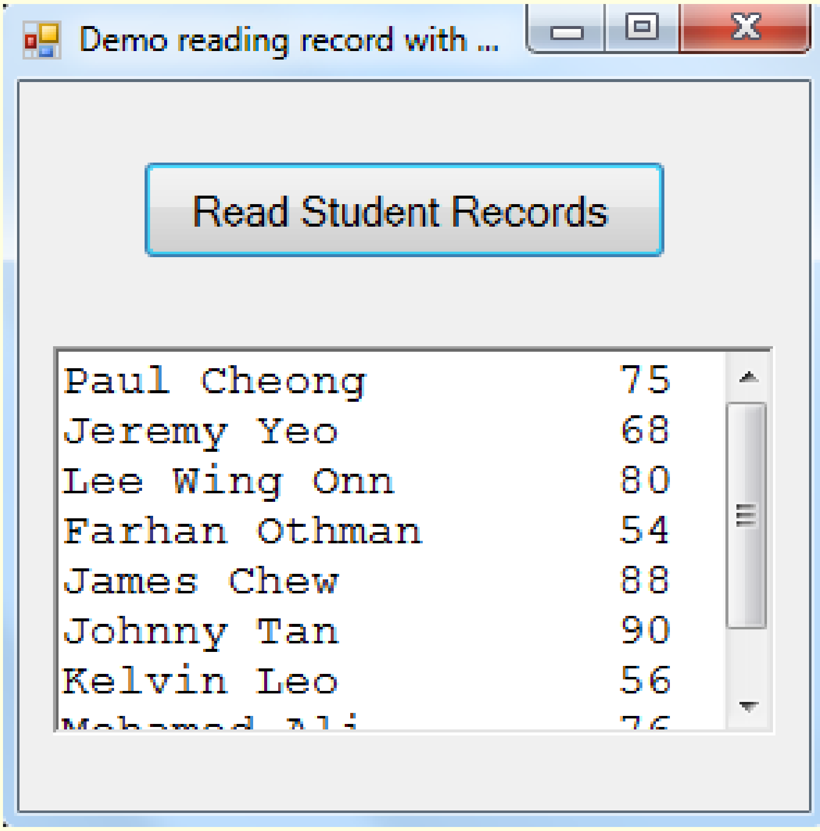
Append record into rich text box

Can you spot the **looping** statement?

IT1753 Principles of Computing PCMP

Example 4
Reading from a data file with multiple field records

- Sample Output



```
Paul Cheong          75
Jeremy Yeo           68
Lee Wing Onn         80
Farhan Othman        54
James Chew           88
Johnny Tan           90
Kelvin Leo           56
Mohamed Ali          76
```

