Exception Handling Lab

Lab 1: No Exception Handling

Replace the entire contents of the **Program.cs** file with the following code.

```
long size;
FileStream fs;

fs = File.Open(@"D:\Samples\Test.txt", FileMode.Open);
size = fs.Length;
fs.Close();
```

Try It Out

Run the application and view the output.

Lab 2: Add Try…Catch

Replace the entire contents of the **Program.cs** file with the following code.

```
long size;
FileStream fs;

try {
  fs = File.Open(@"D:\Samples\Test.txt", FileMode.Open);
  size = fs.Length;
  fs.Close();
}
catch {
  Console.WriteLine("Error Occurred");
}
```

Try It Out

Run the application and view the output.

Lab 3: Add Exception Object

Replace the entire contents of the **Program.cs** file with the following code.

```
long size;
FileStream fs;

try {
  fs = File.Open(@"D:\Samples\Test.txt", FileMode.Open);
  size = fs.Length;
  fs.Close();
}
catch (Exception ex) {
  Console.WriteLine($"Message: {ex.Message}");
  Console.WriteLine($"InnerException: {ex.InnerException}");
  Console.WriteLine($"Source: {ex.Source}");
  Console.WriteLine($"TargetSite: {ex.TargetSite}");
  Console.WriteLine();

Console.WriteLine("********************************************");
  Console.WriteLine(ex.ToString());
}
```

Try It Out

Run the application and view the output.

Lab 4: Add Finally Block

Replace the entire contents of the **Program.cs** file with the following code.

Declare the FileStream object as a nullable type. If this does not work, check the global **Nullable** setting.

```
long size;
FileStream? fs = null;

try {
  fs = File.Open(@"D:\Samples\Test.txt", FileMode.Open);
  size = fs.Length;
}
catch (Exception ex) {
  Console.WriteLine(ex.ToString());
}
finally {
  fs?.Close();
}
```

Try It Out

Run the application and view the output.

Lab 5: Specific Exceptions

Replace the entire contents of the **Program.cs** file with the following code.

```
long size;
FileStream fs = null;

try {
  fs = File.Open(@"D:\Samples\Test.txt", FileMode.Open);
  size = fs.Length;
}
// This one must go before 'ArgumentException'
catch (ArgumentOutOfRangeException ex) {
  Console.WriteLine(ex.Message);
}
catch (ArgumentException ex) {
  Console.WriteLine(ex.Message);
}
catch (PathTooLongException ex) {
  Console.WriteLine(ex.Message);
}
// This one must go before 'IOException'
catch (DirectoryNotFoundException ex) {
  Console.WriteLine(ex.Message);
}
// This one must go before 'IOException'
catch (FileNotFoundException ex) {
  Console.WriteLine(ex.Message);
}
catch (IOException ex) {
  Console.WriteLine(ex.Message);
}
catch (UnauthorizedAccessException ex) {
  Console.WriteLine(ex.Message);
}
catch (NotSupportedException ex) {
  Console.WriteLine(ex.Message);
}
catch (Exception ex) {
  Console.WriteLine(ex.Message);
}
finally {
  fs?.Close();
}
```

Try It Out

Run the application and view the output.

Test ArgumentException

Change the line after the **try** statement to the following.

```
fs = File.Open("", FileMode.Open);
```

Try It Out

Run the application and view the output.

Test DirectoryNotFoundException

Change the line after the **try** statement to the following.

```
fs = File.Open(@"D:\BadDirectory\Test.txt", FileMode.Open);
```

Try It Out

Run the application and view the output.

Test FileNotFoundException

Change the line after the **try** statement to the following.

```
fs = File.Open(@"D:\Samples\Test.txt", FileMode.Open);
```

Try It Out

Run the application and view the output.