Day 5

https://www.scholarhat.com/tutorial/csharp/a-deep-dive-into-csharp-errors-or-exceptions-handling

Exception handling in C# is a technique to handle errors coming when you execute any program. It supports the developers in recognizing and managing unforeseen circumstances, keeping the software from crashing. C# assists in making sure that mistakes are handled gracefully and resources are managed appropriately by employing **try**, **catch**, and, **finally**, **blocks**.

```
// C# program to show how Exceptions occur in a program
using System;

class Program
{
  static void Main()
  {
    try
    {
      // Code that may throw an exception
      int numerator = 10;
      int denominator = 0; // This will cause a DivideByZeroException
      int result = numerator / denominator;
      Console.WriteLine("Result: " + result);
    }
    catch (DivideByZeroException ex)
    {
      // Handling the exception
      Console.WriteLine("Error: Cannot divide by zero.");
```

```
        Console.WriteLine("Exception Message: " + ex.Message);
      }
      // Additional code continues here if needed
      Console.WriteLine("Program continues even after the exception.");
   }
}
```

Exception Handling hands-On:

```
public class ExceptionHandling
{
   public static void Process()
   {
      int a;
      while (true) // Loop until a valid integer is entered
      {
         try
         {
            Console.Write("Enter Value: ");
            a = Convert.ToInt32(Console.ReadLine());
            break; // Exit loop if input is valid
         }
         catch (FormatException)
         {
            Console.WriteLine("Invalid Input! Please enter a valid integer.");
         }
         catch (Exception ex) // General exception handling
         {
```

```
            Console.WriteLine($"An unexpected error occurred: {ex.Message}");
        }
        finally
        {
            Console.WriteLine("Program executed");
        }
      }
      Console.WriteLine($"Valid Input: {a}");
   }
}

class Program
{
   static void Main()
   {
      ExceptionHandling.Process();
   }
}
```