Summary

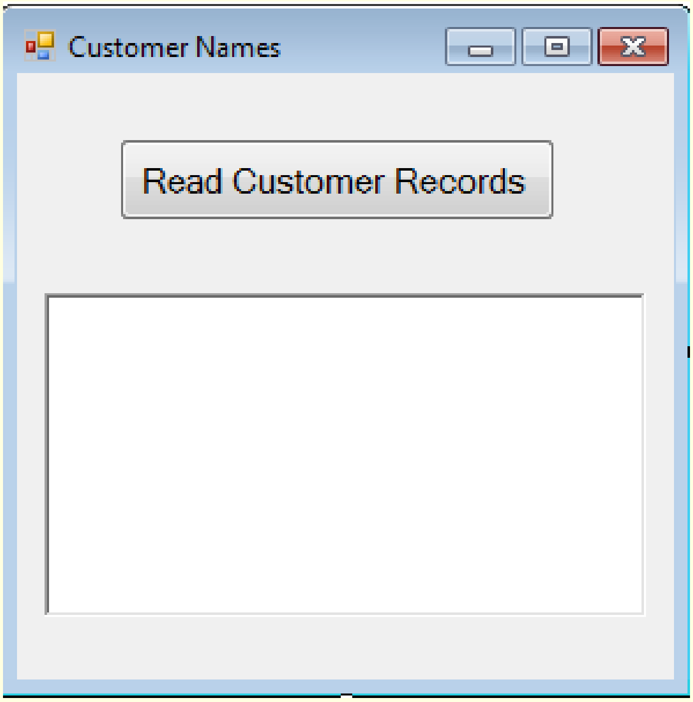
Writing to Text File

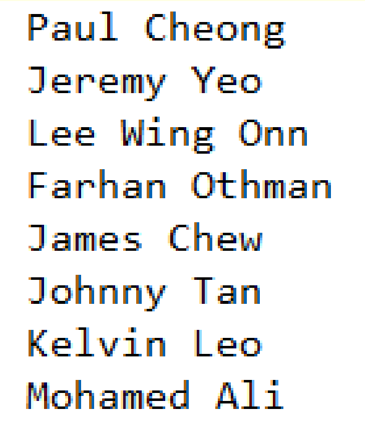
- StreamReader Class is used for reading data from Text Files
- StreamWriter Class is used for writing data to Text Files
- The Split method of a string can be used to separate the fields in the string based on the delimiter passed to the Split method.

Practical 5A

1. Read and display Customer names from data file

- ❑ **Task**: ABC Marketing Ltd has kept its customer names in a text file. You are to develop a program which will read from the file and display it in the form. Use Form frmReadCustomerNames given.
  Note : The data file CustomerNames.txt is given. The sample data is shown below.
- ❑ **Form Design** :



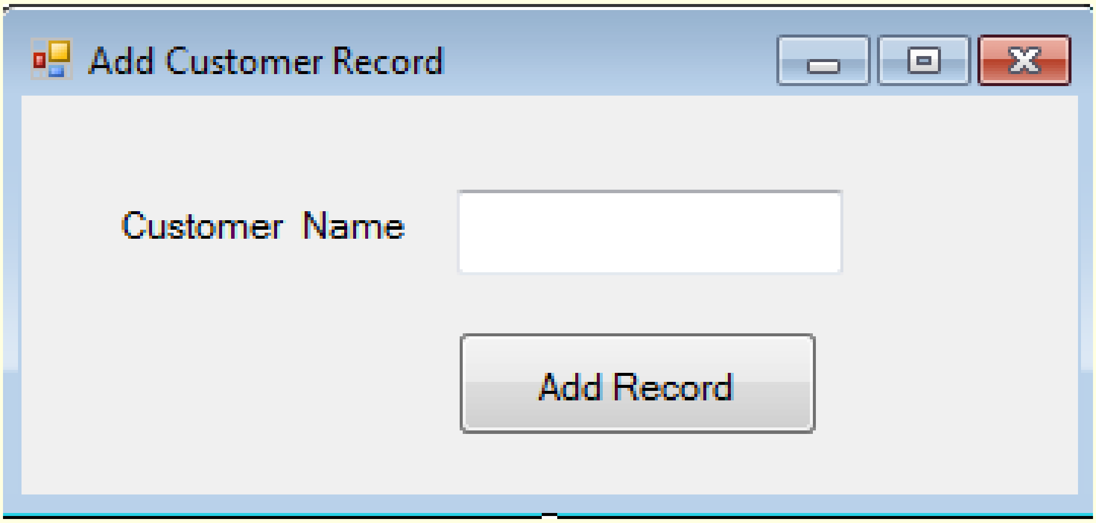


```
Paul Cheong
Jeremy Yeo
Lee Wing Onn
Farhan Othman
James Chew
Johnny Tan
Kelvin Leo
Mohamed Ali
```

Practical 5A

2. Write customer records to data file

- ❑ **Task**: ABC Marketing Ltd also need to add its customer records to a text file. You are to develop a program which will write to the file and Use Form frmAddCustomerRecord given.
  Note : The data file CustomerRecords.txt is given
- ❑ **Form Design** :



