

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Text.RegularExpressions;
7 using System.Threading;
8 using System.IO;
9 namespace peerMethods__Childress_20160210
10 {
11     class Program
12     {
13
14         //AREA OF A CIRCLE QUINN
15         static double AreaOfCircle(double x)
16         {
17             double area = Math.PI * (x * x);
18             return area;
19         }
20         static void AreaOfCircleInput()
21         {
22             Console.WriteLine("Enter the radius of the circle: ");
23             double areaOfCircleInput = double.Parse(Console.ReadLine());
24
25             double areaOfCirccleresult = AreaOfCircle(areaOfCircleInput);
26
27             Console.WriteLine("The area of the circle is: " + areaOfCirccleresult +
28                 " square units.");
29             //menu();
30
31             //NUMBER CHECK RICHARD
32             static string NumberCheck(string input) //changed input to string type,
33             { //so it will accept the user input
34
35                 int menuItem;
36                 do
37                 {
38                     bool numVer = int.TryParse(input, out menuItem);
39                     if ((menuItem != 0)) //when input can be converted to integer,
40                     { //try parse returns 1 (or not 0)
41                         return input; //will return the same string value to Main
42                         (... where you assigned it to string x.. I added a bit to
43                         your Writeline(x)
44                         //break; //break no longer needed. return always jumps out of
45                         method. This will even supercede the while statement below.
46                     }
47                     else if (menuItem == 0) //when input can not be converted to
48                     { //integer tryParse returns zero
49
50                     }
```

```
46         Console.WriteLine("That is not a valid entry, please enter a
           number");
47
48         input = Console.ReadLine(); //need this to give user a chance
           to re-enter a number (to escape the loop!)
49     }
50 }
51 while (menuItem == 0); //this loop will repeat until user enters a
           string that CAN be converted to integer
52 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
53
54
55 }
56 static void NumberCheckInput()
57 {
58     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...
59     string input = Console.ReadLine();
60
61     string x = NumberCheck(input); //takes validated-as-integer return
           value from NumberCheck method and assigns to string x
62     Console.WriteLine("Good. " + x + " is a valid number."); //prints
           string x
63     //menu();
64 }
65
66 //VALID NAME MARGARET
67 static string AllLetterChecker(string input)
68 {
69     string input2 = input;
70     string whiteSpaceRemoved = Regex.Replace(input, @"\s+", ""); //remove
           all whitespace from input replace with empty string
71     bool result = whiteSpaceRemoved.All(Char.IsLetter); //check whether
           each char is a letter
72
73     if (input == "")
74     {
75         return "false";
76     }
77     else if (string.IsNullOrEmpty(input))
78     {
79         return "false";
80     }
81     else if (result)
82     {
83         return "true"; //return string value "true" if all char are
           letters
84     }
85     else if (!result)
```

```
86         {
87             return "false"; //return string value "false" if all char are letters
88
89         }
90         return "";
91     }
92     static void AllLetterCheckerInput()
93     {
94         string input = null;
95         do
96         {
97             Console.Write("First name: ");
98             input = Console.ReadLine();
99             //AllLetterChecker(input);
100             if (AllLetterChecker(input) == "false") //false = input other than letters
101             {
102                 Console.WriteLine("Please use letters to enter your name. \n");
103             }
104
105             } while (AllLetterChecker(input) == "false");
106             string firstName = input.Trim();
107
108             do
109             {
110                 Console.Write("Last name: ");
111                 input = Console.ReadLine();
112                 //AllLetterChecker(input);
113                 if (AllLetterChecker(input) == "false")
114                 {
115                     Console.WriteLine("Please use only letters.\n");
116                 }
117
118                 } while (AllLetterChecker(input) == "false");
119                 string lastName = input.Trim();
120
121                 Console.WriteLine("Your name is {0} {1}.", firstName, lastName);
122
123                 // menu();
124             }
125
126             //NAME AGE STATE CAMERON
127             static void NameAgeStateBuilder()
128             {
129                 Console.Write("Name: ");
130                 string name = Console.ReadLine().Trim();
131
132                 Console.Write("\nAge: ");
133                 string age = Console.ReadLine().Trim();
134
```

