|  |
| --- |
| **Day-12 assignment**  **By**  **Bhanu Rama Krishna Prakash Jakkamsetti**  **8/2/2022** |

|  |
| --- |
| 1.What is exception handling and why we need exception handling. |
| * Exception handling ensures that the flow of the program doesn’t break when an exception occurs. For example, if a program has bunch of statements and an exception occurs midway after executing certain statements then the statements after the exception will not execute and the program will terminate abruptly. |

|  |
| --- |
| 2.Write a sample division program and handle three exceptions discussed in the class, also add super exception at last. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project1  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:Bhanu rama krishna prakash jakkmsetti  \* purpose:exception handling  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  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);  int[] arr = new int[5];  arr[8] = 1;  Console.ReadLine();  }  catch (OverflowException e)  {  Console.WriteLine("enter value in 0-5000 only");  }  catch(DivideByZeroException e)  {  Console.WriteLine("divide by zero will not allow");  }  catch(FormatException e)  {  Console.WriteLine("in correct format or input");  }  catch(IndexOutOfRangeException e)  {  Console.WriteLine("check the index value is with in range only");  }  catch(Exception e)  {  Console.WriteLine("contact bhanu ");  }  }  }  } |
| Output: |
|  |

|  |
| --- |
| 3.Research and write at least 6 exceptions that occur in C# with sample code. |
| 1.OutOfMemoryException |
| Reason: raised when a program does not get enough memory to execute the code. |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  try  {  string a = new  string('r', int.MaxValue);  }  catch (OutOfMemoryException e)  {  Console.WriteLine("program does not get enough memory to execute the code");  }  }  }  } |
|  |
| 2.NullReferenceException |
| Reason: raised when program access members of null object. |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  try  {  string a = null;  if (a.Length == 0)  {  Console.WriteLine(a);  }  }  catch (NullReferenceException e)  {  Console.WriteLine("program access members of null object");  }  }  }  } |
|  |
| 3.InvalidCastException |
| Reason: check is not defined |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  try  {  StringBuilder a = new StringBuilder();  object b=new object();  StreamReader c = (StreamReader)b;  }  catch (InvalidCastException e)  {  Console.WriteLine("check is not defined");  }  }  }  } |
|  |
| 4.ArrayTypeMismatchException |
| Reason: array type mismatch |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  try  {  string[] arr= { "welcome","to","NB"};  object[] arr2 = arr;  arr2[0] = 8;  }  catch (ArrayTypeMismatchException e)  {  Console.WriteLine("check is not defined");  }  }  }  } |
|  |
| 5.IOException |
| Reason: input output exception |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {    static void Main(string[] args)  {    try  {  File.Open("D:\\ex.txt", FileMode.Open);  }  catch (IOException)  {  Console.WriteLine("input output exception");  }    }  }  } |
|  |
| 6.ArgumentException |
| Reason:method calling itself |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Print(int a)  {  Console.WriteLine(a);  Print(++a);  }  static void Main(string[] args)  {  try  {  Print(0);  }  catch (StackOverflowException e)  {  Console.WriteLine("recursion error");  Console.ReadLine();  }  }  }  } |
|  |

|  |
| --- |
| 4.What is the use of finally block illustrate with an example. |
| * The mandatory statements print inside finally block is using whether we exception or not. |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project3  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author:Bhanu rama krishna prakash jakkmsetti  \* purpose:exception handling by using final block  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  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);  int[] arr = new int[5];  arr[8] = 1;  Console.ReadLine();  }  catch (OverflowException e)  {  Console.WriteLine("enter value in 0-5000 only");  }  catch (DivideByZeroException e)  {  Console.WriteLine("divide by zero will not allow");  }  catch (FormatException e)  {  Console.WriteLine("in correct format or input");  }  catch (IndexOutOfRangeException e)  {  Console.WriteLine("check the index value is with in range only");  }  catch (Exception e)  {  Console.WriteLine("contact bhanu ");  }  finally  {  Console.WriteLine(" designed by Bhanu");  Console.ReadLine();  }  }  }  } |
| Output: |
|  |

|  |
| --- |
| 5.Write the points I explained about exception handling. |
| * Exception handling is done gracefully handle the errors * In a single try block can have multiple catch blocks. * Always remember to write general exception at last. * Statements inside finally block is using whether we exception or not. * General syntax for writing Exception handling is * try * catch * final |

|  |  |
| --- | --- |
| 6. What is compilation and runtime error. Write at least 3 difference between them. | |
| Compile-time errors | Runtime errors |
| * These are the syntax errors which are detected by the compiler. | * These are the errors which are not detected by the compiler and produce wrong results. |
| * They prevent the code from running as it detects some syntax errors. | * They prevent the code from complete execution. |
| * It includes syntax errors such as missing of semicolon (;), misspelling of keywords and identifiers etc. | * It includes errors such as dividing a number by zero, finding square root of a negative number etc. |

|  |
| --- |
| 7.Write any six compilation errors with small code snippets and add screenshots. |
| 1.when name space not allowed |
| //using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  int a = 8;  int b = 10;  int c = a / b;  Console.WriteLine(c);  Console.ReadLine();    }  }  } |
|  |
| 2.when we didn’t put ; |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  int a = 8;  int b = 10;  int c = a / b  Console.WriteLine(c);  Console.ReadLine();    }  }  } |
|  |
| 3.spelling mistake |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  int a = 8;  int b = 10;  int c = a / b;  Console.WriTeLine(c);  Console.ReadLine();    }  }  } |
|  |
| 4.using un assigned value |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  int a = 8.5;  int b = 10;  int c = a / b;  Console.WriteLine(c);  Console.ReadLine();    }  }  } |
|  |
| Calling the variable without using obj |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  class Lol  {  int a = 20;  int b = 10;  int c = a / b;  }  internal class Program  {  static void Main(string[] args)  {  Lol lol = new Lol();  Console.WriteLine(c);  Console.ReadLine();    }  }  } |
|  |
| By change the name space name |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_ project4  {  internal class Program  {  static void Main(string[] args)  {  int a = 20;  int b = 10;  int c = a / b;  Console.WriteLine(c);  Console.ReadLine();    }  }  } |
|  |

|  |
| --- |
| 8.Write any six runtime errors with small code snippets and add screenshots. |
| 1.calling out of index |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  int[] arr = new int[5];  arr[6] = 1;  }  }  } |
|  |
| 2.divide by zero exception |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  int a = 10;  int b = 0;  int c = a/b;  }  }  } |
|  |
| 3.large values giving |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  int a =Convert.ToInt32(Console.ReadLine());  Console.ReadLine();  }  }  } |
|  |
| 4.in correct format |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  int a =Convert.ToInt32(Console.ReadLine());  Console.ReadLine();  }  }  } |
|  |
| 5.out of memory |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  string a = new  string('r',int.MaxValue);  }  }  } |
|  |
| 6.null reference |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Day12\_project4  {  internal class Program  {  static void Main(string[] args)  {  string a = null;  if (a.Length==0)  {  Console.WriteLine(a);  }  }  }  } |
|  |