You have completed

Week 6 Quiz

You scored **10/11**

Have not received the confirmation Email? Click here to re-send.

Display chosen by module coordinator

SCROLL DOWN TO VIEW DETAILED RESULTS

Requirements

https://nus-cs2113-ay2122s1.github.io/website/se-book-adapted/chapters/requirements.html

1. Some requirements can be discarded if they are considered ‘out of scope’.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As stated in the textbook.

2. This is an example of an NFR: All functionality of the system should be usable by hearing impaired users as well.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. It is not a functionality, but rather how the functionality should operate.

3. Ideally, a requirement should not be divisible any further.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. The textbook uses the term atomic to describe this quality.

4. Suppose you are writing an online game for primary school children in Singapore. This is a private project that you are doing by yourself. It will be free for anyone to play. A potential stakeholder of this project is the Singapore Government.

(1 mark) ✗

You scored 0 / 1 mark

True

○ False

General Comments

True.

A country's government is potentially a stakeholder for any software that is being used by its citizens. For example, the government would not want a software that can be harmful to the community, especially if the community in concern is considered vulnerable. The point is, a party that is not directly involved in a project can still be a stakeholder.

5. As per the textbook, brown-field projects are usually harder than green-field projects.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. There is no such comparison in the textbook.

6. One may have to spend an extra effort in digging NFRs out as early as possible because they are easier to miss.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As stated in the textbook.

7. Non-functional requirements specify the constraints under which system is developed and operated

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

8. Requirements should be specified as close to implementation as possible, so as to minimize errors in implementing it.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. In contrast, requirements should be implementation-free.

Gathering requirements

https://nus-cs2113-ay2122s1.github.io/website/se-book-adapted/chapters/gatheringRequirements.html

9. Wireframe diagrams can be used for prototyping a UI.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. An example is given in the textbook.

10. Focus groups are a kind of informal group interview.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.

Quoting the textbook:
Focus groups are a kind of informal interview within an interactive group setting.

11. Brainstorming aims to generate ideas; not to validate them.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.

Quoting the textbook:

In a brainstorming session there are no "bad" ideas. The aim is to generate ideas; not to validate them.

10/11 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11

You have completed

Week 7 Quiz

You scored **24/32**

Have not received the confirmation Email? Click here to re-send.

Display chosen by module coordinator

SCROLL DOWN TO VIEW DETAILED RESULTS

Guideline: Maximize Readability

1. It is recommended to arrange the code of a method to follow the arrow-head style so that it is easier to read.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. Arrow-head style is to be avoided.

2. Magic literals are to be avoided.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As given in the textbook.

3. SLAP stands for Single Level of Abstraction Per method.

(1 mark) ✗

You scored 0 / 1 mark

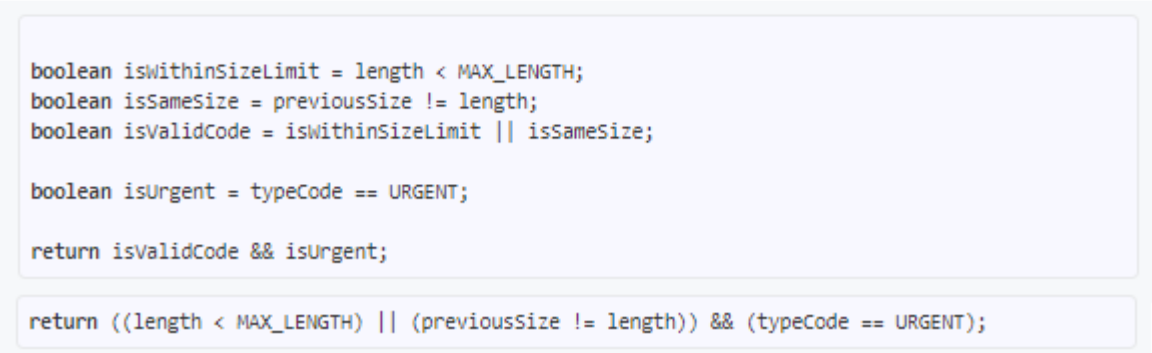
True

○ False

General Comments

True. As per the textbook.

4. As per the textbook, given the two code segments are equivalent, the second one is the better choice because it is shorter.



```
boolean isWithinSizeLimit = length < MAX_LENGTH;
boolean isSameSize = previousSize != length;
boolean isValidCode = isWithinSizeLimit || isSameSize;

boolean isUrgent = typeCode == URGENT;

return isValidCode && isUrgent;
```

```
return ((length < MAX_LENGTH) || (previousSize != length)) && (typeCode == URGENT);
```

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. This is the opposite of the example given in the textbook.

5. One can use whitespace to emphasize the logical structure of the code.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.

Quoting the textbook:
Just like we use section breaks, chapters and paragraphs to organize a story, use classes, methods, indentation and line spacing in your code to group related segments of the code. For example, you can use blank lines to group related statements together.

6. One can use guard clauses to make the 'happy path' more prominent.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.

Quoting the textbook:
The happy path (i.e. the execution path taken when everything goes well) should be clear and prominent in your code. ... One technique that could help in this regard is the use of guard clauses.