```
135         Console.WriteLine("\nState: ");
136         string state = Console.ReadLine().Trim();
137
138         Console.WriteLine("Your name is {0}. You are {1} years old and live  ↗
            in {2}.", name, age, state);
139     }
140
141     //COUNTDOWN TIMER CADALE
142     static void CountDown()
143     {
144         //Take user input
145         Console.WriteLine("Enter year to start in yyyy format (for example,  ↗
            2016): ");
146         Int32 year = Convert.ToInt32(Console.ReadLine());
147
148         Console.WriteLine("Enter starting month (for example 07 for July or 11  ↗
            for November): ");
149         Int32 month = Convert.ToInt32(Console.ReadLine());
150
151         Console.WriteLine("Enter starting day (For example, enter 8 for the 8th  ↗
            day of the month): ");
152         Int32 day = Convert.ToInt32(Console.ReadLine());
153
154
155         Console.WriteLine("Enter starting hour (For example, enter 12 for 12:03):  ↗
            ");
156         Int32 hour = Convert.ToInt32(Console.ReadLine());
157
158
159         Console.WriteLine("Enter starting minutes (For example, enter 03 for  ↗
            12:03): ");
160         Int32 minute = Convert.ToInt32(Console.ReadLine());
161
162         Console.WriteLine("Enter seconds to start: ");
163         Int32 second = Convert.ToInt32(Console.ReadLine());
164
165         //Take user input
166         DateTime time = new DateTime(year, month, day, hour, minute,  ↗
            second); //user enters year, month, day, hours, minutes, seconds
167         DateTime timeStarted = time;
168         DateTime timeUp = new DateTime(2016, 2, 8, 12, 0, 0);
169
170         Console.WriteLine("Press Any Key to Begin Countdown To Zero");
171         Console.ReadKey();
172
173         //execute timer
174         do
175         {
176
177             Console.Clear();
178             Console.WriteLine(timeStarted.ToString("COUNTDOWN BEGAN: " +  ↗
                "dddd hh:mm:ss\n"));
```

```

179     Console.WriteLine(time.ToString("dddd hh:mm:ss"));
180     //Console.WriteLine("Sleep for 1 seconds.");
181     Thread.Sleep(1000);
182     time = time.AddSeconds(-1);
183     Console.Clear();
184     Console.WriteLine(timeStarted);
185     Console.WriteLine(time.ToString("dddd hh:mm:ss"));
186
187     } while (!(time.Equals(timeUp)));
188
189     Console.WriteLine("Countdown Done!");
190
191 }
192
193 //BUILD HOUSE MARY
194 static void setConsoleSize()
195 {
196     System.Console.SetWindowPosition(0, 0);    // sets window position to ↗
197         upper left
198     System.Console.SetBufferSize(200, 300);    // make sure buffer is ↗
199         bigger than window
200     System.Console.SetWindowSize(150, 54);    //set window size to almost ↗
201         full screen
202         //width - maxSet(127,57) ↗
203         (width, height)
204
205     //System.Console.ResetColor(); //resets fore and background colors to ↗
206     default
207
208 } // End setConsoleSize()
209 static double HouseBuilder(double input)
210 {
211     Console.Clear();
212
213     setConsoleSize();
214
215     double aDelta = input;
216     int aDeltaStart = Convert.ToInt32(aDelta);
217     double bDelta = aDelta - aDelta + 1;
218
219     var positionA = String.Join("", Enumerable.Repeat(" ",
220         Convert.ToInt32(aDelta)));
221     var positionB = String.Join("", Enumerable.Repeat(" ",
222         Convert.ToInt32(bDelta)));
223     var top = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32
224         (aDelta) + 1));
225
226     //draw peak (first two lines)
227     Console.WriteLine(top + "*");

```

```

223     Console.WriteLine(positionA + "*" + positionB + "*");
224
225     //loop to draw roof
226     for (int i = 0; i < aDeltaStart; i++)
227     {
228         aDelta = aDelta - 1;
229         bDelta = bDelta + 2;
230         positionA = String.Join("", Enumerable.Repeat(" ",
231             Convert.ToInt32(aDelta)));
232         positionB = String.Join("", Enumerable.Repeat(" ",
233             Convert.ToInt32(bDelta)));
234         Console.WriteLine(positionA + "*" + positionB + "*");
235
236         if (aDelta < 1) //finish roof when true
237         {
238             for (int j = 0; j < aDeltaStart * 2 + 2; j++) //build base of
239                 roof
240             {
241                 Console.Write(positionA + "*");
242             }
243
244             for (int k = 0; k < aDeltaStart; k++) //draw sides
245             {
246                 positionA = String.Join("", Enumerable.Repeat(" ",
247                     Convert.ToInt32(aDelta)));
248                 positionB = String.Join("", Enumerable.Repeat(" ",
249                     Convert.ToInt32(bDelta)));
250                 Console.WriteLine(positionA + "*" + positionB + "*");
251                 if (k == aDeltaStart - 1) //draw floor
252                 {
253                     for (int l = 0; l < aDeltaStart * 2 + 3; l++)
254                     {
255                         Console.Write(positionA + "*");
256                     }
257                 }
258                 else
259                 {
260                     continue;
261                 }
262             }
263         }
264     }
265     return input;
266 }
267 static void HouseBuilderInput()
268 {
269     Console.WriteLine("enter a number: ");

```

