|  |
| --- |
| Day12 morning assignment  By  Paluru Mounika  08-02-2022 |

|  |
| --- |
| **1.what is exception handling and why we need exception handling?** |
| **Ans:**  **Exeption handling:**  1.Exeption handling is the process to handle the run-time errors  **Why we need exeption handling:**  -exeption handling is done to ensure that our application will not crash.  -will not display any technical details .  -to make sure we handle errors gracefully and display friendly messages. |

|  |
| --- |
| **2.write simple division program and handle 3 exceptions discussed in the class and also add super exception.** |
| **Program:** division with 3 exception |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Author:paluru mounika  //Purpose: division program using exception handling  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day12project1  {  internal class Program  {  static void Main(string[] args)  {    try  {  int a, b, c;  Console.WriteLine("Enter a");  a = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter b");  b = Convert.ToInt32(Console.ReadLine());  c = a / b;  Console.WriteLine(c);  Console.ReadLine();    }    catch (OverflowException )  {  Console.WriteLine("the numbers between 0 t0 600000" );  Console.ReadLine();  }  catch (DivideByZeroException )  {  Console.WriteLine("cannot divide with Zero");  Console.ReadLine();  }  catch(FormatException )  {  Console.WriteLine("only numbers are allowed");  Console.ReadLine();  }  catch(Exception )  {  Console.WriteLine("some error accure please contact mouni@gmail.com");  Console.ReadLine();  }  }  }  } |
| **Output:** |
| **Overflowexception Formateexception**    **DivideByZeroException** |

|  |
| --- |
| **3.research and write atlest 6 exceptions that occur in c# witx df** |
| **a)IndexOutofRangeException:**  **code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Main(string[] args)  {  int[] ar = { 3, 4, 6, 7, 2 };  //causing exception  for(int i=0;i<=ar.Length;i++)  Console.WriteLine(ar[i]);  }      }  } |
| **b)out of memory exception:**  **code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Main(string[] args)  {  string val = new string('r', int.MaxValue);  Console.ReadLine();    }      }  } |
| **Null reference exception:**  **Code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Main(string[] args)  {  int[] values = null;  for (int ctr = 0; ctr <= 9; ctr++)  values[ctr] = ctr \* 2;  foreach(var value in values)  Console.WriteLine(value);  Console.ReadLine();    }      }  } |
| **Array type mismatch exception:**  **Code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Main(string[] args)  {  string[] array1 = { "fan", "table", "tv" };  object[] array2 = array1;  array2[0] = 5;  Console.ReadLine();      }      }  } |
| **Stack over flow exception:**  **Code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Recurse(int val)  {  Console.WriteLine(val);  Recurse(++val);  }  static void Main(string[] args)  {  Recurse(1);  Console.ReadLine();    }      }  } |

|  |
| --- |
| **4.what is the use of “finally” block,illustrate with an example.** |
| 1. Finally block is used to excute a given set of statements whether an exception is thrown or not thrown.   **Example:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_3\_Exception\_Handlings  {  internal class Program  {  static void Main(string[] args)  {  try  {  int a, b, c;  Console.WriteLine("Enter a: ");  a = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter b: ");  b = Convert.ToInt32(Console.ReadLine());  c = a / b;  Console.WriteLine(c);  Console.ReadLine();  }  catch (FormatException)  {  Console.WriteLine("Enter Numbers. Please check again");  Console.ReadLine();  }  catch (DivideByZeroException)  {  Console.WriteLine("can't divide by zero");  Console.ReadLine();  }  finally  {  Console.WriteLine("Error Occured");  Console.ReadLine();  }  }  }  } |

|  |
| --- |
| **5.write the 6 points explained about exception.** |
| 1.Exception handling is done to handle error any system should not crash |
| 2.single try block can have multiple catch blocks. |
| 3.always write general exception at last. |
| 4.statements inside the finally block will be excuted by irrespective of catch block. |
| 5.general syntax:  1.try  2.catch  3.finally |

|  |  |
| --- | --- |
| **6.what is compilation error and run time error write 3differences?** | |
| **Compilation error** | **Runtime error** |
| Errors that correspond to the syntax or semantics | Errors that we encounter during the code execution during runtime. |
| Compiler can easily detect compile-time errors during the development of code | Compiler cannot detect the error in runtime. Thus we need to identify it during execution of code. |
| We can easily fix a compile-time error during the development of code**.** | A compiler cannot identify a run-time error. But we can fix it after the execution of code. |
| Occurs due to violation | Occurs due to performing an illeagal oparations |

|  |
| --- |
| **7.write 6 compilation error with small code snippet and put screen shorts** |
| |  | | --- | | **Missig semicolon:** | | **lowercase letters:** | | **Name space not included:** | | **Spelling mistake:** | | **Value not declare:** | | **Missing parenthesis:** | |

|  |
| --- |
| **8.write any 6 Runtime errors with small code snippets and add runtime errors screenshots.** |
| **a.IndexoutofrangeException:**  **code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Main(string[] args)  {  int[] ar = { 3, 4, 6, 7, 2 };  //causing exception  for(int i=0;i<=ar.Length;i++)  Console.WriteLine(ar[i]);  }      }  } |
|  |
| **b.out of memory exception**  **code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Main(string[] args)  {  string val = new string('r', int.MaxValue);  Console.ReadLine();    }      }  } |
| **Error:** |
| **c.null reference exception**  **code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Main(string[] args)  {  int[] values = null;  for (int ctr = 0; ctr <= 9; ctr++)  values[ctr] = ctr \* 2;  foreach(var value in values)  Console.WriteLine(value);  Console.ReadLine();    }      }  }  **Error:** |
| |  | | --- | | **Array type mismatch**  **Code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Main(string[] args)  {  string[] array1 = { "fan", "table", "tv" };  object[] array2 = array1;  array2[0] = 5;  Console.ReadLine();      }      }  } | | **Error:** | | **Stack over flow exception:**  **Code:**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace day12project  {  internal class Program  {  static void Recurse(int val)  {  Console.WriteLine(val);  Recurse(++val);  }  static void Main(string[] args)  {  Recurse(1);  Console.ReadLine();    }      }  } | | **Error:** | |  | |  | |