

DEAKIN UNIVERSITY

OBJECT ORIENTED DEVELOPMENT

ONTRACK SUBMISSION

C# Essentials: Arrays and Lists

Submitted By:

Peter STACEY

pstacey

2020/03/26 08:29

Tutor:

Dipto PRATYAKSA

Outcome	Weight
Evaluate Code	◆◆◆◆◆
Principles	◆◆◆◆◆
Build Programs	◆◆◆◆◆
Design	◆◆◆◆◆
Justify	◆◆◆◆◆

This task has a large number of subtasks all related to arrays and lists, with several problems to work through, with their use. This includes a number of pieces of code to evaluate and also to write up using correct conventions. Additionally, there are multiple programs to solve and code, resulting in a significant amount of programming to be completed. The task contains designed pieces of code and my video has further diagrams and descriptions. Between the submitted csharp files and video, evidence is provided against each of the criteria of the task.

March 26, 2020



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4
5  namespace Task_3._1P
6  {
7      class Program
8      {
9          static void Main(string[] args)
10         {
11             // *****
12             // PART 1: STEP 1
13             // *****
14             // declares an array of type double with 10 elements
15             double[] myArray = new double[10];
16
17             // assigning the first element of the array
18             myArray[0] = 1.0;
19
20             // assigning the second element of the array
21             myArray[1] = 1.1;
22
23             // assigning the third element of the array
24             myArray[2] = 1.2;
25
26             // assigning the fourth element of the array
27             myArray[3] = 1.3;
28
29             // assigning the fifth element of the array
30             myArray[4] = 1.4;
31
32             // assigning the sixth element of the array
33             myArray[5] = 1.5;
34
35             // assigning the seventh element of the array
36             myArray[6] = 1.6;
37
38             // assigning the eighth element of the array
39             myArray[7] = 1.7;
40
41             // assigning the ninth element of the array
42             myArray[8] = 1.8;
43
44             // assigning the tenth element of the array
45             myArray[9] = 1.9;
46
47             // *****
48             // PART 1: STEP 2
49             // *****
50             Console.WriteLine("The element at index 0 in the array is " +
51                               ↪ myArray[0]);
51             Console.WriteLine("The element at index 1 in the array is " +
52                               ↪ myArray[1]);
```

```
52     Console.WriteLine("The element at index 2 in the array is " +
53         ↪ myArray[2]);
54     Console.WriteLine("The element at index 3 in the array is " +
55         ↪ myArray[3]);
56     Console.WriteLine("The element at index 4 in the array is " +
57         ↪ myArray[4]);
58     Console.WriteLine("The element at index 5 in the array is " +
59         ↪ myArray[5]);
60     Console.WriteLine("The element at index 6 in the array is " +
61         ↪ myArray[6]);
62     Console.WriteLine("The element at index 7 in the array is " +
63         ↪ myArray[7]);
64     Console.WriteLine("The element at index 8 in the array is " +
65         ↪ myArray[8]);
66     Console.WriteLine("The element at index 9 in the array is " +
67         ↪ myArray[9]);
68
69     // *****
70     // PART 2: STEP 1
71     // *****
72     int[] myIntArray = new int[10];
73
74     for (int i = 0; i < myIntArray.Length; i++)
75     {
76         myIntArray[i] = i;
77     }
78
79     // *****
80     // PART 2: STEP 2
81     // *****
82     for (int i = 0; i < myIntArray.Length; i++)
83     {
84         Console.WriteLine("The element at position {0} is {1}",
85             i, myIntArray[i]);
86     }
87
88     // *****
89     // PART 3
90     // *****
91     int[] studentArray = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };
92     int total = 0;
93
94     for (int i = 0; i < studentArray.Length; i++)
95     {
96         total += studentArray[i];
97     }
98
99     Console.WriteLine("The total marks for the studen is " + total);
100    Console.WriteLine("This consists of " + studentArray.Length + " marks");
101    Console.WriteLine("Therefore the average mark is "
102        + (total / studentArray.Length));
103
104    // *****
```

```
97      // PART 4
98      // *****
99      String[] studentNames = new String[6];
100
101      for (int i = 0; i < studentNames.Length; i++)
102      {
103          Console.Write("Student {0} name: ", i + 1);
104          studentNames[i] = Console.ReadLine();
105      }
106
107      for (int i = 0; i < studentNames.Length; i++)
108      {
109          Console.WriteLine("Student {0}: {1}", i + 1, studentNames[i]);
110      }
111
112      // *****
113      // PART 5
114      // *****
115      double[] values = new double[10];
116      double currentLargest, currentSmallest;
117
118      for (int i = 0; i < values.Length; i++)
119      {
120          Console.Write("Enter a double for position {0}: ", i);
121          String input = Console.ReadLine();
122          // Note - no error checking. Expects valid input only
123          values[i] = Convert.ToDouble(input);
124      }
125
126      currentLargest = values[0];
127
128      for (int i = 0; i < values.Length; i++)
129      {
130          if (values[i] > currentLargest)
131              currentLargest = values[i];
132          Console.WriteLine(values[i]);
133      }
134
135      Console.WriteLine("The largest value is " + currentLargest);
136
137      currentSmallest = values[0];
138
139      for (int i = 0; i < values.Length; i++)
140      {
141          if (values[i] < currentSmallest)
142              currentSmallest = values[i];
143      }
144
145      Console.WriteLine("The smallest value is " + currentSmallest);
146
147      // *****
148      // PART 6
149      // *****
```

