| **DAY 12 : Assignment**  **By**  **Vihar D.** |
| --- |

| **Assignment 1** |
| --- |
| **What is Exception Handling and why do we need exception handling ?** |
| **Answer :** |
| * **Exception Handling is done to ensure that the application will not crash.** * **It will not display any technical details and to make sure we handle errors gracefully and display friendly messages.** * **Exception is an event or object which is thrown at runtime.** * **Exceptions are derived from the system namespace.** |

| **Assignment 2** |
| --- |
| **Write a C# code for division program, add exceptions along with super exception at the end.** |
| **Answer :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace div\_3except\_super  {  internal class Program  {  static void Main(string[] args)  {  try  {  int a, b, c;  Console.WriteLine("\n enter dividend value : ");  a = Convert.ToInt32(Console.ReadLine());  Console.WriteLine($"\n enter divisor value to divide {a} : ");  b = Convert.ToInt32(Console.ReadLine());  c = a / b;  Console.WriteLine($"\n division of {a} / {b} is {c}");  Console.ReadLine();  }  //Overflow Exception-----------------------------------------  catch (OverflowException)  {  Console.WriteLine("\n Please enter the number in range of 0 - 50000");  }  //DivideByZero Exception-----------------------------------------  catch (DivideByZeroException)  {  Console.WriteLine("\n Please enter divisor value != 0");  }  //Format Exception-----------------------------------------  catch (FormatException)  {  Console.WriteLine("\n Please enter integers only");  }  catch (Exception)  {  Console.WriteLine("\n Error 404, Please contact ADMIN.");  }  finally  {  Console.WriteLine("\n\n\n\n\n\n\*\*\*\*\*\*\*\*\*\*\* Developed by Vihar D \*\*\*\*\*\*\*\*\*\*\*");  Console.ReadLine();  }  }  }  } |
| **Output :** |
|  |

| **Assignment 3** |
| --- |
| **Research and write at least 6 exceptions that occur in C# with code snippets.** |
| **Answer :** |
| **i : DivideByZeroException (Code)** |
| **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Threading.Tasks;**  **namespace random**  **{**  **internal class Program**  **{**  **//DivideByZeroException**  **static void Main(string[] args)**  **{**  **int a = 5;**  **int b = 0;**  **int c = a / b;**  **Console.WriteLine(c);**  **}**  **}**  **}** |
| **Output :** |
|  |

| **ii : FileNotFoundException (Code)** |
| --- |
| **using System;**  **using System.IO;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Threading.Tasks;**  **namespace random**  **{**  **internal class Program**  **{**  **//FileNotfoundException**  **static void Main(string[] args)**  **{**  **File.Open("d:\\random.txt", FileMode.Open);**  **}**  **}**  **}** |
| **Output :** |
|  |

| **iii : StackOverFlowException (Code)** |
| --- |
| **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Threading.Tasks;**  **namespace random**  **{**  **internal class Program**  **{**  **static void Postinc(int x)**  **{**  **Console.WriteLine(x);**  **Postinc(x++);**  **}**  **//StackOverFlowException**  **static void Main(string[] args)**  **{**  **Postinc(0);**  **}**  **}**  **}** |
| **Output :** |
|  |

| **iv : IndexOutOfRangeException (Code)** |
| --- |
| **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Threading.Tasks;**  **namespace random**  **{**  **internal class Program**  **{**  **//IndexOutOfRangeException**  **static void Main(string[] args)**  **{**  **int[] num = new int[10];**  **num[0] = 5;**  **num[10] = 10;**  **num[15] = 15;**  **}**  **}**  **}** |
| **Output :** |
|  |

| **v : ArrayTypeMismatchException (Code)** |
| --- |
| **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Threading.Tasks;**  **namespace random**  **{**  **internal class Program**  **{**  **//ArrayTypeMismatchException**  **static void Main(string[] args)**  **{**  **string[] array1 = { "Hello", "World" };**  **object[] array2 = array1;**  **array2[0] = 100;**  **}**  **}**  **}** |
| **Output :** |
|  |

| **vi : OutOfMemoryException (Code)** |
| --- |
| **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Threading.Tasks;**  **namespace random**  **{**  **internal class Program**  **{**  **//OutOfMemoryException**  **static void Main(string[] args)**  **{**  **string str1 = new string('x', int.MaxValue);**  **}**  **}**  **}** |
| **Output :** |
|  |

| **Assignment 4** |
| --- |
| **What is the use of finally block and illustrate with an example ?** |
| **Answer :** |
| **The finally block refers to a block of statements which will always be executed regardless of other exceptions that may or may not occur during an execution.** |
| **Finally Block (code) :** |
| **using System;**  **using System.Collections.Generic;**  **using System.Linq;**  **using System.Text;**  **using System.Threading.Tasks;**  **namespace random**  **{**  **internal class Program**  **{**  **static void Main(string[] args)**  **{**  **int a, b, c;**  **Console.WriteLine("\n enter integer a : ");**  **a = Convert.ToInt32(Console.ReadLine());**  **Console.WriteLine("\n enter integer b : ");**  **b = Convert.ToInt32(Console.ReadLine());**  **try**  **{**  **c = a / b;**  **Console.WriteLine($"Division of {a} / {b} = {c}");**  **}**  **catch (DivideByZeroException)**  **{**  **Console.WriteLine("\n value of b ( should be ) > 0");**  **}**  **//FINALLY BLOCK --------------**  **finally**  **{**  **Console.WriteLine("\n finally block ( executed! )");**  **Console.ReadLine();**  **}**  **}**  **}**  **}** |
| **Output :** |
|  |

| **Assignment 5** |
| --- |
| **What are the points about Exception Handling discussed in the session ?** |
| **Answer :** |
| * **Exception Handling is done to handle errors gracefully.** * **General exception block must be implemented at the end of all catch blocks.** * **Finally block statements are implemented irrespective of other exceptions.** * **One try block can have multiple catch blocks.**   **SYNTAX :** |

| **Assignment 6** |
| --- |
| **What are Compilation & Runtime errors and write at least 3 differences ?** |
| **Answer :** |
| | **Compilation Error** | **Runtime Error** | | --- | --- | | * **This error more often refers to semantic or syntax errors.** | * **This error refers to what we encounter during code execution during runtime screen.** | | * **Usually occurred due to developer mistakes.** | * **They usually are logical errors.** | | * **Easy to resolve these errors.** | * **These errors are hard to identify.** | |

| **Assignment 7** |
| --- |
| **Write any 5 compilation errors with code snippets and screenshots** |
| **Answer :** |
| 1. **Improper naming conventions :**      1. **Improper knowledge on type casting :**      1. **Improper importing of namespace :**      1. **Improper usage of methods and properties :**      1. **Improper Semicolon placement :** |

| **Assignment 8** |
| --- |
| **Write any 6 runtime errors with code snippets and screenshots.** |
| **Answer :** |
| 1. **DivideByZeroException :**        1. **NullReferenceException :**        1. **InvalidCastException :**        1. **OutOfMemoryException :**        1. **ArrayTypeMismatchException :**        1. **StackOverflowException :** |