|  |
| --- |
| DAY 12 MORNING ASSIGNMENT  DATE:08/02/2022  DAY :TUESDAY  M.SAI HARICHANDANA |
| 1. What is exception handling and why do need exception handling ?  * EH is done to ensure that our application will not crash (or)will not display any technical details and to make sure that we handle eassy gracefully display friendly messages |

|  |
| --- |
| 2.Write a simple division program and handle three discussed in the class also and super exceptions that occur in c# with sample code ? |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace DAY\_12\_ASSIGNMENTS  {    namespace Day\_12\_\_Assignments  {  internal class Program  {  static void Main(string[] args)  {  try  {  int s1, s2, s3;  Console.WriteLine("Enter a value");  s1 = Convert.ToInt32(Console.ReadLine());  Console.WriteLine(" Enter b value");  s2 = Convert.ToInt32(Console.ReadLine());  s3 = s1 / s2;  Console.WriteLine(s3);  Console.ReadLine();  }  catch (DivideByZeroException ex)  {  Console.WriteLine("Denominator is zero.Enter a valid input");  }  catch (FormatException ex)  {  Console.WriteLine("Format is not correct .Enter a numerical value");  }  catch (OverflowException ex)  {  Console.WriteLine("Given input is out of range .Enter a value between 1 to 1000000000");  }  catch (Exception ex)  {  Console.WriteLine("some errors occured");  }  Console.ReadLine();  }  }  }  } |
| OUTPUT : |
| DIVIDE ZERO EXCEPTION    FORMAT EXCEPTION :    OVERFLOW EXCEPTION : |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | 5. Write the five points explained about exception handling ?   * Exception handling is done to handle to error exceptions was gracefully so that applications will not crash and without displaying   Any errors to the end customer.   * A single try block can have multiple catch blocks * Always remember to write general exceptions at the last * Statement which are in finally gets executed always at irrespective of occurance of exception * The general syntax (or) a flow for writing exception is try, catch, finally | | 6. What is compilation error and runtime error and write the difference between ? | | * COMPILATION ERROR: * When we compile the code the errors we get when the code doesn,t get compiled are called compilation errors |  * RUNTIME ERROR :      * A Runtime error in a program is an error that occurs while the   Program is running after being successfully compiled runtime errors are commonly called referred to as “bugs “.   |  |  | | --- | --- | | COMPILATION ERROR | RUN TIME ERROR | | 1.Ann error identified by the compiler is called a compile error | 1.A run time error causes the program to terminate abnormally during execution. | | 2.if a compile error occurs ,an executable version of the program is not created | 2.An example is an runtime to divide by zero | | 3.use your editor to correct the error then recompile your program | 3.In java many runtime errors are represented as exceptions that can be caught and dealt |  |  | | --- | | 4. what is the use of “ finally” block illustrate with an example ? | | * Statements inside finally block executed whether execution occurs   (or) doesn,t occur   * It is used to display some common message. * It is used to close any opened data base connections (or) any opened connects |  |  | | --- | | Code : | | using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day\_12\_\_Assignments  {  internal class Program  {  static void Main(string[] args)  {  try  {  int s1, s2, s3;  Console.WriteLine("Enter a value");  s1 = Convert.ToInt32(Console.ReadLine());  Console.WriteLine(" Enter b value");  s2 = Convert.ToInt32(Console.ReadLine());  s3 = s1 / s2;  Console.WriteLine(s3);    }  catch (DivideByZeroException ex)  {  Console.WriteLine("Denominator is zero.Enter a valid input");  }  catch (FormatException ex)  {  Console.WriteLine("Format is not correct .Enter a numerical value");  }  catch (OverflowException ex)  {  Console.WriteLine("Given input is out of range .Enter a valuebetween 1 to 1000000000");  }  catch (Exception ex)  {  Console.WriteLine("some errors occured");  }  finally  {  Console.WriteLine("\n\n\n\n\n\n Designed by Chandana");  Console.ReadLine();  }  }  }  }  Output : |  |  | | --- | | 8. write any six runtime errors with small code snippets and add run time error screen shots | | * {   Int a=10;  Console. WriteLine(a/b);  } | |  | |  |  * .int a,b,c;   Console.WriteLine(“Enter a number”);  a =Convert .ToInt32(Console.ReadLine());  Console.WriteLine(“Enter a number”);  b =Convert. ToInt32(Console.ReadLine();  c =a/b;  Console.WriteLine(c);  Console.ReadLine();     * int a,b,c;   Console.WriteLine(“Enter a number”);  a =Convert .ToInt32(Console.ReadLine());  Console.WriteLine(“Enter a number”);  b =Convert. ToInt32(Console.ReadLine();  c =a/b;  Console.WriteLine(c);  Console.ReadLine();     * Int [ ] data =new int [5]   Data[6] =11;     |  | | --- | | Write any six compilation errors with small code snippet add compilation error screen shots ? | | * Missing paranthasis   Int a = 10;  Console.WriteLine(a);  }   * Not intialising variable:   {  Int a;  Console.WriteLine();  }     * .when we give spelling mistakes or if there is a case difference:   {  Int a =10;  Console.WriteLine();  }     * When namespace is not imported   {  Int a=10;  Console.WriteLine(a);  } | | * When semicolon is removed:   {  Int a =10;  Console.WriteLine(a);  } | | * When the variable id=s not declared :   {  Console.WriteLine(“Enter a value”)’  b = Console.ReadLine();  } | |
|  |
|  |