```
150     int[,] myMultiArray = new int[3, 4] { { 1, 2, 3, 4 }, { 1, 1, 1, 1 }, {  
    ↪ 2, 2, 2, 2 } };  
  
151  
152     for (int i = 0; i < myMultiArray.GetLength(0); i++)  
153     {  
154         for (int j = 0; j < myMultiArray.GetLength(1); j++)  
155         {  
156             Console.Write(myMultiArray[i, j] + "\t");  
157         }  
158         Console.WriteLine();  
159     }  
  
160  
161     List<String> myStudentList = new List<string>();  
162  
163     Random randomValue = new Random();  
164     int randomNumber = randomValue.Next(1, 12);  
165  
166     Console.WriteLine("You now need to add all " + randomNumber  
167         + " students to your class list");  
168  
169     for (int i = 0; i < randomNumber; i++)  
170     {  
171         Console.Write("Please enter the name of Student " + (i + 1) + ": ");  
172         myStudentList.Add(Console.ReadLine());  
173         Console.WriteLine();  
174     }  
175  
176     // *****  
177     // PART 7  
178     // *****  
179     int FuncOne(int[] values)  
180     {  
181         if (values.Length <= 10)  
182         {  
183             return GetOddProduct(values);  
184         }  
185         else  
186         {  
187             return NumberOfEvens(values);  
188         }  
189     }  
190  
191     int GetOddProduct(int[] values)  
192     {  
193         int oddProduct = 1;  
194         for (int i = 0; i < values.Length; i++)  
195         {  
196             if (values[i] % 2 == 1)  
197                 oddProduct *= values[i];  
198         }  
199         return oddProduct;  
200     }  
201
```

```
202     int NumberOfEvens(int[] values)
203     {
204         int numberOfEvens = 0;
205         for (int i = 0; i < values.Length; i++)
206         {
207             if (values[i] % 2 == 0)
208                 numberOfEvens++;
209         }
210         return numberOfEvens;
211     }
212
213
214     int[] numArray = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
215
216     // Expect 9 elements, odd product = 1*3*5*7*9=945
217     Console.WriteLine("Result from FuncOne for array of {0} elements: {1}",
218         numArray.Length, FuncOne(numArray));
219
220     int[] numArray2 = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
221
222     // Expect 12 elements, number of evens = 2,4,6,8,10,12 = 6
223     Console.WriteLine("Result from FuncOne for array of {0} elements: {1}",
224         numArray2.Length, FuncOne(numArray2));
225
226     // *****
227     // PART 8
228     // *****
229     void FuncTwo(List<double> values) // list is passed in by reference,
230         ↪ so we can modify it
231     {
232         double sum = values.Sum();
233         double average = sum / values.Count;
234         for (int i = 0; i < values.Count; i++)
235         {
236             values[i] -= average;
237         }
238         return;
239     }
240
241     List<double> myList = new List<double>() { 1.0, 2.0, 3.0, 4.0, 5.0 };
242
243     for (int i = 0; i < myList.Count; i++)
244     {
245         Console.Write(myList[i] + "\t");
246     }
247     Console.WriteLine();
248
249     FuncTwo(myList);
250
251     for (int i = 0; i < myList.Count; i++)
252     {
253         Console.Write(myList[i] + "\t");
254     }
```

```
254 Console.WriteLine();
255
256 // *****
257 // PART 9
258 // *****
259 int[] FuncThree(int[,] values)
260 {
261     List<int> result = new List<int>();
262     for (int j = 0; j < values.GetLength(1); j++)
263     {
264         for (int i = 0; i < values.GetLength(0); i++)
265         {
266             if (values[i, j] % 3 == 0)
267                 result.Add(values[i, j]);
268         }
269     }
270     return result.ToArray();
271 }
272
273 int[,] myMulti = { { 1, 2, 3, 4, 5 },
274                   { 6, 7, 8, 9, 10 },
275                   { 11, 12, 13, 14, 15 } };
276
277 int[] mySingle = FuncThree(myMulti);
278
279 for (int i = 0; i < mySingle.Length; i++)
280 {
281     // Expected result 6, 12, 3, 9, 15
282     Console.WriteLine(mySingle[i]);
283 }
284
285 // *****
286 // PART 10
287 // *****
288 int[,] FuncFour(int[] values)
289 {
290     int[,] result = new int[values.Length, 10];
291     for (int i = 0; i < values.Length; i++)
292     {
293         for (int j = 1; j <= 10; j++)
294         {
295             result[i, j - 1] = values[i] * j;
296         }
297     }
298     return result;
299 }
300
301 int[] intArray = { 1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 10 };
302 int[,] result = FuncFour(intArray);
303
304 Console.WriteLine(" \t|1\t2\t3\t4\t5\t6\t7\t8\t9\t10");
305 Console.WriteLine("-----")
    ↪ -----");
```

```
306         for (int i = 0; i < result.GetLength(0); i++)
307         {
308             Console.Write(intArray[i] + "\t");
309             for (int j = 0; j < 10; j++)
310             {
311                 Console.Write(result[i, j] + "\t");
312             }
313             Console.WriteLine();
314         }
315         Console.WriteLine();
316     }
317 }
318 }
```