

DEAKIN UNIVERSITY

OBJECT ORIENTED DEVELOPMENT

ONTRACK SUBMISSION

C# Essentials: Selection and Casting

Submitted By:

Peter STACEY

pstacey

2020/03/18 08:08

Tutor:

Dipto PRATYAKSA

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

The task involves evaluating provided code that contain errors, to identify problems and then correct them before coding. Additionally, the task involves working in an object oriented way through the use of C# and there are multiple problems to be solved, each with an individual program. While there is some design, all of the problems can be solved in single file programs and with simple design.

March 18, 2020



```
1  using System;
2
3  namespace Task_1._1P
4  {
5      class IfStatement
6      {
7          static void Main(string[] args)
8          {
9              int number = 0;
10
11              Console.WriteLine("Enter the number (as an integer): ");
12              try
13              {
14                  number = Convert.ToInt32(Console.ReadLine());
15              }
16              catch (FormatException) // Thrown for non-integer input
17              {
18                  Console.WriteLine("ERROR: Input was not a number");
19                  System.Environment.Exit(1);
20              }
21
22              if (number == 1)
23              {
24                  Console.WriteLine("ONE");
25              }
26              else if (number == 2)
27              {
28                  Console.WriteLine("TWO");
29              }
30              else if (number == 3)
31              {
32                  Console.WriteLine("THREE");
33              }
34              else if (number == 4)
35              {
36                  Console.WriteLine("FOUR");
37              }
38              else if (number == 5)
39              {
40                  Console.WriteLine("FIVE");
41              }
42              else if (number == 6)
43              {
44                  Console.WriteLine("SIX");
45              }
46              else if (number == 7)
47              {
48                  Console.WriteLine("SEVEN");
49              }
50              else if (number == 8)
51              {
52                  Console.WriteLine("EIGHT");
53              }
```

```
54         else if (number == 9)
55         {
56             Console.WriteLine("NINE");
57         }
58         else
59         {
60             Console.WriteLine("ERROR: Number must be from 1-9");
61         }
62     }
63 }
64 }
```

```
1  using System;
2
3  namespace Program_2
4  {
5      class SwitchStatement
6      {
7          static void Main(string[] args)
8          {
9              int number = 0;
10
11              Console.WriteLine("Enter a number (as an integer): ");
12              try
13              {
14                  number = Convert.ToInt32(Console.ReadLine());
15              }
16              catch (FormatException)
17              {
18                  Console.WriteLine("ERROR: Input was not an integer");
19                  System.Environment.Exit(1);
20              }
21              switch (number)
22              {
23                  case 1: Console.WriteLine("One"); break;
24                  case 2: Console.WriteLine("Two"); break;
25                  case 3: Console.WriteLine("Three"); break;
26                  case 4: Console.WriteLine("Four"); break;
27                  case 5: Console.WriteLine("Five"); break;
28                  case 6: Console.WriteLine("Six"); break;
29                  case 7: Console.WriteLine("Seven"); break;
30                  case 8: Console.WriteLine("Eight"); break;
31                  case 9: Console.WriteLine("Nine"); break;
32                  default: Console.WriteLine("ERROR: Number must be from 1-9"); break;
33              }
34
35              Console.ReadLine();
36          }
37      }
38  }
```

```
1  using System;
2
3  namespace Program_5
4  {
5      /// <summary>
6      /// Calculates recommended cooking time in a microwave
7      /// </summary>
8      class Microwave
9      {
10         // Reads String input in the console
11         /// <summary>
12         /// Reads String input in the console
13         /// </summary>
14         /// <returns>
15         /// The String input of the user
16         /// </returns>
17         /// <param name="prompt">The String prompt for the user</param>
18         public String ReadString(String prompt)
19         {
20             Console.Write(prompt + ": ");
21             return Console.ReadLine();
22         }
23
24         // Reads integer input in the console
25         /// <summary>
26         /// Reads integer input in the console
27         /// </summary>
28         /// <returns>
29         /// The input of the user as an integer
30         /// </returns>
31         /// <param name="prompt">The String prompt for the user</param>
