```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;
using System.Threading;
using System.IO;
namespace peerMethods__Childress_20160210
{
    class Program
    {

        //AREA OF A CIRCLE QUINN
        static double AreaOfCircle(double x)
        {
            double area = Math.PI * (x * x);
            return area;
        }
        static void AreaOfCircleInput()
        {
            Console.WriteLine("Enter the radius of the circle: ");
            double areaOfCircleInput = double.Parse(Console.ReadLine());

            double areaOfCircleresult = AreaOfCircle(areaOfCircleInput);

            Console.WriteLine("The area of the circle is: " + areaOfCircleresult + " square units.");
            //menu();
        }

        //NUMBER CHECK RICHARD
        static string NumberCheck(string input) //changed input to string type, so it will accept the user input
        {

            int menuItem;
            do
            {
                bool numVer = int.TryParse(input, out menuItem);
                if ((menuItem != 0)) //when input can be converted to integer, try parse retuns 1 (or not 0)
                {
                    return input; //will return the same string value to Main()... where you assigned it to string x.. I added a bit to your Writeline(x)
                    //break; //break no longer needed. return always jumps out of method. This will even supercede the while statement below.
                }
                else if (menuItem == 0) //when input can not be converted to integer tryParse returns zero
                {
                    Console.WriteLine("That is not a valid entry, please enter a number");

                    input = Console.ReadLine(); //need this to give user a chance to re-enter a number (to escape the loop!)
                }
            }
            while (menuItem == 0); //this loop will repeat until user enters a string that CAN be converted to integer
            return input; //not sure why, but in a method, I think all code blocks have to have a return. Although your code will never reach this point, VS wants it here


        }
        static void NumberCheckInput()
        {
            Console.WriteLine("enter a number"); //moved this to Main() from NumberCheck().. this way the programmer can ask any questions and pass the input to your method below...
            string input = Console.ReadLine();

            string x = NumberCheck(input); //takes validated-as-integer return value from NumberCheck method and assigns to string x
            Console.WriteLine("Good. " + x + " is a valid number."); //prints string x
            //menu();
        }

        //VALID NAME MARGARET
        static string AllLetterChecker(string input)
        {
            string input2 = input;
            string whiteSpaceRemoved = Regex.Replace(input, @"\s+", ""); //remove all whitespace from input replace with empty string
            bool result = whiteSpaceRemoved.All(Char.IsLetter); //check whether each char is a letter

            if (input == "")
            {
                return "false";
            }
            else if (string.IsNullOrWhiteSpace(input))
            {
                return "false";
            }
            else if (result)
            {
                return "true"; //return string value "true" if all char are letters
            }
            else if (!result)
            {
                return "false"; //return string value "false" if all char are letters

            }
            return "";
        }
        static void AllLetterCheckerInput()
        {
            string input = null;
            do
            {
                Console.Write("First name: ");
                input = Console.ReadLine();
                //AllLetterChecker(input);
                if (AllLetterChecker(input) == "false") //false = input other than letters
                {
                    Console.WriteLine("Please use letters to enter your name.\n");
                }

            } while (AllLetterChecker(input) == "false");
            string firstName = input.Trim();

            do
            {
                Console.Write("Last name: ");
                input = Console.ReadLine();
                //AllLetterChecker(input);
                if (AllLetterChecker(input) == "false")
                {
                    Console.WriteLine("Please use only letters.\n");
                }

            } while (AllLetterChecker(input) == "false");
            string lastName = input.Trim();

            Console.WriteLine("Your name is {0} {1}.", firstName, lastName);

           // menu();
        }

        //NAME AGE STATE CAMERON
        static void NameAgeStateBuilder()
        {
            Console.Write("Name: ");
            string name = Console.ReadLine().Trim();

            Console.Write("\nAge: ");
            string age = Console.ReadLine().Trim();

            Console.Write("\nState: ");
            string state = Console.ReadLine().Trim();

            Console.WriteLine("Your name is {0}. You are {1} years old and live in {2}.", name, age, state);
        }

        //COUNTDOWN TIMER CADALE
        static void CountDown()
        {
            //Take user input
            Console.Write("Enter year to start in yyyy format (for example, 2016): ");
            Int32 year = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter starting month (for example 07 for July or 11 for November): ");
            Int32 month = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter starting day (For example, enter 8 for the 8th day of the month): ");
            Int32 day = Convert.ToInt32(Console.ReadLine());


            Console.Write("Enter starting hour (For example, enter 12 for 12:03): ");
            Int32 hour = Convert.ToInt32(Console.ReadLine());


            Console.Write("Enter starting minutes (For example, enter 03 for 12:03): ");
            Int32 minute = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter seconds to start: ");
            Int32 second = Convert.ToInt32(Console.ReadLine());

