# .NET Developer Practicum

**The .NET Developer practicum is evaluated on:**

1. Object Oriented Design

* Clear separation of concerns
* Use of Data / Service / Controller layers
* Use of interfaces for modularity
* Use of IOC via Unity (XML configured)

1. Readability

* Regular comments
* Standard formatting / spacing

1. Maintainability

* Separated input data files for simple test domain change
* Separated unit test project
* Clear separation of concerns

1. Testability

* Mock driven test methods
* Separate test data embedded resource (editable)
* Modular component design

**Technical Requirements:**

1. Create this solution as a console application
2. Solution must have unit tests
3. Solution must have a build script that can compile and test the solution from the command line
4. Put your solution in a GitHub repository, and send us a link when done

**Rules:**

1. You must enter time of day as “morning” or “night”
2. You must enter a comma delimited list of dish types with at least one selection
3. The output must print food in the following order: entrée, side, drink, dessert
4. There is no dessert for morning meals
5. Input is not case sensitive
6. If invalid selection is encountered, display valid selections up to the error, then print error
7. In the morning, you can order multiple cups of coffee
8. At night, you can have multiple orders of potatoes
9. Except for the above rules, you can only order 1 of each dish type

**Dishes for Each time of day**

|  |  |  |
| --- | --- | --- |
| Dish Type | morning | night |
| 1 (entrée) | eggs | steak |
| 2 (side) | Toast | potato |
| 3 (drink) | coffee | wine |
| 4 (dessert) | *Not Applicable* | cake |

**Sample Input and Output:**

Input: morning, 1, 2, 3

Output: eggs, toast, coffee

Input: morning, 2, 1, 3

Output: eggs, toast, coffee

Input: morning, 1, 2, 3, 4

Output: eggs, toast, coffee, error

Input: morning, 1, 2, 3, 3, 3

Output: eggs, toast, coffee(x3)

Input: night, 1, 2, 3, 4

Output: steak, potato, wine, cake

Input: night, 1, 2, 2, 4

Output steak, potato(x2), cake

Input: night, 1, 2, 3, 5

Output: steak, potato, wine, error

Input: night, 1, 1, 2, 3, 5

Output: steak, error