

***Front-end Advanced***

**Training Assignment**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 7/1/2019 |

**Hanoi, mm/yyyy**

RECORD OF CHANGES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
| 1 | 30/May/2019 | Create a new assignment | Create new | DieuNT1 | VinhNV |
| 2 | 07/Jun/2019 | Update Fsoft Template | Update | DieuNT1 | VinhNV |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[Day 5-6. Unit 3: OOP and Prototype 4](#_Toc20736977)

[Objectives 4](#_Toc20736978)

[Specification 4](#_Toc20736979)

[Follow up: 6](#_Toc20736980)

[Hints: 6](#_Toc20736981)

|  |  |
| --- | --- |
|  | **CODE: FEA.M.A302 (Quiz App)**  **TYPE: Medium**  **LOC: 300**  **DURATION: 180 minutes** |

# Day 5-6. Unit 3: OOP and Prototype

Objectives

* Understand the core concepts, step to analysis, design and program to OOP
* Able to recognize class, field, method, interactions between classes
* Understand the 4 attributes of OOP: Abstraction, Inheritance, Encapsulation and Polymorphism
* Undertand OOP Principles to create Abstract class, Interface
* Able to modeling real world object into OOP style to solve problem such as Employee Management, Book Management

Specification

You are given a Quiz app using OOP style (folder **src**).

The app consists of 3 mains parts (see below figure)

1. The title: JavaScript Online Quiz
2. The Question Container which contains the Question and a list of choice. User can only choose 1 answer from the list.
3. Button Group: Previous Question, Next Question, Submit

|  |
| --- |
|  |

Suppose we have 20 Questions in total.

If the current question is 1 (count from 1 to 20), only “**Next Question”** button is shown. Clicking on this button will show the next question (in this case question number 2).

If current question is from 2 to 20, then **Previous Question** must be shown. Clicking on that button will show the previous question.  
If you are at last question (Question number 20), you must show **Submit Quiz** button and hide Next Question (there is no more next question). If user click on this button, you must show his/her total score below the Button Group

|  |
| --- |
|  |

Find below the list of questions represented in JavaScript:

1. const myQuestions = [
2. {
3. question: 'Javascript is \_\_\_\_\_\_\_\_\_ language.',
4. answers: {
5. a: 'Programming',
6. b: 'Application',
7. c: 'None of These',
8. d: 'Scripting'
9. },
10. multi: false,
11. correctAnswer: 'd'
12. },
13. {
14. question:
15. 'Which of the following is a valid type of function javascript supports?',
16. answers: {
17. a: 'named function',
18. b: 'anonymous function',
19. c: 'both of the above',
20. d: 'none of the above'
21. },
22. multi: false,
23. correctAnswer: 'c'
24. },
25. {
26. question:
27. 'Which built-in method returns the index within the calling String object of the first occurrence of the specified value?',
28. answers: {
29. a: 'getIndex()',
30. b: 'location()',
31. c: 'indexOf()',
32. d: 'getLocation()'
33. },
34. multi: false,
35. correctAnswer: 'c'
36. },
37. {
38. question: 'Which one of the following is valid data type of JavaScript',
39. answers: {
40. a: 'number',
41. b: 'void',
42. c: 'boolean',
43. d: 'nothing'
44. },
45. multi: false,
46. correctAnswer: 'c'
47. }
48. ];

Problem 01:

Using your understanding of OOP (and `**this**` keyword) fix the problem when user click on **Previous** button, **Next** button, **Submit** button

Problem 02:

Now you have to extend the apps to support Multiple Choice Question (in HTML it’s a checkbox instead of radios). Implement the **MultiChoiceQuestion** class to fulfil the requirement.