```
1 using System;
 2 using System.Collections.Generic;
 3 using System.Ling;
 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
                double area = Math.PI * (x * x);
17
18
                return area;
19
20
           static void AreaOfCircleInput()
21
                Console.WriteLine("Enter the radius of the circle: ");
22
23
                double areaOfCircleInput = double.Parse(Console.ReadLine());
24
                double areaOfCircleresult = AreaOfCircle(areaOfCircleInput);
25
26
27
                Console.WriteLine("The area of the circle is: " + areaOfCircleresult →
                  + " square units.");
28
                //menu();
29
           }
30
           //NUMBER CHECK RICHARD
31
           static string NumberCheck(string input) //changed input to string type,
32
              so it will accept the user input
33
           {
34
35
                int menuItem;
36
                do
37
                {
                    bool numVer = int.TryParse(input, out menuItem);
38
                    if ((menuItem != 0)) //when input can be converted to integer,
39
                      try parse retuns 1 (or not 0)
40
                    {
41
                        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)
42
                        //break; //break no longer needed. return always jumps out of ₹
                         method. This will even supercede the while statement below.
43
                    else if (menuItem == 0) //when input can not be converted to
44
                                                                                       P
                      integer tryParse returns zero
45
```

```
\dots0160210-master\peerMethods -Childress-20160210\Program.cs
```

```
Console.WriteLine("That is not a valid entry, please enter a
46
                         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
                                                                                        P
                  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
                string x = NumberCheck(input); //takes validated-as-integer return
61
                  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
            //VALID NAME MARGARET
66
            static string AllLetterChecker(string input)
67
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
                if (input == "")
73
74
                {
                    return "false";
75
76
77
                else if (string.IsNullOrWhiteSpace(input))
78
79
                    return "false";
80
                }
81
                else if (result)
82
83
                    return "true"; //return string value "true" if all char are
                      letters
84
                else if (!result)
85
```

```
\dots0160210-master\peerMethods -Childress-20160210\Program.cs
                                                                                          3
 86
                     return "false"; //return string value "false" if all char are
 87
                                                                                         P
 88
 89
 90
                 return "";
 91
             }
 92
             static void AllLetterCheckerInput()
 93
 94
                 string input = null;
 95
                 do
 96
                 {
                     Console.Write("First name: ");
 97
 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
                 } while (AllLetterChecker(input) == "false");
105
106
                 string firstName = input.Trim();
107
108
                 do
109
                 {
                     Console.Write("Last name: ");
110
111
                     input = Console.ReadLine();
                     //AllLetterChecker(input);
112
113
                     if (AllLetterChecker(input) == "false")
114
                     {
                         Console.WriteLine("Please use only letters.\n");
115
116
                     }
117
                 } while (AllLetterChecker(input) == "false");
118
119
                 string lastName = input.Trim();
120
                 Console.WriteLine("Your name is {0} {1}.", firstName, lastName);
121
122
123
                // menu();
124
             }
125
             //NAME AGE STATE CAMERON
126
             static void NameAgeStateBuilder()
127
128
             {
129
                 Console.Write("Name: ");
                 string name = Console.ReadLine().Trim();
130
131
                 Console.Write("\nAge: ");
132
                 string age = Console.ReadLine().Trim();
133
```

134

```
...0160210-master\peerMethods -Childress-20160210\Program.cs
                                                                                        4
135
                 Console.Write("\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
             //COUNTDOWN TIMER CADALE
141
142
             static void CountDown()
143
144
                 //Take user input
                 Console.Write("Enter year to start in yyyy format (for example,
145
                   2016): ");
146
                 Int32 year = Convert.ToInt32(Console.ReadLine());
147
148
                 Console.Write("Enter starting month (for example 07 for July or 11
                   for November): ");
149
                 Int32 month = Convert.ToInt32(Console.ReadLine());
150
                 Console.Write("Enter starting day (For example, enter 8 for the 8th
151
                   day of the month): ");
152
                 Int32 day = Convert.ToInt32(Console.ReadLine());
153
154
155
                 Console.Write("Enter starting hour (For example, enter 12 for 12:03): →
156
                 Int32 hour = Convert.ToInt32(Console.ReadLine());
157
158
                 Console.Write("Enter starting minutes (For example, enter 03 for
159
                   12:03): ");
                 Int32 minute = Convert.ToInt32(Console.ReadLine());
160
161
162
                 Console.Write("Enter seconds to start: ");
163
                 Int32 second = Convert.ToInt32(Console.ReadLine());
164
165
                 //Take user input
                 DateTime time = new DateTime(year, month, day, hour, minute,
166
                   second); //user enters year, month, day, hours, minutes, seconds
167
                 DateTime timeStarted = time;
                 DateTime timeUp = new DateTime(2016, 2, 8, 12, 0, 0);
168
169
                 Console.WriteLine("Press Any Key to Begin Countdown To Zero");
170
171
                 Console.ReadKey();
172
173
                 //execute timer
174
                 do
175
                 {
176
                     Console.Clear();
177
                     Console.WriteLine(timeStarted.ToString("COUNTDOWN BEGAN: " +
178
                       "dddd hh:mm;ss\n"));
```

```
...0160210-master\peerMethods -Childress-20160210\Program.cs
                                                                                         5
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
                 Console.WriteLine("Countdown Done!");
189
190
191
             }
192
193
             //BUILD HOUSE MARY
194
             static void setConsoleSize()
195
             {
196
197
                 System.Console.SetWindowPosition(0, 0); // sets window position to →
                   upper left
198
                 System.Console.SetBufferSize(200, 300); // make sure buffer is
                                                                                        P
                   bigger than window
                 System.Console.SetWindowSize(150, 54);
                                                          //set window size to almost →
199
                   full screen
200
                                                           //width - maxSet(127,57)
                          (width, height)
201
                 //System.Console.ResetColor(); //resets fore and background colors to ₹
202
                    default
203
             } // End setConsoleSize()
204
             static double HouseBuilder(double input)
205
206
207
                 Console.Clear();
208
209
                 setConsoleSize();
210
211
                 double aDelta = input;
```

```
212
                 int aDeltaStart = Convert.ToInt32(aDelta);
213
                 double bDelta = aDelta - aDelta + 1;
214
215
                 var positionA = String.Join("", Enumerable.Repeat(" ",
216
                   Convert.ToInt32(aDelta)));
                 var positionB = String.Join("", Enumerable.Repeat(" ",
217
                   Convert.ToInt32(bDelta)));
                 var top = String.Join("", Enumerable.Repeat(" ", Convert.ToInt32
218
                   (aDelta) + 1));
219
220
221
                 //draw peak (first two lines)
                 Console.WriteLine(top + "*");
222
```

```
...0160210-master\peerMethods -Childress-20160210\Program.cs
```