7. One should never prioritize efficiency or performance over readability.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False.

Quoting the textbook:
Note that there are cases where optimizing takes priority over other things.

8. As per the KISS principle, one should *always* prefer the simpler solution over more clever solutions.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. Not *always*. Rather, one should not discard the simpler solution just because there is a more 'clever' solution. Instead, the 'clever' solution should be chosen only if the additional cost of complexity is justifiable.

9. In general, a smaller number of long methods is better than a larger number of short methods.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. In general, long methods are to be avoided.

10. In some cases, hand-optimizing code can make it harder for the compiler to optimize the same code.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

Guideline: Follow a Standard

11. Names such as i, j, k should not be used as variable names as they are not descriptive enough.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

They can be used in small scopes, such as a counter in a loop.

12. Of the two statements below, the first one has the correct spacing.

```
a = (b + c) * d;
a=(b+c)*d;
```

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

For example, spaces are required around the '+' sign.

13. It is better if each developer followed they own style of coding so that the code can be traced to the author easily.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

Tracing the code to an author can be done using other tools such as Revision Control Software.

14. One aim of adopting a coding standard is to make the entire code base look like it was written by one person.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

Quoting the textbook:

>The aim of a coding standard is to make the entire code base look like it was written by one person.

15. This is an acceptable opening sentence for a javadoc header comment.

```
/**
 * Add the value to the current list.
 * ...
 */
```

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

It should be '`Adds ...`'

16. When wrapping a long statement (an example given below), one should indent lines using two tabs instead of the usual one tab.

```
totalSum = a + b + c
          + d + e;
```

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

Spaces (not tabs) should be used for indentation.

17. Instead of the following style, one should use the Egyptian style.

```
while (!done)
{
    doSomething();
    done = moreToDo();
}
```

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

Egyptian style is preferred.

18. Wildcard imports (an example given below) should not be used unless there are many classes being imported from the same package.

```
import java.util.*;
```

(1 mark) ✗

You scored 0 / 1 mark

True

○ False

General Comments

All imports must be explicit.

19. A developer should understand the importance of following a coding standard. However, there is no need to follow one.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

The main point of a coding standard is for everyone to follow it.

20. There are tools that can help to enforce some parts of a coding standard e.g. indentation rules.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

Quote from the textbook:

> IDEs can help to enforce some parts of a coding standard e.g. indentation rules.

21. This variable name is compliant with the coding standard.

```
boolean processingStatus = false;
```

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

Boolean variables should be named to sound like booleans.

22. This variable name is compliant with the coding standard.

```
final static int RED = 1;
```

(1 mark) ✓

You scored 1 / 1 mark

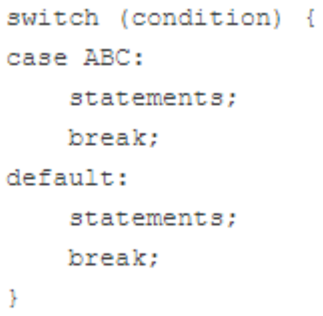
True

False

General Comments

A constant can be named using ALL_CAPS.

23. This indentation is compliant with the coding standard:



```
switch (condition) {
case ABC:
    statements;
    break;
default:
    statements;
    break;
}
```

True

False

General Comments

As per our coding standard, case clauses should not be indented.

Guideline: Name Well

24. These variable names, found in a single class, is problematic.

- colorBlack: hex value for color black
- colorWhite: hex value for color white
- colorBlue: number of times blue is used
- colorGrey: hex value for color grey

(1 mark) ✓

You scored 1 / 1 mark

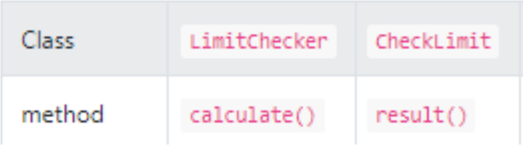
True

False

General Comments

True. They all look similar but one of them has a different meaning.

25. The third column has the better choice for a class/method name.



Class	`LimitChecker`	`CheckLimit`
method	`calculate()`	`result()`

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. Class names should be nouns and method names should be verbs.

26. This is an acceptable variable name.

```
ArrayList<Person> friend; // persons the account has “friended”
```

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. A singular name is being used to represent a collection of objects.

Guideline: Avoid Unsafe Shortcuts

27. It is recommended to include a default branch in case statements.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

28. All local variables used in a method should be declared at the start of the method, for easier readability.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. Rather, define variables in the least possible scope. For example, if the variable is used only within the if block of the conditional statement, it should be declared inside that if block.

29. Using empty catch blocks is not recommended.

(1 mark) ✓

You scored 1 / 1 mark

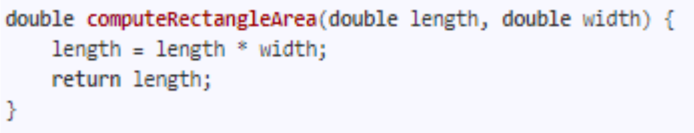
True

False

General Comments

True. As per the text book.