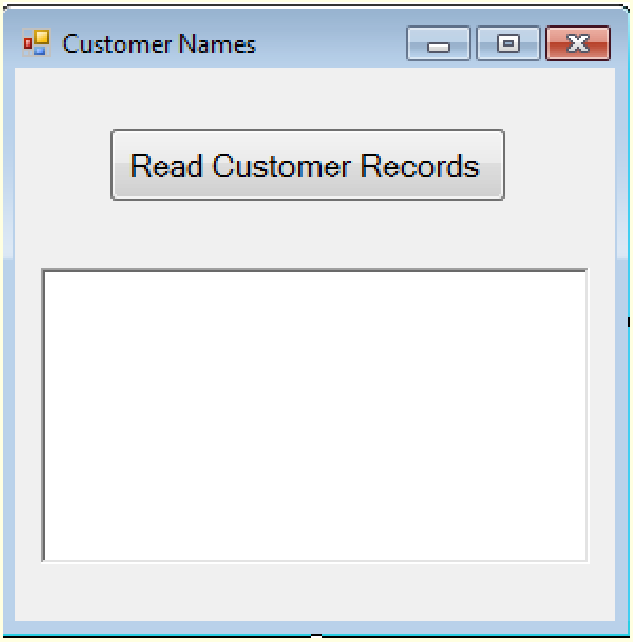
IT1753 Principles of Computing PCMP

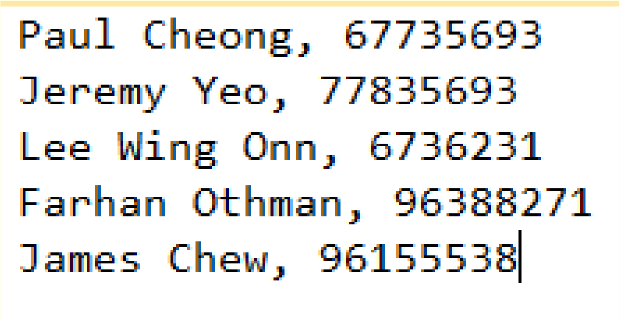
Practical 5A

3. Read and display Customer records with more than 1 field from data file

- ❑ **Problem Statement**: ABC Marketing Ltd has kept its customer records (each record has a name and contact number) in a text file. You are to develop a program which will read from the file and display it in the form. Use Form frmRecordMultiField.cs is given. Note : The data file MultiFieldCustomerNames.txt is given. The sample data is shown below.

- ❑ **Form Design** :





```
Paul Cheong, 67735693
Jeremy Yeo, 77835693
Lee Wing Onn, 6736231
Farhan Othman, 96388271
James Chew, 96155538
```

Practical 5A

4. Create the form in Figure 4A below. When the Run button is clicked, use a **FOR** loop to display “Hello world” **10** times in the Rich Text Box (see Figure 4B).

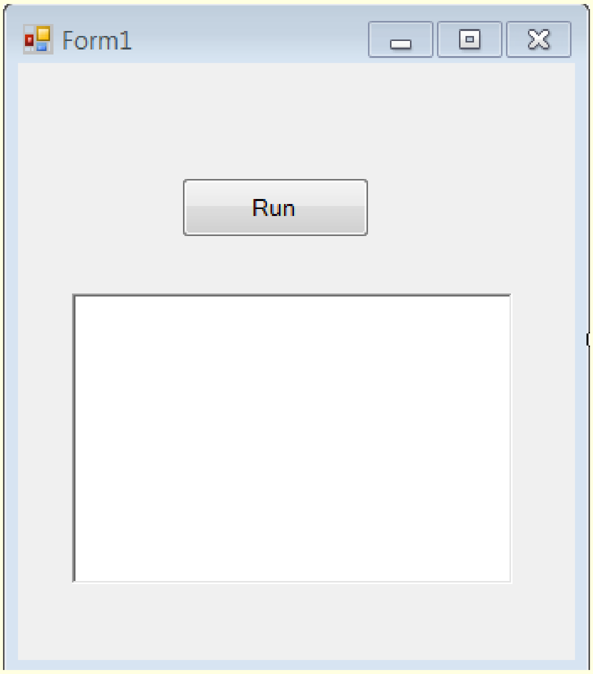
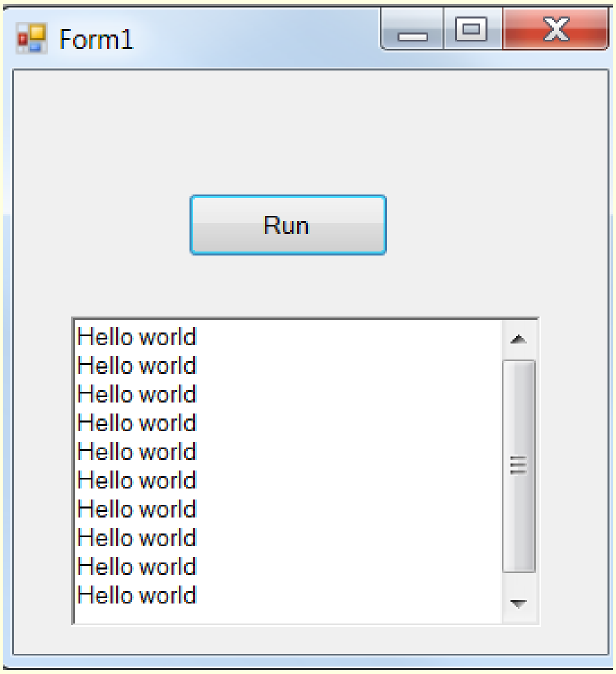