```
270
271     string input = Console.ReadLine();
272     double inputConvert = Convert.ToDouble(input);
273
274     HouseBuilder(inputConvert);
275
276     //menu();
277 }
278
279 //LOST MY TEETH SIRAHN
280 static void ToothCalculator()
281 {
282     //get user input
283
284     Console.WriteLine("Welcome to the Tooth Fairy Calculator.\nLet's      ↗
285         predict how many teeth you will have in the future!\n");
286     Console.Write("Enter a future year to start in yyyy format (for      ↗
287         example, 2016): \n");
288     Int32 year = Convert.ToInt32(Console.ReadLine());
289     Console.Clear();
290
291     Console.Write("Enter future month (for example 07 for July or 11 for      ↗
292         November): \n");
293     Int32 month = Convert.ToInt32(Console.ReadLine());
294     Console.Clear();
295
296     Console.Write("Enter future day (For example, enter 8 for the 8th day      ↗
297         of the month): \n");
298     Int32 day = Convert.ToInt32(Console.ReadLine());
299     Console.Clear();
300
301     //////////
302
303     DateTime futureTime = new DateTime(year, month, day, 0, 0, 0); //user ↗
304         enters year, month, day,
305
306     DateTime now = DateTime.Now;
307
308     TimeSpan elapsed = futureTime.Subtract(now);
309
310     double daysThatWillHaveElapsed = elapsed.TotalDays;
311
312     double numberOfTeethLost = daysThatWillHaveElapsed / 365;
313
314     Int16 teethLostAsInteger = Convert.ToInt16(numberOfTeethLost);
315
316     Console.WriteLine("You will have lost approximately " +      ↗
317         teethLostAsInteger.ToString() + " teeth by that time.");
```

```
316         //Console.WriteLine("Press Any Key to Begin Countdown To Zero");
317         Console.ReadKey();
318     }
319 }
320
321 //SOBRIETY TEST JACOB
322 static string SobrietyTest(string alpha)
323 {
324     StringBuilder reverse = new StringBuilder();
325     for (int g = alpha.Length - 1; g >= 0; g--)
326     {
327         reverse.Append(alpha[g]);
328     }
329     return reverse.ToString();
330 }
331 static void SobrietyTestInput()
332 {
333     //SobrietyTest Method
334     string alpha = "abcdefghijklmnopqrstuvwxyz";
335     string sober = SobrietyTest(alpha);
336     Console.WriteLine(sober);
337     //menu();
338 }
339
340 //ALPHA SPLITTER IMARI
341 static string SortAndNumberSentence(string str)
342 {
343
344     string noPunctuation = Regex.Replace(str, @"[^w\s]", ""); //replace
345     //all that is not alphanumeric and not whitespace with empty string
346
347     string noPunctuationNospace = noPunctuation.Trim(); //trims whatever
348     //whitespace may be at front and back of string
349
350     Console.WriteLine(noPunctuationNospace + "nnnx"); //DEBUG
351
352     string[] sortArray = noPunctuationNospace.Split(' '); //splits string
353     //based on whitespace, assigns each word to array index
354     //foreach (string item in sortArray) //debug
355     //{
356     //    Console.WriteLine("x" + item); //DEBUG
357     //}
358     Array.Sort(sortArray); //sorts array into alphabetical order
359     Console.ReadKey();
360     StringBuilder sb = new StringBuilder();
361     int counter = 0;
362     for (int i = 0; i < sortArray.Length; i++)
363     {
364         if (string.IsNullOrEmpty(sortArray[i]))
365         {
366             continue;
367         }
368     }
369 }
```