            //Take user input
            DateTime time = new DateTime(year, month, day, hour, minute, second); //user enters year, month, day, hours, minutes, seconds
            DateTime timeStarted = time;
            DateTime timeUp = new DateTime(2016, 2, 8, 12, 0, 0);

            Console.WriteLine("Press Any Key to Begin Countdown To Zero");
            Console.ReadKey();

            //execute timer
            do
            {

                Console.Clear();
                Console.WriteLine(timeStarted.ToString("COUNTDOWN BEGAN: " + "dddd hh:mm;ss\n"));
                Console.WriteLine(time.ToString("dddd hh:mm;ss"));
                //Console.WriteLine("Sleep for 1 seconds.");
                Thread.Sleep(1000);
                time = time.AddSeconds(-1);
                Console.Clear();
                Console.WriteLine(timeStarted);
                Console.WriteLine(time.ToString("dddd hh:mm;ss"));

            } while (!(time.Equals(timeUp)));

            Console.WriteLine("Countdown Done!");

        }

        //BUILD HOUSE MARY
        static void setConsoleSize()

        {
            System.Console.SetWindowPosition(0, 0);   // sets window position to upper left
            System.Console.SetBufferSize(200, 300);   // make sure buffer is bigger than window
            System.Console.SetWindowSize(150, 54);   //set window size to almost full screen
                                                     //width - maxSet(127,57)      (width, height)

            //System.Console.ResetColor(); //resets fore and background colors to  default

        }  // End  setConsoleSize()
        static double HouseBuilder(double input)
        {
            Console.Clear();

            setConsoleSize();

            double aDelta = input;
            int aDeltaStart = Convert.ToInt32(aDelta);
            double bDelta = aDelta - aDelta + 1;


            var positionA = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32(aDelta)));
            var positionB = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32(bDelta)));
            var top = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32(aDelta) + 1));


            //draw peak (first two lines)
            Console.WriteLine(top + "*");
            Console.WriteLine(positionA + "*" + positionB + "*");

            //loop to draw roof
            for (int i = 0; i < aDeltaStart; i++)
            {
                aDelta = aDelta - 1;
                bDelta = bDelta + 2;
                positionA = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32(aDelta)));
                positionB = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32(bDelta)));
                Console.WriteLine(positionA + "*" + positionB + "*");

                if (aDelta < 1) //finish roof when true
                {
                    for (int j = 0; j < aDeltaStart * 2 + 2; j++) //build base of roof
                    {
                        Console.Write(positionA + "*");
                    }

                    for (int k = 0; k < aDeltaStart; k++) //draw sides
                    {
                        positionA = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32(aDelta)));
                        positionB = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32(bDelta)));
                        Console.WriteLine(positionA + "*" + positionB + "*");
                        if (k == aDeltaStart - 1) //draw floor
                        {
                            for (int l = 0; l < aDeltaStart * 2 + 3; l++)
                            {
                                Console.Write(positionA + "*");
                            }
                        }
                        else
                        {
                            continue;
                        }
                    }
                }
                else
                {
                    continue;
                }
            }

            return input;
        }
        static void HouseBuilderInput()
        {
            Console.WriteLine("enter a number: ");
```

```csharp

            string input = Console.ReadLine();
            double inputConvert = Convert.ToDouble(input);

            HouseBuilder(inputConvert);

            //menu();
        }

        //LOST MY TEETH SIRAHN
        static void ToothCalculator()
        {
            //get user input

            Console.WriteLine("Welcome to the Tooth Fairy Calculator.\nLet's predict how many teeth you will have in the future!\n");
            Console.Write("Enter a future year to start in yyyy format (for example, 2016): \n");
            Int32 year = Convert.ToInt32(Console.ReadLine());
            Console.Clear();

            Console.Write("Enter future month (for example 07 for July or 11 for November): \n");
            Int32 month = Convert.ToInt32(Console.ReadLine());
            Console.Clear();

            Console.Write("Enter future day (For example, enter 8 for the 8th day of the month): \n");
            Int32 day = Convert.ToInt32(Console.ReadLine());
            Console.Clear();

            /////////

            DateTime futureTime = new DateTime(year, month, day, 0, 0, 0); //user enters year, month, day,

            DateTime now = DateTime.Now;

            TimeSpan elapsed = futureTime.Subtract(now);

            double daysThatWillHaveElapsed = elapsed.TotalDays;

            double numberOfTeethLost = daysThatWillHaveElapsed / 365;

            Int16 teethLostAsInteger = Convert.ToInt16(numberOfTeethLost);

            Console.WriteLine("You will have lost approximately " + teethLostAsInteger.ToString() + " teeth by that time.");




            //Console.WriteLine("Press Any Key to Begin Countdown To Zero");
            Console.ReadKey();

        }

        //SOBRIETY TEST JACOB
        static string SobrietyTest(string alpha)
        {
            StringBuilder reverse = new StringBuilder();
            for (int g = alpha.Length - 1; g >= 0; g--)
            {
                reverse.Append(alpha[g]);
            }
            return reverse.ToString();
        }