32         public int ReadInteger(String prompt)
33         {
34             int number = 0;
35             String numberInput = ReadString(prompt);
36             while (!(int.TryParse(numberInput, out number)))
37             {
38                 Console.WriteLine("Please enter a whole number");
39                 numberInput = ReadString(prompt);
40             }
41             return Convert.ToInt32(numberInput);
42         }
43
44         // Reads double input in the console
45         /// <summary>
46         /// Reads double input in the console
47         /// </summary>
48         /// <returns>
49         /// The input of the user as a double
50         /// </returns>
51         /// <param name="prompt">The String prompt for the user</param>
52         public double ReadDouble(String prompt)
53         {
```

```
54         double number = 0.0;
55         String numberInput = ReadString(prompt);
56         while (!(double.TryParse(numberInput, out number)))
57         {
58             Console.WriteLine("Please enter a number");
59             numberInput = ReadString(prompt);
60         }
61         return Convert.ToDouble(numberInput);
62     }
63
64     // Returns the number of items to cook
65     /// <summary>
66     /// Returns the number of items to cook
67     /// </summary>
68     /// <returns>
69     /// An integer of the number of items
70     /// </returns>
71     public int NumberOfItems()
72     {
73         String prompt = "Enter the number of items";
74         int items = ReadInteger(prompt);
75         while (items < 1)
76         {
77             Console.WriteLine("Please enter at least 1 item");
78             items = ReadInteger(prompt);
79         }
80         return items;
81     }
82
83     // Returns the time in minutes, to cook a single item
84     /// <summary>
85     /// Returns the time in minutes, to cook a single item
86     /// </summary>
87     /// <returns>
88     /// A double of the cooking time for one item
89     /// </returns>
90     public double SingleCookingTime()
91     {
92         String prompt = "Enter the time for one item (minutes)";
93         double time = ReadDouble(prompt);
94         while (time <= 0.0)
95         {
96             Console.WriteLine("Please enter a time more than 0.0");
97             time = ReadDouble(prompt);
98         }
99         return time;
100     }
101
102     // Returns the recommended cooking time for the number of items
103     /// <summary>
104     /// Returns the recommended cooking time for the number of items
105     /// </summary>
106     /// <returns>
```

```
107     /// A double of the cooking time for the number of items, or -1
108     /// if more than 3 items are to be cooked
109     /// </returns>
110     public double CookingTime(int items, double singleTime)
111     {
112         switch (items)
113         {
114             case 1: return singleTime;
115             case 2: return singleTime * 1.5;
116             case 3: return singleTime * 2;
117             default: return -1; // number of items not recommended
118         }
119     }
120
121     // Outputs recommended cooking time for the number of items
122     /// <summary>
123     /// Outputs the recommended cooking time for the number of items
124     /// </summary>
125     public void RecommendedCookingTime()
126     {
127         int items = NumberOfItems();
128         double singleTime = SingleCookingTime();
129         double cookingTime = CookingTime(items, singleTime);
130         if (cookingTime > -1)
131         {
132             String output = String.Format("Recommended cooking time: {0}
133             ↪ minutes", cookingTime);
134             Console.WriteLine(output);
135         }
136         else
137         {
138             Console.WriteLine("Maximum of 3 items recommended");
139         }
140     }
141
142     static void Main(String[] args)
143     {
144         Microwave microwave = new Microwave();
145         microwave.RecommendedCookingTime();
146
147         Console.ReadLine();
148     }
149 }
```

```
1  using System;
2
3  namespace Program_6
4  {
5      class DoCasting
6      {
7          static void Main(string[] args)
8          {
9              int sum = 17;
10             int count = 5;
11
12             int intAverage = sum / count;
13             Console.WriteLine(intAverage); // average is integer division and not
14             → precise
15
16             double doubleAverage = 0.0;
17             doubleAverage = sum / count;
18             Console.WriteLine(doubleAverage); // still integer division and not
19             → precise answer
20
21             doubleAverage = (double)sum / count;
22             Console.WriteLine(doubleAverage); // Now more precise as the division
23             → now uses doubles
24         }
25     }
26 }
```