

## AY2021S2-CS2113\_CS2113T-Mock

Your Name: \_\_\_\_\_

Your ID: \_\_\_\_\_

# of Questions: 26

Duration: 90 minutes

Total Exam Points: 25.00

---

Answers are written/marked in bold

### Question #: 1

[Note: this question has 2 parts. Use bullet points and indicate the answer to each part explicitly]

Your team has been asked to implement BikeShare, a software application for managing a bicycle-sharing system where bicycles are made available for shared use to individuals on a short-term basis.

Part 1: Give one must-have user story

Part 2: Give two *non-functional requirements* of BikeShare that are directly related to functional requirement above

**Part 1:**

- **As a bike user, I can unlock the bike so that I can start the trip**

**Part 2:**

- **The app should handle updates for 1000+ bicycles at a time without crashing.**
- **The app should update the status of the bike in less than 10 seconds.**

**Item Weight:** 3.0

---

## Question #: 2

Design an efficient and effective test cases for the method given below. Show the intermediate steps to your test case design. Give at least 7 but no more than 10 test cases.

```
/**
 * Returns the class size of the specified module.
 * @param moduleCode should be in the range 1000..6999
 * @throws Exception if the moduleCode is not in range or the user is
 * not logged in or if the user is not an instructor
 */
int getClassSize(int moduleCode)throws Exception{
    //...
}
```

	Equivalence classes	Values to test
moduleCode	<b>=&lt; 999</b>	<b>999</b>
	<b>1000...6999</b>	<b>1000, 6999</b>
	<b>&gt;= 7000</b>	<b>7000</b>
Is instructor?	<b>Yes</b>	<b>Yes</b>
	<b>No</b>	<b>No</b>
Is logged in?	<b>Yes</b>	<b>Yes</b>
	<b>No</b>	<b>No</b>

Test cases:

	moduleCode	Is instructor?	Is logged in?	Expected
1	<b>1000</b>	<b>Yes</b>	<b>Yes</b>	<b>Success</b>
3	<b>6999</b>	<b>Yes</b>	<b>Yes</b>	<b>Success</b>
3	<b>999</b>	<b>Yes</b>	<b>Yes</b>	<b>Exception</b>
4	<b>7000</b>	<b>Yes</b>	<b>Yes</b>	<b>Exception</b>
5	<b>Any value in 1000...6999</b>	<b>No</b>	<b>Yes</b>	<b>Exception</b>
6	<b>Any value in 1000...6999</b>	<b>Yes</b>	<b>No</b>	<b>Exception</b>
7	<b>A non-boundary value in 1000...6999</b>	<b>Yes</b>	<b>Yes</b>	<b>Success</b>

Item Weight: 4.0

---

### Question #: 3

Suppose you are reviewing the code below. Assume the coding standard is similar to the one you used in the CS2113/T.

```
1. /**
2.  * Sets the address to the given value
3.  */
4. public void address(String address) {
5.     //set address to a default value if null
6.     if (address == null) {
7.         this.address = "ABC avenue";
8.         return;
9.     }
10.    //set address to given value
11.    this.address = address;
12.}
```

Choose the incorrect option.

- A. The method is named inappropriately
- B. The header comment is missing some vital information about the method's behavior
- C. The commenting intensity inside this method (i.e. excluding the header comment) is a perfect**
- D. The code has an issue related to magic numbers

Item Weight: 1.0

---

Question #: 4

Identify the coding standard violation(s) in this code

```
1. public void do_something() {  
2.     for (int i = 0; i < 5; i++) {  
3.         something();  
4.         somethingElse();  
5.     }  
6. }
```

Write your answer in the following format:

LineNumber: violation

(Example format: for a piece of code unrelated to this question:

10: redundant comment

**1: Method names should be in camelCase**

Item Weight: 1.0

---

Question #: 5

Choose the incorrect option:

You may not want to reuse components in your software because:

- A. The code may be immature
- B. The code may not have the appropriate license
- C. The code is bug-free**
- D. The code may be malicious

Item Weight: 1.0

---

**Question #:** 6

You are testing the `isWithinAWeek()` method below. You have already chosen 4 as one of the test inputs.

```
/**
 * Returns true if the length is within the span of a week (in days)
 */
boolean isWithinAWeek(int length) {
    ...
}
```

Which of the following inputs is least suitable as the next test case?