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

```
...0160210-master\peerMethods -Childress-20160210\Program.cs
```

```
7
```

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

```
...0160210-master\peerMethods -Childress-20160210\Program.cs
```

```
8
```

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

```
...0160210-master\peerMethods -Childress-20160210\Program.cs
                                                                                          9
364
365
                          sb.Append((counter + 1) + ". " + sortArray[i] + " \n");
366
367
                         counter++;
368
                     }
369
                 }
370
                 return sb.ToString();
371
             }
372
             static void SortAndNumberSentenceInput()
373
                 Console.WriteLine("Type a sentence. Any sentence.");
374
375
                 string sentence = Console.ReadLine();
376
                 string result = SortAndNumberSentence(sentence);
377
                 Console.WriteLine(result);
378
                 //menu();
379
             }
380
381
             //TOP STUDENT JENNIFER
382
383
             //FAMILY GUY LAWRENCE
384
             static string FamilyGuyLister(string input)
385
386
                 string[] inputArray = input.Split(',');
387
                 string[] outputArray = new string[inputArray.Length];
388
                 //int index = 0;
389
                 //Console.WriteLine(inputArray[3].ToString().Trim());
390
                 Console.WriteLine("\n\nFamily Guy Characters: \n");
391
392
                 for (int i = 0; i < inputArray.Length; i++)</pre>
393
                     if (string.IsNullOrWhiteSpace(inputArray[i]))
394
395
                     {
396
                         continue;
397
                     inputArray[i].ToString().Trim();
398
399
                     Console.WriteLine(inputArray[i].ToString().Trim());
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();
                 FamilyGuyLister(input);
408
409
             }
410
411
             //PRINT ARRAY KRISTA
```

string[] input = new[] { "ham", "bacon", "cheese", "beer", "Happy

static void ArrayPrinter()

412413

414

{

```
Super Bowl" };
415
                 for (int i = 0; i < input.Length; i++)</pre>
416
417
                      if (string.IsNullOrWhiteSpace(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
             {
                 string firstName = "sally";
432
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++)</pre>
440
                 {
441
                      if (string.IsNullOrEmpty(inputArray[i]))
442
                      {
443
                          continue;
444
                      else if (inputArray[i].ToString().Equals(firstName))
445
446
447
                          string result = inputArray[i].ToString();
                          char x = char.ToUpper(result[0]);
448
449
                          correctfirstName = x + firstName.Substring(1);
                     }
450
                 }
451
452
                 for (int i = 0; i < inputArray.Length; i++)</pre>
453
454
                      if (string.IsNullOrEmpty(inputArray[i]))
455
456
                     {
457
                          continue;
458
                      }
                     else if (inputArray[i].ToString().Equals(lastName))
459
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
                                                                                          P
                   old girl Vicky from the valley", correctfirstName,
                                                                                          P
                   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
                 //AreaOfCircleInput();
496
497
                 //NumberCheckInput();
498
                 //AllLetterCheckerInput();
499
                 //NameAgeStateBuilder();
500
                 //CountDown();
501
                 //HouseBuilderInput();
502
                 //ToothCalculator();
                 //xxxxxxxxsobriety test??????
503
504
                 //SortAndNumberSentenceInput();
                 //SobrietyTestInput();
505
                 //FamilyGuyListerInput();
506
507
                 //ArrayPrinter();
508
                 //ProperNameKim();
                 //UserName();
509
510
511
                 Console.ReadKey();
512
             }
513
         }
514 }
515
```