|  |
| --- |
| **Day12 Morning Assignments**  **By**  **Manoj Yekolla**  **08-feb-2022** |

|  |
| --- |
| **1. What is Exception Handling and why we need exception handling.** |
| * Exception Handling is done to ensure that our application will not crash. * Will not display any technical details and to make sure we handle errors gracefully and display friendly messages. * Exception Handling is a process to handle Runtime errors. |

|  |
| --- |
| **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\_Project1  {  internal class Program  {  static void Main(string[] args)  {  try  {  int a, b, c;  Console.WriteLine("Enter the First Number :");  a = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter the Second Number :");  b = Convert.ToInt32(Console.ReadLine());  c = a / b;  Console.WriteLine("Answer :{0}", c);  Console.ReadLine();    }  catch (OverflowException )  {  Console.WriteLine("---------Overflow Exception--------------");  Console.WriteLine("Enter 0 to 8000000 and above does not valid :");  Console.ReadLine();  }  catch (DivideByZeroException)  {  Console.WriteLine("--------Divide By Zero Exception-------");  Console.WriteLine("canot divide by zero ");  Console.ReadLine();  }  catch (FormatException)  {  Console.WriteLine("-----------FormatException---------");  Console.WriteLine("only Numbers are allowed ");  Console.ReadLine();  }  catch (Exception) //super exception  {  Console.WriteLine("---------Super Exception----------");  Console.WriteLine("some errors are occured : please contact manojy@gmail.com");  Console.ReadLine();  }  }  }  } |
| Output :  Screenshot (186) |
| Screenshot (187) |
| Screenshot (188) |
| Screenshot (189) |

|  |
| --- |
| **3. Research and write exceptions that occur in C# with sample code ?** |
| 1. **IndexOutOfRange Exception :** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_Project1  {  internal class Program  {  static void Main(string[] args)  {  int[] ar = { 1, 2, 3, 4, 5 };  // causing exception  for (int i = 0; i <= ar.Length; i++)  Console.WriteLine(ar[i]);  }  }  } |
| Exception :  Screenshot (191) |
|  |

|  |
| --- |
| **3(b) Out Of Memory Exception ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Main(string[] args)  {  string val = new string('r', int.MaxValue);  Console.ReadLine();  }  }  } |
| Output :  Screenshot (197) |

|  |
| --- |
| **3(c) Null Reference Exception ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  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();  }  }  } |
| Output  Screenshot (199) |

|  |
| --- |
| **3(d) Array Type Mismatch Exception ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Main(string[] args)  {  string[] array1 = { "cat", "dog", "fish" };  object[] array2 = array1;  array2[0] = 5;  Console.ReadLine();  }  }  } |
| Output :  Screenshot (201) |

|  |
| --- |
| **3(e) Stack Overflow Exception ?** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Recurse(int val)  {  Console.WriteLine(val);  Recurse(++val);  }    static void Main(string[] args)  {  Recurse(1);  Console.ReadLine();  }  }  } |
| Output :  Screenshot (203) |

|  |
| --- |
| **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;  namespace Day12\_Project2  {  internal class Program  {  static void Main(string[] args)  {  try  {  int a, b, c;  Console.WriteLine("Enter the First Number :");  a = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter the Second Number :");  b = Convert.ToInt32(Console.ReadLine());  c = a / b;  Console.WriteLine("Answer :{0}", c);  Console.ReadLine();  }  catch (OverflowException)  {  Console.WriteLine("---------Overflow Exception--------------");  Console.WriteLine("Enter 0 to 8000000 and above does not valid :");    }  catch (DivideByZeroException)  {  Console.WriteLine("--------Divide By Zero Exception-------");  Console.WriteLine("canot divide by zero ");    }  catch (FormatException)  {  Console.WriteLine("-----------FormatException---------");  Console.WriteLine("only Numbers are allowed ");    }  catch (Exception) //super exception  {  Console.WriteLine("---------Super Exception----------");  Console.WriteLine("some errors are occured : please contact manojy@gmail.com");    }  finally  {  Console.WriteLine("\n\n\nDesigned by Manoj Yekolla");  Console.ReadLine();  }  }  }  } |
| Output :  Screenshot (193) |
| Screenshot (195) |

|  |
| --- |
| **5. Write the 5 points I explained about exception handling.** |
| * Exception Handling is done to ensure that our application will not crash. * A Single Try block can have multiple catch blocks. * Aways a General Exception is last , If you right it top it will handle all exception remaining exceptions are error. * Statements return inside finally block will be executed irrespective wether exception or not. * A generally syntax flow for writing exception is- Try ,Catch ,Finally blocks.   try  {  // Block of Code To Be Executed  }  catch (Exception)  {  // Exception Message, for displaying purpose.  }  finally  {  // These Block of statements will be executed  } |

|  |
| --- |
| **6. What is compilation and Runtime error**  **Write atleast 3 differences between them** |
| **Compilation Error :**   * These is to check syntax and sematics code. * These type of errors can easily identify and can be resolved. * Most compilation errors are developer mistakes |
| **Runtime errors**   * These is to run the code. * Most runtime errors are Logical errors. * These errors are hard to identify and resolve them. |

|  |
| --- |
| **7. Write any compilation errors with small code snippet.**  **Add compilation error screen shots.** |
| 1. Not using Proper Naming convention |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Main(string[] args)  {  string a = "Nb Health care Tech";  console.WriteLine(a);  }  }  } |
| Screenshot (205) |

|  |
| --- |
| **7(b) Not using Semicolon ?** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Main(string[] args)  {  string a = "Nb HealthCare Tech";  Console.WriteLine(a)  }  }  } |
| Screenshot (205) |

|  |
| --- |
| **7(c) Missing Naming convention ?** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Main(string[] args)  {  string a = "Nb HealthCare Tech";  Console.Writeline(a);  }  }  } |
| Screenshot (205) |

|  |
| --- |
| **8. Write any runtime errors with small code snippets and add**  **run time error screen shots.** |
| 1. **Runtime Error -Array Type MissMatch** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Main(string[] args)  {  string[] arr1 = { "Hi", "yekolla" };  object[] arr2 = arr1;  arr2[0] = 15;  Console.ReadLine();  }  }  } |
| Output :  Screenshot (207) |

|  |
| --- |
| **8(b) Null Reference Run time error ?** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Main(string[] args)  {  string value = null;  if(value.Length == 0)  {  Console.WriteLine(value);  }  Console.ReadLine();    }  }  } |
| Output :  Screenshot (209) |

|  |
| --- |
| **8(c) Out of Memory Runtime error ?** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project2  {  internal class Program  {  static void Main(string[] args)  {  string val = new string('m', int.MaxValue);  Console.ReadLine();    }  }  } |
| Output :  Screenshot (211) |