- A. 3
- B. 1
- C. 7
- D. 8

**Item Weight:** 2.0

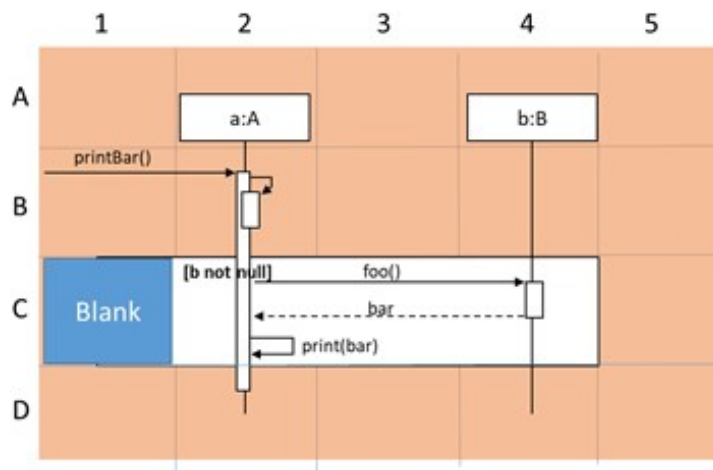
---

### Question #: 7

Consider the code fragment below:

```
class A {  
    B b;  
    // ... more code  
  
    void printBar() {  
        // some self call  
        if(b != null) {  
            bar = b.foo();  
            print(bar);  
        }  
    }  
}
```

Choose the appropriate block to fill the blank cell, C1.



A.

if

B. (Answer)

opt

C.

alt

D.

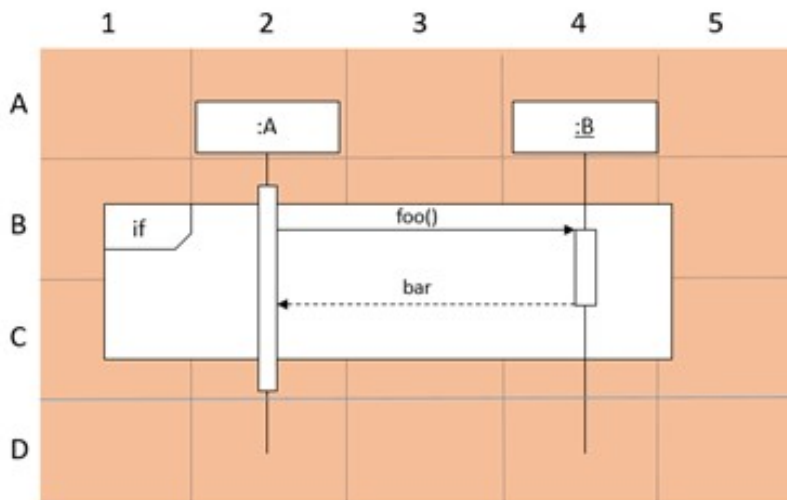
switch

Item Weight: 1.0



**Question #: 8**

Identify the errors in the following sequence diagram (using the cell(s) (i.e., A1, A2, ..., D5))



[Note: No partial credits]

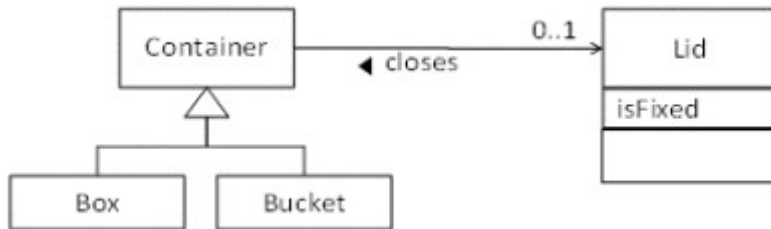
- A. B1 → opt
- B. A4 → no underline for object in sequence diagram
- C. B2 → lifeline should be dashed, not solid
- D. B4 → lifeline should be dashed, not solid
- E. D2 → lifeline should be dashed, not solid
- F. D4 → lifeline should be dashed, not solid
- G. C3

**Item Weight:** 2.0

---

Question #: 9

Consider the class diagram below.



Which of the following diagrams is an object diagram compliant with the given class diagram?

