\*\* Exercise: An example of a try block

Step 2:

using static System.Console;

Step 3:

WriteLine("Before parsing");

Write("Which page number you are in? ");

string input = Console.ReadLine();

try

{

int pg = int.Parse(input);

WriteLine($"You are in page number {pg}.");

}

catch

{

}

WriteLine("After parsing");

\*\* Exercise: Catch exceptions using the catch block

Step 1:

catch(Exception ex)

{

WriteLine($"{ex.GetType()} says {ex.Message}");

}

\*\* Exercise: Catch specific exceptions

Step 1:

catch (FormatException)

{

WriteLine("The page number you entered is not a valid number format.");

}

catch (Exception ex)

{

WriteLine($"{ex.GetType()} says {ex.Message}");

}

Step 3:

catch(OverflowException)

{

WriteLine("Your page number is a valid number format but it is either too big or small.");

}

catch (FormatException)

{

WriteLine("The page number you entered is not a valid number format.");

}

\*\* Exercise: Use finally statement

string path = @"C:\Code\Lesson3";

FileStream file = null;

StreamWriter writer = null;

try

{

if (Directory.Exists(path))

{

file = File.OpenWrite(Path.Combine(path, "file.txt"));

writer = new StreamWriter(file);

writer.WriteLine("Hello, C#!");

}

else

{

WriteLine($"{path} does not exist!");

}

}

catch (Exception ex)

{

// if the path doesn't exist the exception will be caught

WriteLine($"{ex.GetType()} says {ex.Message}");

}

finally

{

if (writer != null)

{

writer.Dispose();

WriteLine("The writer's unmanaged resources have beendisposed.");

}

if (file != null)

{

file.Dispose();

WriteLine("The file's unmanaged resources have beendisposed.");

}

}

\*\* Exercise: Dispose using the using statement

using (FileStream file2 = File.OpenWrite(

Path.Combine(path, "file2.txt")))

{

using (StreamWriter writer2 = new StreamWriter(file2))

{

try

{

writer2.WriteLine("Welcome, .NET Core!");

}

catch (Exception ex)

{

WriteLine($"{ex.GetType()} says {ex.Message}");

}

} // automatically calls Dispose if the object is not null

} // automatically calls Dispose if the object is not null

\*\* Activity D-1

static void MultipleCatch(params string[] args)

{

try

{

byte b = byte.Parse(args[0]);

Console.WriteLine(b);

}

catch (IndexOutofRangeException ex)

{

Console.WriteLine("Atleast one argument is required (IndexOutOfRangeException)");

}

catch (FormatException ex)

{

Console.WriteLine("Not a number! (FormatException)");

}

catch (OverflowException ex)

{

Console.WriteLine("More than a byte (OverflowException)...");

}