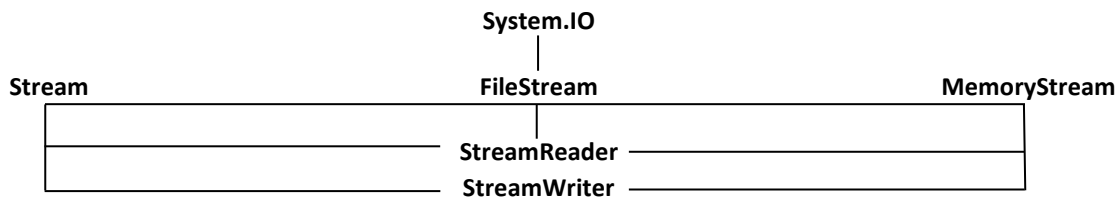


## File Stream



The System.IO namespace provides the following classes in order to work with file streams.

### 1. FileStream

This class is used to work with files and file related options.

#### Syntax:

```
FileStream <obj-name>=new FileStream(Path,FileMode,FileAccess)
```

Where

- Path is the complete path of the file name.
- **FileMode:** Specifies the mode of working with the file.
  - FileMode.Create:  
Create a new file
  - FileMode.CreateNew:  
Creates a new file with a warning if the filename same exist.
  - FileMode.Open:  
Opens the file.
- **FileAccess:** File access specifies the access accessibility of the file with regards to operation.
  - FileAccess.Read:  
Reads the file
  - FileAccess.Write:  
Write the data into the file.
  - FileAccess.ReadWrite:  
The file can be accessed read as well as write operation.

### 2. Stream

This class is used to commonly access any stream of characters from input devices like keyboard.

### 3. MemoryStream

This class can be used directly access stream from main memory (RAM).

All the above classes have the following subclasses

#### Class StreamWriter

This class is used to write a stream of data into a FileStream.

```
StreamWriter <obj-name>=new StreamWriter (FileStream)
```

Close()	Closes the stream writers.
Flush()	Clears the entire buffer stream.
Write(string)	Writers characters into the file stream.
WriteLine(string)	Writes a line of characters in to the file stream.

#### Class StreamReader

This class is used to read a stream of characters in the file stream.

```
StreamReader <obj-name>=new StreamReader(FileStream)
```

Close()	Closes the stream reader.
Peek()	Gets the next available character in the file stream.
Read()	Reads characters from file stream.
ReadToEnd()	Reads from the current cursor position at the end of the position.
ReadLine()	Read the lines of characters from the file stream.

Create an interface that can store employee information into a file using the file stream objects.

Step 1: create the following interface

The image shows a Windows Forms application window titled "IO". The window contains a form with the following elements:

- Employee ID :
- Name :
- Job :
- Salary :
- A list box labeled "listBox1" on the right side.
- Four buttons at the bottom: "Create", "Save", "Read", and "Close".

Step 2: Add the action for the form

```

string Filename = @"E:\Nagendra Data\C# Course Content\Class #9 C#-IO\IO Programs\Files\Sample.txt";
private bool nonNumberEntered = false;
FileStream fs;

//Create
try
{
    fs = new FileStream(Filename, FileMode.Create);
    MessageBox.Show("File is created as " + Filename, "Create", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
catch (FileNotFoundException fnfe)
{
    MessageBox.Show(fnfe.Message);
}
catch (NullReferenceException ne)
{
    MessageBox.Show(ne.Message);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    button1.Enabled = false;
    fs.Close();
    MessageBox.Show("File is created and closed");
}

//Save
try
{
    fs = new FileStream(Filename, FileMode.Append);
    string s = "";
    StreamWriter sw = new StreamWriter(fs);
    s = textBox1.Text + "," + textBox2.Text + "," + comboBox1.SelectedItem.ToString() + ","
+ textBox3.Text;
    sw.WriteLine(s);
    sw.Flush();
    sw.Close();
    MessageBox.Show("One Employee record created", "Create", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    fs.Close();
}

//Read
try
{
    fs = new FileStream(Filename, FileMode.Open);
    listBox1.Visible = true;
    StreamReader sr = new StreamReader(fs);
    while (sr.Peek() > -1)
    {
        listBox1.Items.Add(sr.ReadLine());
    }
}

```

```
        }
        sr.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        fs.Close();
    }
//Close
    this.Close();
}
```