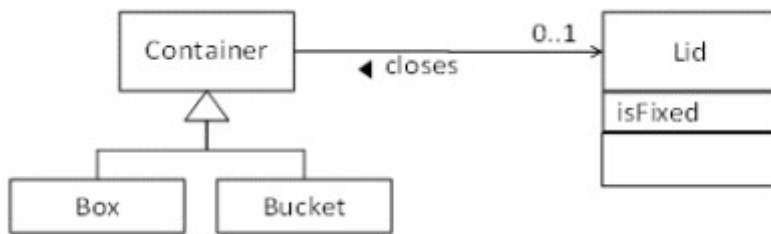
- A.   
Object diagram A shows an object **:Container** with an arrow pointing to an object **:Lid**.
- B.   
Object diagram B shows an object **:Lid** with an arrow pointing to an object **:Box**.
- C.   
Object diagram C shows an object **:Container** with an arrow pointing to an object **:Bucket**.
- D. Answer   
Object diagram D shows an object **:Bucket** with an arrow pointing to an object **:Lid**.

Item Weight: 1.0

---

**Question #:** 10

Consider the class diagram below.



Identify the correct statement

**A. The following code compiles**

```
Container c = new Box();
```

B. Container is an abstract class

C. A Lid must be associated to a Container but it is optional for a Container to have a Lid

D. There is a dependency and coupling but not association between Container and Lid

**Item Weight:** 1.0

---

**Question #: 11**

When developing software to compete with Facebook, an iterative approach is more suitable than a sequential approach.

- A. True**
- B. False

**Item Weight:** 0.5

---

**Question #: 12**

Testing is a verification type QA activity

- A. True**
- B. False

**Item Weight:** 0.5

---

**Question #: 13**

Refactoring can improve performance of the refactored code.

- A. True**
- B. False

**Item Weight:** 0.5

---

**Question #: 14**

The module TP uses both CI and CD.

- A. True
- B. False**

**Item Weight:** 0.5

---

**Question #:** 15

Gradle is a continuous integration tool.

- A. True
- B. False**

**Item Weight:** 0.5

---

**Question #:** 16

Method overloading can happen within a single class.

- A. True**
- B. False

**Item Weight:** 0.5

---

**Question #:** 17

System testing covers mostly negative test cases while acceptance testing covers mostly positive test cases

- A. True
- B. False**

**Item Weight:** 0.5

---

**Question #:** 18

This is a correct usage of composition to represent the fact that a Task may consist of smaller Tasks



**A. True**

B. False

**Item Weight:** 0.5

---

**Question #:** 19

A coding standard can contain rules that can be objectively enforced

**A. True**

B. False

**Item Weight:** 0.5

---

**Question #:** 20

This is a valid functional requirement: "The quiz should be accessible to 250 students concurrently"

A. True

**B. False**

**Item Weight:** 0.5

---

**Question #:** 21

When writing developer documentation, it is more important to be comprehensive than comprehensible

- A. True
- B. False**

**Item Weight:** 0.5

---

**Question #:** 22

Grayed out code reported by IntelliJ IDEA is an example of static analysis

- A. True**
- B. False

**Item Weight:** 0.5

---

**Question #:** 23

The module project uses a depth-first iterative model

- A. True
- B. False**

**Item Weight:** 0.5

---

**Question #:** 24

The code below compiles

```
void bar(boolean isValid) throws Exception{
    if(!isValid){
        throw new String("Invalid data");
    }
}
```

- A. True
- B. False**

**Item Weight:** 0.5

---

**Question #:** 25

The following command feeds the text in input.txt into AddressBook program and saves the output in the output.txt file.

```
java input.txt > AddressBook > output.txt
```

- A. True
- B. False**

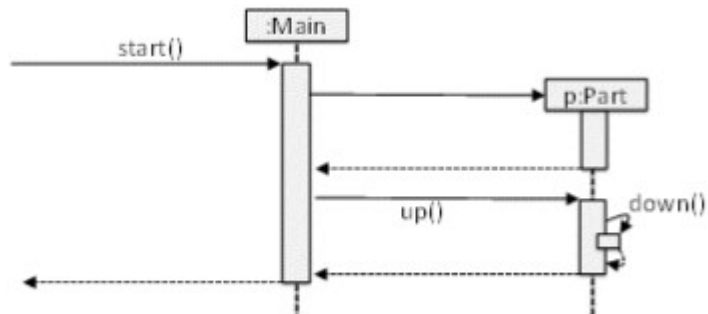
**Item Weight:** 0.5

---



Question #: 26

Consider the sequence diagram below:



The code below is compliant with the diagram

```
class Main{
void start(){
    // some code here
    Part p = new Part();

    // some more code here
}
}
```

A. True

B. False

Item Weight: 0.5