Fig 4A



```
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
```

Fig 4B

Practical 5A

5. Modify question 4 to read in a text string and an integer. When the Run button is clicked, use a **FOR** loop to display the text entered in the Text Box for X number of times, where X is the integer entered by the user.

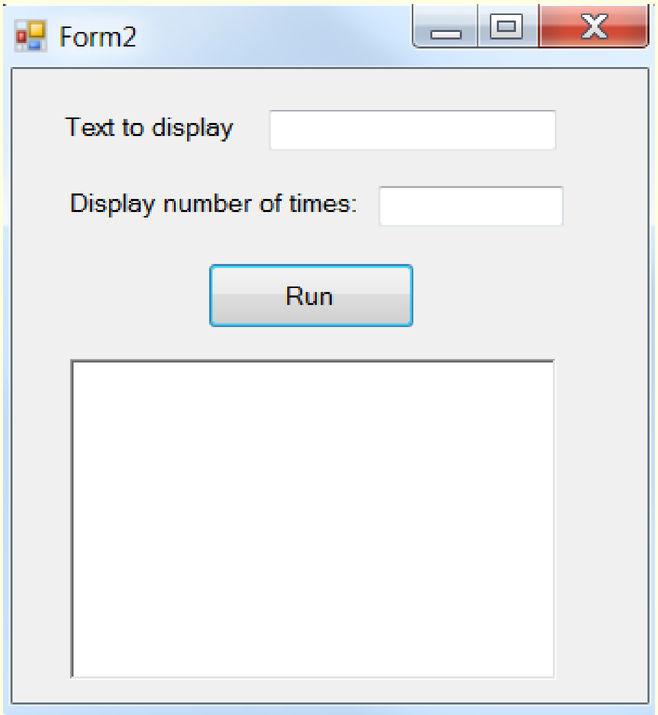
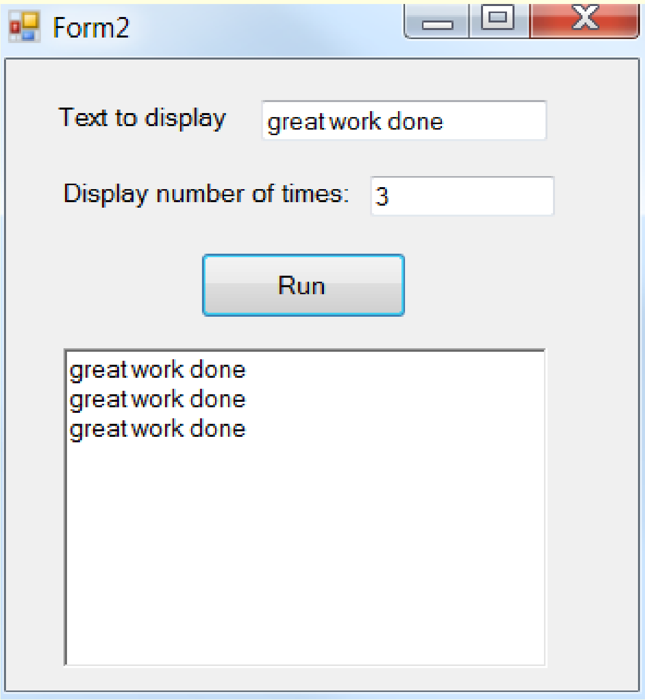


Fig 5A



Text to display: great work done

Display number of times: 3

```
great work done
great work done
great work done
```

Fig 5B

Practical 5A

6. Project Work

- ❑ Apply the lesson learnt into your project.
- ❑ When program ends, store the data entered by the user to a text file.
- ❑ Retrieve the data from the text file when program starts.

End of Topic 5A

Looping for Repetitive Tasks
Storing Data using Text File