```
364         {
365             sb.Append((counter + 1) + ". " + sortArray[i] + " \n");
366             counter++;
367         }
368     }
369     return sb.ToString();
370 }
371 static void SortAndNumberSentenceInput()
372 {
373     Console.WriteLine("Type a sentence. Any sentence.");
374     string sentence = Console.ReadLine();
375     string result = SortAndNumberSentence(sentence);
376     Console.WriteLine(result);
377     //menu();
378 }
379
380 //TOP STUDENT JENNIFER
381
382 //FAMILY GUY LAWRENCE
383 static string FamilyGuyLister(string input)
384 {
385     string[] inputArray = input.Split(',');
386     string[] outputArray = new string[inputArray.Length];
387     //int index = 0;
388     //Console.WriteLine(inputArray[3].ToString().Trim());
389
390     Console.WriteLine("\n\nFamily Guy Characters: \n");
391     for (int i = 0; i < inputArray.Length; i++)
392     {
393         if (string.IsNullOrEmpty(inputArray[i]))
394         {
395             continue;
396         }
397         inputArray[i].ToString().Trim();
398         Console.WriteLine(inputArray[i].ToString().Trim());
399     }
400     return inputArray.ToString();
401 }
402
403 static void FamilyGuyListerInput()
404 {
405     Console.WriteLine("Type the name of all Family Guy Characters
406         separated by a comma: ");
407     string input = Console.ReadLine();
408     FamilyGuyLister(input);
409 }
410
411 //PRINT ARRAY KRISTA
412 static void ArrayPrinter()
413 {
414     string[] input = new[] { "ham", "bacon", "cheese", "beer", "Happy" }
```

```
        Super Bowl" };
415     for (int i = 0; i < input.Length; i++)
416     {
417         if (string.IsNullOrEmpty(input[i]))
418         {
419             continue;
420         }
421         input[i].ToString().Trim();
422         Console.WriteLine(input[i].ToString().Trim());
423         //menu();
424     }
425
426
427 }
428
429 //PROPER NAME KIM
430 static void ProperNameKim()
431 {
432     string firstName = "sally";
433     string lastName = "bobally";
434     string correctfirstName = null;
435     string correctlastName = null;
436
437     string[] inputArray = new string[] { firstName, lastName };
438
439     for (int i = 0; i < inputArray.Length; i++)
440     {
441         if (string.IsNullOrEmpty(inputArray[i]))
442         {
443             continue;
444         }
445         else if (inputArray[i].ToString().Equals(firstName))
446         {
447             string result = inputArray[i].ToString();
448             char x = char.ToUpper(result[0]);
449             correctfirstName = x + firstName.Substring(1);
450         }
451     }
452
453     for (int i = 0; i < inputArray.Length; i++)
454     {
455         if (string.IsNullOrEmpty(inputArray[i]))
456         {
457             continue;
458         }
459         else if (inputArray[i].ToString().Equals(lastName))
460         {
461             string result = inputArray[i].ToString();
462             char x = char.ToUpper(result[0]);
463             correctlastName = x + lastName.Substring(1);
464         }
465     }
```

```
466         Console.WriteLine("{0} {1} went down to the alley. Bumped into her  
         old girl Vicky from the valley", correctfirstName,  
         correctlastName);  
  
467  
468     }  
469  
470     //USERNAME ASHLEY  
471     static void UserName()  
472     {  
473         Console.WriteLine("Enter your full name: ");  
474         string fullName = Console.ReadLine();  
475  
476         Console.WriteLine("Enter your phone number: ");  
477         string phone = Console.ReadLine();  
478  
479         string info = fullName + "," + phone;  
480         Console.WriteLine(info);  
481  
482         StreamWriter writer = new StreamWriter("username.txt");  
483         using (writer)  
484         {  
485             writer.WriteLine(info);  
486         }  
487  
488     }  
489  
490     static void Main(string[] args)  
491     {  
492  
493  
494  
495  
496         //AreaOfCircleInput();  
497         //NumberCheckInput();  
498         //AllLetterCheckerInput();  
499         //NameAgeStateBuilder();  
500         //CountDown();  
501         //HouseBuilderInput();  
502         //ToothCalculator();  
503         //xxxxxxxxsobriety test?????  
504         //SortAndNumberSentenceInput();  
505         //SobrietyTestInput();  
506         //FamilyGuyListerInput();  
507         //ArrayPrinter();  
508         //ProperNameKim();  
509         //UserName();  
510  
511         Console.ReadKey();  
512     }  
513 }  
514 }  
515
```