1.Write a program that takes a positive integer from the console and prints the **square root** of this integer. If the input is **negative or invalid** print "Invalid Number" in the console. In all cases print "Good Bye".

|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Linq; |
|  | using System.Text; |
|  | using System.Threading.Tasks; |
|  |  |
|  | namespace Chapter\_12\_Solution\_4 |
|  | { |
|  | class Program |
|  | { |
|  | static void Main(string[] args) |
|  | { |
|  | Console.Write("Enter number: "); |
|  | string input = Console.ReadLine(); |
|  | int n = -1; |
|  | bool invalidNumber = false; |
|  |  |
|  | try |
|  | { |
|  | n = Convert.ToInt32(input); |
|  | } |
|  | catch (FormatException e) |
|  | { |
|  | Console.WriteLine("Invalid number!"); |
|  | invalidNumber = true; |
|  | } |
|  | finally |
|  | { |
|  | if (n < 0) |
|  | { |
|  | if (!invalidNumber) Console.WriteLine("Invalid number!"); |
|  | } |
|  | else Console.WriteLine(Math.Sqrt(n)); |
|  | } |
|  |  |
|  | Console.WriteLine("Good Bye"); |
|  | } |
|  | } } |
|  |  |

2. Write a method ReadNumber(int start, int end) that reads an integer from the console in the range [start…end]. In case the input integer is not valid or it is not in the required range throw appropriate exception. Using this method, write a program that takes 10 integers a1, a2, …, a10 such that 1 < a1 < … < a10 < 100.

|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Linq; |
|  | using System.Text; |
|  | using System.Threading.Tasks; |
|  |  |
|  | namespace Chapter\_12\_Solution\_5 |
|  | { |
|  | class Program |
|  | { |
|  | static void ReadNumber(int start, int end) |
|  | { |
|  | int count = 1, number; |
|  |  |
|  | do |
|  | { |
|  | Console.Write("Number{0}: ", count); |
|  | number = Int32.Parse(Console.ReadLine()); |
|  |  |
|  | if (number >= end || number <= start) |
|  | { |
|  | Console.WriteLine("Wrong input!"); |
|  | break; |
|  | } |
|  | else |
|  | start = number; |
|  |  |
|  | count++; |
|  | } while (count < 11); |
|  | } |
|  |  |
|  | static void Main(string[] args) |
|  | { |
|  | Console.Write("Start: "); |
|  | int start = Int32.Parse(Console.ReadLine()); |
|  | Console.Write("End: "); |
|  | int end = Int32.Parse(Console.ReadLine()); |
|  |  |
|  | if (end <= start + 10) |
|  | Console.WriteLine("Wrong input"); |
|  | else |
|  | ReadNumber(start, end); |
|  | } |
|  | } } |
|  |  |

3.  Write a method that takes as a parameter the name of a **text file, reads the file and returns its content as** string. What should the method do if and **exception is thrown**?

|  |
| --- |
| namespace Chapter\_12\_Solution\_6 |
|  | { |
|  | class Program |
|  | { |
|  | static void Main(string[] args) |
|  | { |
|  | try |
|  | { |
|  | using (StreamReader sr = new StreamReader("TestFile.txt")) |
|  | { |
|  | String line = sr.ReadToEnd(); |
|  | Console.WriteLine(line); |
|  | } |
|  | } |
|  | catch (Exception e) |
|  | { |
|  | Console.WriteLine("The file could not be read:"); |
|  | Console.WriteLine(e.Message); |
|  | } |
|  | } |
|  | } |
|  | } |

4. Write a program that gets from the user the full path to a file (for example C:\\Windows\\win.ini), reads the content of the file and prints it to the console. Find in MSDN how to us the System.IO.File.ReadAllText(…) method. Make sure all possible exceptions will be caught and a user-friendly message will be printed on the console.

|  |
| --- |
| namespace Chapter\_12\_Solution\_6 |
|  | { |
|  | class Program |
|  | { |
|  | static void Main(string[] args) |
|  | { |
|  | try |
|  | { |
|  | using (StreamReader sr = new StreamReader(@"C:\Users\Ivan\Desktop\text.txt")) |
|  | { |
|  | String line = sr.ReadToEnd(); |
|  | Console.WriteLine(line); |
|  | } |
|  | } |
|  | catch (FileNotFoundException e) |
|  | { |
|  | Console.WriteLine(e.Message); |
|  | } |
|  | catch (DirectoryNotFoundException e) |
|  | { |
|  | Console.WriteLine("The specified path is invalid!"); |
|  | } |
|  | catch (IOException e) |
|  | { |
|  | Console.WriteLine("Invalid syntax!"); |
|  | } |
|  | } |
|  | } |
|  | } |

5. Write a program that **downloads a file from Internet** by given URL, e.g. (<http://www.devbg.org/img/Logo-BASD.jpg>).

|  |
| --- |
| namespace Chapter\_12\_Solution\_8 |
|  | { |
|  | class Program |
|  | { |
|  | static void Main(string[] args) |
|  | { |
|  | WebClient Client = new WebClient(); |
|  |  |
|  | try |
|  | { |
|  | Client.DownloadFile("http://3.bp.blogspot.com/-qXtmJRAlJcA/U413iy\_YzKI/AAAAAAAAOn8/Ajr4B8h9TcE/s1600/google-logo-high-res.png", @"C:\Users\Ivan\Desktop\image.png"); |
|  | } |
|  | catch (ArgumentException) |
|  | { |
|  | Console.WriteLine("The address or fileName parameter is null!"); |
|  | } |
|  | catch (WebException) |
|  | { |
|  | Console.WriteLine("Error! Possible causes:\n1. The URI formed by combining BaseAddress and address is invalid.\n2. filename is null or Empty.\n3. The file does not exist.\n4. An error occurred while downloading data."); |
|  | } |
|  | catch (NotSupportedException) |
|  | { |
|  | Console.WriteLine("The method has been called simultaneously on multiple threads."); |
|  | } |
|  | } |
|  | } } |
|  |  |