Day 12 Assignment

By K.Sanjay

|  |
| --- |
| 1. What is Exception Handling and why  we need exception handling. |
| Exception Handling:   * Exception Handling is done to ensure the our application will not trash * Will not display any technical details and to make sure we handle error gracefully and display friendly message * Exception is an event or object which is thrown at runtime. |

|  |
| --- |
| 2. Write a simple division program  and handle three exceptions discussed  in the class., also add super exception  at the last. |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  /\*\*\*\*\*\*\*\*\* Author : K.Sanjay \*\*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\* Purpose : Exception Handling \*\*\*\*\*\*\*\*/  namespace Day\_12\_Assignment\_8th\_Feb\_2022  {    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(c);  }  catch (OverflowException)  {  Console.WriteLine("Enter only 0 to 50000");  Console.ReadLine();  }  catch (DivideByZeroException)  {  Console.WriteLine("We cannot divide with zero");  Console.ReadLine();  }  catch(FormatException)  {  Console.WriteLine("we give only numbers.Please check again");  Console.ReadLine();  }  catch(Exception)  {  Console.WriteLine("Please contact Sanjay abcde@gmail.com");  }  Console.ReadLine();  }  }  } |
|  |

|  |
| --- |
| 3. Research and write atleast 6 exceptions that occur in C# with sample code. |
| 1.system. OutofMemory   * Reason: Errors that are generated due to insufficient free memory is handled by this excption. * Eample:   Public class check  {  Public static void main()  {  String val = new String(“k”,int.maxvalue);  }  } |
| 2.System.nullReferenceException   * Reason : Raised when program access members of null object.   Example : internal class Program  {  static void Main(string[] args)  {  Student std = null;  string name = std.StudentName;  Console.ReadLine();    }  } |
| 3.System.InvalidCastException   * Reason : The errors that are generated during typecasting handled by this exception. * Eample : using System.IO;   using System.Text;  class check  {  static void Main()  {  // an instance of the string builder class is created which is then assigned to a new object through implicit casting and then casting is tried explicitly to convert the instance of stringbuilder class to streamreader class  StringBuilder ref1 = new StringBuilder();  object ref2 = ref1;  StreamReader ref3 = (StreamReader)ref2;  }  } |
| 4.System.ArrayTypeMissmatchException   * Reason : The errors that are generated when there is a missmatch of type with array type is handled by this exception * Example: class check   {  static void Main()  {  // a string is defined and assigned the values which is then assigned to object class array and then an integer is tried to put in the same array which causes an exception  string[] arr1 = { "Welcome", "to", "CSharp" };  object[] arr2 = arr1;  arr2[0] = 8;  }  } |
| 5. System.IO.IOException   * Reason : The errors that are generated by input, the output is handled by this exception. * Example: using System;   using System.IO;  //a class called check is defined  class check  {  //main methos is called  static void Main()  {  try  {  //a file is tried to open which do not exist and causes an exception  File.Open("D:\\ex.txt", FileMode.Open);  }  catch (IOException)  {  Console.WriteLine("Inputoutput Exception is handled");  }  }  } |
| 6.System.TimeoutException   * Reason : The time interval alloted to an operation has expried. |

|  |
| --- |
| 4. What is the use of "finally" block illustrate with an example. |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  /\*\*\*\*\*\*\*\*\* Author : K.Sanjay \*\*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\* Purpose : Using Finally Exception \*\*\*\*\*\*\*\*/  namespace Day\_12\_Assignment\_8th\_Feb\_2022  {    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(c);  }  catch (OverflowException)  {  Console.WriteLine("Enter only 0 to 50000");    }  catch (DivideByZeroException)  {  Console.WriteLine("We cannot divide with zero");    }  catch(FormatException)  {  Console.WriteLine("We have to give only numbers.Please check again");    }  catch(Exception)  {  Console.WriteLine("Please contact Sanjay abcde@gmail.com");  }  finally  {  Console.WriteLine("\n\n\n\n\n\n Designed by SANJAY");  Console.ReadLine();  }    }  }  } |
| Output |
|  |

|  |
| --- |
| 5. Write the 5 points I explained about exception handling. |
| * Exception handling is done to handle exceptions or errors gracefully. So that the application will not crash. * A single try block can have multiple catch blocks. * Allways remember to write a general exception at the last. * Statements returns inside the finally block will be executed irrespective of wether exception occurs or not. * The general syntax or flow for write exception is try,catch and finally block. |

|  |
| --- |
| 6. What is compilation and Runtime error  Write atleast 3 differences between them |
| Compilation Errors   1. Compilation can easily detect compile time errors during the writing of code. 2. Compile time error generally refers to the errors that corresponding to the syntax. 3. We can easily fix a compile time error. |
| Runtime Errors |
| 1. A compiler cannot easily detect a runtime error. 2. A runtime error refers to the error that we encounter during the code execution during run time 3. Compiler cannot identify a run time error but we can fit after the execution of code. |

|  |
| --- |
| 7. Write any 6 compilation errors with small code snippet.  Add compilation error screen shots. |
| 1.Use of unassigned local variable ‘p’ |
| 2.Semicolumn Expected |
| 3.Spelling mistake |
| 4.Namespace |
| 5.syntax error |
| 6.Logical Error |

|  |
| --- |
| 8. Write any 6 runtime errors with small code snippets and add  run time error screen shots. |
| Runtime Error |
| These are the types in runtime errors   1. Logic Error. A logic error occurs when a developer enters the wrong statements into the application's source code. ... 2. Memory Leak. ... 3. Division by Zero Error. ... 4. Undefined Object Error. ... 5. Input/Output Device Error. ... 6. Encoding Error. |