|  |
| --- |
| Day12 Morning Assignment  By  Anusha Bellala  8-2-2022 |

|  |
| --- |
| 1. What is Exception Handling and why we need exception handling. |
| Exception Handling to ensure that our application will not crash or will not display any technical details and make sure we handle errors gracefully and display friendly messages.  **Need for Exception Handling:**  Exception Handling in c# is a process to handle run time errors. We perform exception handling so that normal flow of application can be maintained even after runtime errors. In c# exception is an event or object which is thrown at runtime. All exceptions the derived from System. |

|  |
| --- |
| 2. Write a simple division program and handle three exceptions discussed in the class., also add super exception at the last. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_Morning  {  internal class Program  {  static void Main(string[] args)  {  try  {  int a, b, c;  Console.WriteLine("Enter first number:");  a = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter second number:");  b = Convert.ToInt32(Console.ReadLine());  c = a / b;  Console.WriteLine("Answer={0}", c);  Console.WriteLine(“\n\n\n\n\n\designed by Anusha”);  Console.ReadLine();  }  catch(OverflowException)  {  Console.WriteLine("number should be 0 to 50000 are allowed");  }  catch(DivideByZeroException)  {  Console.WriteLine("cannot divide by zero");  }  catch(FormatException)  {  Console.WriteLine("only numbers are allowed. please double check");  }  catch(Exception)  {  Console.WriteLine("Any errors will occur contact admin@mycompany.com");  }  }  }  } |
| Ouput: |

|  |
| --- |
| 3. Research and write atleast 6 exceptions that occur in C# with sample code |

|  |
| --- |
| * **System.IO.DirectoryNotFoundException:** |
| Reason: The exception that is thrown when a part of file or directory cannot be found. |
| **Example code:**  StreamReader s = new StreamReader(@"c:\samplefiles\data.txt");  Console.WriteLine(s.ReadToEnd()); |
|  |

|  |
| --- |
| * **System.NullReferenceException**: |
| Reason: When we try to access a member on type whose value is null. |
| **Example code:**  string str = null;  Console.WriteLine(str.Substring(5));**m.NullReferenceException System.NullReferenceException** |
|  |

|  |
| --- |
| * **System.InvalidCastException**: |
| Reason: When the conversation of instance of one type to another type is not supported. |
| Example code:  bool flag = true;  IConvertible conv = flag;  Char ch = conv.ToChar(null); |
|  |

|  |
| --- |
| * **System.ArgumentOutOfRangeException:** |
| Reason: The ArgumentOutOfRangeException exception is thrown when the argument passed to a method is not null and contains a value that is not within the set of expected values |
| Example code:  var numbers = new List<int>();  var index = 0;  Console.WriteLine("Removing item at index {0}", index);  numbers.RemoveAt(index); |
|  |

|  |
| --- |
| * **System.OverflowException:** |
| Reason:  An OverflowException is thrown at run time under the following conditions: An arithmetic operation produces a result that is outside the range of the data type returned by the operation. |
| Example code:  int a = 2147483647;  int b = 2147483647;  int c = checked(a + b);  Console.WriteLine(c); |
|  |

|  |
| --- |
| * **System.OutofMemoryException:** |
| Reason: the System. OutOfMemoryException typically occurs when the common language runtime ( CLR ) is unable to allocate enough memory that would be necessary to perform the current operation. |
| Example code:  StringBuilder stringBuilder = new StringBuilder(17, 17);  stringBuilder.Append("Welcome to the ");    stringBuilder.Insert(0, "world of C# programming", 1);  Console.WriteLine(stringBuilder.ToString()); |
|  |

|  |
| --- |
| 4. What is the use of "finally" block illustrate with an example. |
| A finally block contains all the crucial statements that must be executed whether exception occurs or not. The statements present in this block will always execute regardless of whether exception occurs in try block or not such as closing a connection, stream etc. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp2  {  internal class Program  {  static void Main(string[] args)  {  try  {  int a, b, c;  Console.WriteLine("Enter first number:");  a = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter second number:");  b = Convert.ToInt32(Console.ReadLine());  c = a / b;  Console.WriteLine("Answer={0}", c);    Console.ReadLine();  }  catch (OverflowException)  {  Console.WriteLine("number should be 0 to 50000 are allowed");  }  catch (DivideByZeroException)  {  Console.WriteLine("cannot divide by zero");  }  catch (FormatException)  {  Console.WriteLine("only numbers are allowed. please double check");  }  catch (Exception)  {  Console.WriteLine("Any errors will occur contact admin@mycompany.com");  }  finally  {  Console.WriteLine("\n\n\n\n\nDesigned by Anusha");  Console.ReadLine();  }  }  }  } |
| Ouput: |

|  |
| --- |
| 5. Write the 5 points I explained about exception handling. |
| * Exception Handling is done to handle errors gracefully. so, that the application will not crash. * A single try block will have multiple catch blocks. * Always remember the general exception at last. * Statements will executed irrespective of Exception or not. * Syntax will written:   try  {  }  catch(Exception)  {  }  finally  {  } |

|  |
| --- |
| 6. What is compilation and Runtime error and Write atleast 3 differences between them. |

|  |  |
| --- | --- |
| **Compile time errors** | **Runtime errors** |
| 1.Compilation errors are generally referred to the error corresponding to the syntax or semantics. | .1.Runtime errors on the other hand refer to the error encountered during the execution of code at run time. |
| 2.Compilation errors get detected by compiler at the time of code development. | 2.Runtime errors are not get detected by compiler and hence identified at the time of code of execution. |
| 3.Compiletime errors already mentioned can get fixed at the time of code development. | 3.Runtime errors are getting to fixing state after once code get executed and errors get identified. |
|  |  |

|  |
| --- |
| 7.Write any 6 compilation errors with small code snippet.  Add compilation error screen shots. |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 8. Write any 6 runtime errors with small code snippets and add run time error screen shots. |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
| Inifinite loop:  using System;  using System.Collections;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp3  {  internal class Program  {  static void Main(string[] args)  {  int i = 5;  while(i<=5)  {  Console.WriteLine(i);  }  Console.ReadLine();  }  }  } |
| Output: |

|  |
| --- |
| using System;  using System.Collections;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace ConsoleApp3  {  internal class Program  {  static void Main(string[] args)  {  int n = 5;    for( int i=5; i < 3 ; i++)  Console.WriteLine(i);  Console.ReadLine();  }  }  } |
| Output: |