12th Day Assignment

By

B.P.N.V.S.Sudheer

08-02-22

|  |
| --- |
| 1.What is exception handling and why we need exception handling |
| * Exception Handling is done to ensure that our application will not crush |
| * It will not display any technical details to make sure we handle errors gracefully and display friendly messages |
| 2.Write a simple division program and handle 3 exceptions discussed in the class also add super exception at the last |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Runtime.Serialization;  using System.Text;  using System.Threading.Tasks;    namespace ConsoleApp1  {  internal class Program  {  static void Main(string[] args)  {  try  {  int a, b, c;  Console.WriteLine("enter a value");  a = Convert.ToInt32(Console.ReadLine());    Console.WriteLine("enter b value");  b = Convert.ToInt32(Console.ReadLine());    c = a / b;  Console.WriteLine(c);  Console.WriteLine("\n\n\n\n\n Design by sudheer");  Console.ReadLine();  }  catch (OverflowException)  {  Console.WriteLine("only numbers 1 to 50000");  Console.WriteLine("\n\n\n\n\n Design by sudheer");  Console.ReadLine();    }  catch (DivideByZeroException)  {  Console.WriteLine("divide number without zero");  Console.WriteLine("\n\n\n\n\n Design by sudheer");  Console.ReadLine();    }    catch (FormatException)  {  Console.WriteLine("wrong input will enter");  Console.WriteLine("\n\n\n\n\n Design by sudheer");  Console.ReadLine();    }  catch (Exception)  {  Console.WriteLine("some errors occured contact by @sudheer.com");  Console.ReadLine();    }        }  }  } |
| Output: |
|  |

|  |
| --- |
|  |
|  |
|  |

|  |
| --- |
| 3.what is the use of finally block illustrate with an example |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Runtime.Serialization;  using System.Text;  using System.Threading.Tasks;    namespace ConsoleApp1  {  internal class Program  {  static void Main(string[] args)  {  try  {  int a, b, c;  Console.WriteLine("enter a value");  a = Convert.ToInt32(Console.ReadLine());    Console.WriteLine("enter b value");  b = Convert.ToInt32(Console.ReadLine());    c = a / b;  Console.WriteLine(c);  Console.WriteLine("\n\n\n\n\n Design by sudheer");  Console.ReadLine();  }  catch (OverflowException)  {  Console.WriteLine("only numbers 1 to 50000");  Console.WriteLine("\n\n\n\n\n Design by sudheer");      }  catch (DivideByZeroException)  {  Console.WriteLine("divide number without zero");  Console.WriteLine("\n\n\n\n\n Design by sudheer");    }  catch (FormatException)  {  Console.WriteLine("wrong input will enter");  Console.WriteLine("\n\n\n\n\n Design by sudheer");      }  catch (Exception)  {  Console.WriteLine("some errors occured contact by @sudheer.com");      }  finally  {  Console.WriteLine("@contact");  Console.ReadLine();  }        }  }  } |
| Output: |
|  |

|  |
| --- |
| 4.write the points explained about exception |
| * Exception handling that our applcation will not crush we handle errors gracefully and display friendly messages |
| * A single try block can have multiple blocks |
| * Always have general exception should be at last |
| * Statement return inside the final block will be excecuted irrespective of wheather the exception error (or) not statement inside the final box will be executed all the time |
| * Syntax of writing exceptions |
| try  {    }  catch(exception)  {  }  finally  {  } |
| 5.What is compliation and runtime error write atleast 3 differences between them |

|  |  |
| --- | --- |
| Compliation error | Run time error |
| 1.Compliation errors are generally referred to the error corresponding to syntax | 1.Runtime errors it does not show compliation error but it execute the code wrongly |
| 2.compile time errors get detected by complier at the time of code development | 2.Runtime errors are not get detected by complier and we identify the error |
| 3.compile time errors as 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 |

|  |
| --- |
| 6.Write any 6 compilation errors with small code snippet add compilation error screen shots |
| (a)Readline error |
|  |
| (b)semicolon error |
|  |
| (c)writeline error |
|  |
| (d)Conversion error |

|  |
| --- |
|  |
| (e)flower bracket error |
|  |
| (f)Length expected error |
|  |

|  |
| --- |
| 7.write 6 run time errors with small code snippet |
| (a)Output run time error |
|  |
| (b)index outofrange |
|  |
| (c)Divide by zero |
|  |
| (d)overflow error |

|  |
| --- |
|  |
| (e)Format error |
|  |