        static void SobrietyTestInput()
        {
            //SobrietyTest Method
            string alpha = "abcdefghijklmnopqrstuvwxyz";
            string sober = SobrietyTest(alpha);
            Console.WriteLine(sober);
            //menu();
        }

        //ALPHA SPLITTER IMARI
        static string SortAndNumberSentence(string str)
        {

            string noPunctuation = Regex.Replace(str, @"[^\w\s]", ""); //replace all that is not alphanumeric and not whitespace with empty string

            string noPunctuationNospace = noPunctuation.Trim(); //trims whatever whitespace may be at front and back of string
                                                                //Console.WriteLine(noPunctuationNospace +"nnnx"); //DEBUG

            string[] sortArray = noPunctuationNospace.Split(' '); //splits string based on whitespace, assigns each word to array index
            //foreach (string item in sortArray) //debug
            //{
            //    Console.WriteLine("x" + item); //DEBUG
            //}
            Array.Sort(sortArray); //sorts array into alphabetical order
            //Console.ReadKey();
            StringBuilder sb = new StringBuilder();
            int counter = 0;
            for (int i = 0; i < sortArray.Length; i++)
            {
                if (string.IsNullOrWhiteSpace(sortArray[i]))
                {
                    continue;
                }
                {

                    sb.Append((counter + 1) + ". " + sortArray[i] + " \n");
                    counter++;
                }
            }
            return sb.ToString();
        }
        static void SortAndNumberSentenceInput()
        {
            Console.WriteLine("Type a sentence. Any sentence.");
            string sentence = Console.ReadLine();
            string result = SortAndNumberSentence(sentence);
            Console.WriteLine(result);
            //menu();
        }

        //TOP STUDENT JENNIFER

        //FAMILY GUY LAWRENCE
        static string FamilyGuyLister(string input)
        {
            string[] inputArray = input.Split(',');
            string[] outputArray = new string[inputArray.Length];
            //int index = 0;
            //Console.WriteLine(inputArray[3].ToString().Trim());

            Console.WriteLine("\n\nFamily Guy Characters: \n");
            for (int i = 0; i < inputArray.Length; i++)
            {
                if (string.IsNullOrWhiteSpace(inputArray[i]))
                {
                    continue;
                }
                inputArray[i].ToString().Trim();
                Console.WriteLine(inputArray[i].ToString().Trim());
            }
            return inputArray.ToString();
        }

        static void FamilyGuyListerInput()
        {
            Console.WriteLine("Type the name of all Family Guy Characters separated by a comma: ");
            string input = Console.ReadLine();
            FamilyGuyLister(input);
        }

        //PRINT ARRAY KRISTA
        static void ArrayPrinter()
        {
            string[] input = new[] { "ham", "bacon", "cheese", "beer", "Happy
```

```
              Super Bowl" };
            for (int i = 0; i < input.Length; i++)
            {
                if (string.IsNullOrWhiteSpace(input[i]))
                {
                    continue;
                }
                input[i].ToString().Trim();
                Console.WriteLine(input[i].ToString().Trim());
                //menu();
            }


        }

        //PROPER NAME KIM
        static void ProperNameKim()
        {
            string firstName = "sally";
            string lastName = "bobally";
            string correctfirstName = null;
            string correctlastName = null;

            string[] inputArray = new string[] { firstName, lastName };

            for (int i = 0; i < inputArray.Length; i++)
            {
                if (string.IsNullOrEmpty(inputArray[i]))
                {
                    continue;
                }
                else if (inputArray[i].ToString().Equals(firstName))
                {
                    string result = inputArray[i].ToString();
                    char x = char.ToUpper(result[0]);
                    correctfirstName = x + firstName.Substring(1);
                }
            }

            for (int i = 0; i < inputArray.Length; i++)
            {
                if (string.IsNullOrEmpty(inputArray[i]))
                {
                    continue;
                }
                else if (inputArray[i].ToString().Equals(lastName))
                {
                    string result = inputArray[i].ToString();
                    char x = char.ToUpper(result[0]);
                    correctlastName = x + lastName.Substring(1);
                }
            }
            Console.WriteLine("{0} {1} went down to the alley. Bumped into her old girl Vicky from the valley", correctfirstName, correctlastName);

        }

        //USERNAME ASHLEY
        static void UserName()
        {
            Console.WriteLine("Enter your full name: ");
            string fullName = Console.ReadLine();

            Console.WriteLine("Enter your phone number: ");
            string phone = Console.ReadLine();

            string info = fullName + "," + phone;
            Console.WriteLine(info);

            StreamWriter writer = new StreamWriter("username.txt");
            using (writer)
            {
                writer.WriteLine(info);
            }

        }

        static void Main(string[] args)
        {




            //AreaOfCircleInput();
            //NumberCheckInput();
            //AllLetterCheckerInput();
            //NameAgeStateBuilder();
            //CountDown();
            //HouseBuilderInput();
            //ToothCalculator();
            //xxxxxxxxsobriety test??????
            //SortAndNumberSentenceInput();
            //SobrietyTestInput();
            //FamilyGuyListerInput();
            //ArrayPrinter();
            //ProperNameKim();
            //UserName();

            Console.ReadKey();
        }
    }
}

```