30. This method’s use of the parameters involves an error-prone practice.



```
double computeRectangleArea(double length, double width) {
    length = length * width;
    return length;
}
```

(1 mark) ✓

You scored 1 / 1 mark

True

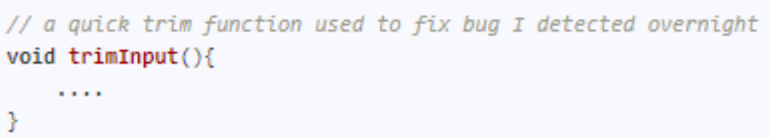
False

General Comments

True. It reuses a parameter as a local variable.

Guideline: Comment Minimally, Sufficiently

31. One type of useful code comments is 'note-to-self' type comments programmers add to remind themselves of additional info (such as the one given below).



```
// a quick trim function used to fix bug I detected overnight
void trimInput(){
    ....
}
```

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

Comments should be written to the reader, not as note-to-self.

32. Code comments show explain the WHAT and HOW aspects of the code but not the WHY aspect.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

Of the three, WHY is not the one that should be omitted.

33. Only one of these comments is useful.

```
// increment x
x++;

//trim the input
trimInput();
```

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

Both are redundant comments

25/33 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31 32 33

You have completed

Week 8 Quiz

You scored **9/16**

Have not received the confirmation Email? Click here to re-send.

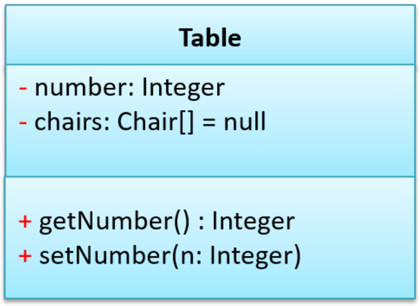
Display chosen by module coordinator

SCROLL DOWN TO VIEW
DETAILED RESULTS

Class diagrams: classes

UML → Class Diagrams → Classes → What
UML → Class Diagrams → Class-Level Members → What

1. In the following class diagram, the '+' and '-' signs to indicate *accessibility* of the attributes and the methods.



(1 mark) ✓

You scored 1 / 1 mark

True

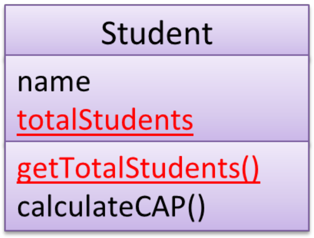
False

General Comments

False. It’s called *visibility*, not *accessibility*.

2.

In the class diagram below, there is only one class-level method.



(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. The other is a class-level attribute.

3.

The first diagram follows the class diagram notation correctly, but the other two don’t.

(1 mark) ✗

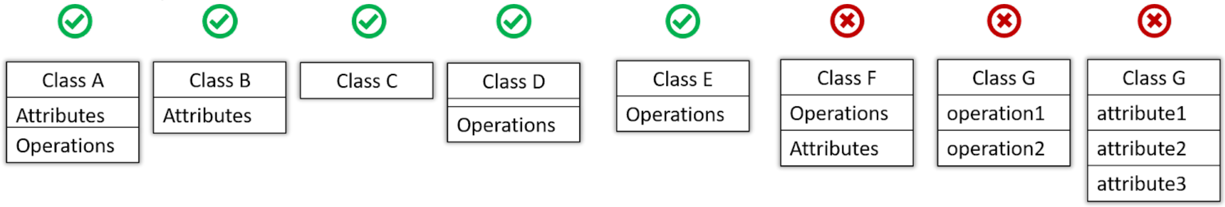
You scored 0 / 1 mark

True

○ False

General Comments

True. Quoting the textbook:

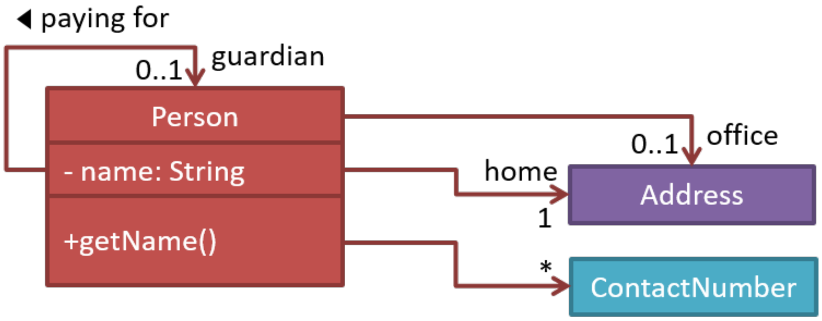


Class diagrams: association lables, roles, multiplicity, navigability

OOP → Associations → What

UML → Class Diagrams → Associations → What
UML → Class Diagrams → Associations → Labels
UML → Class Diagrams → Associations → Roles
OOP → Associations → Multiplicity
UML → Class Diagrams → Associations → Multiplicity
OOP → Associations → Navigability
UML → Class Diagrams → Associations → Navigability

4.



There can be associations between Person objects.

(1 mark) ✓

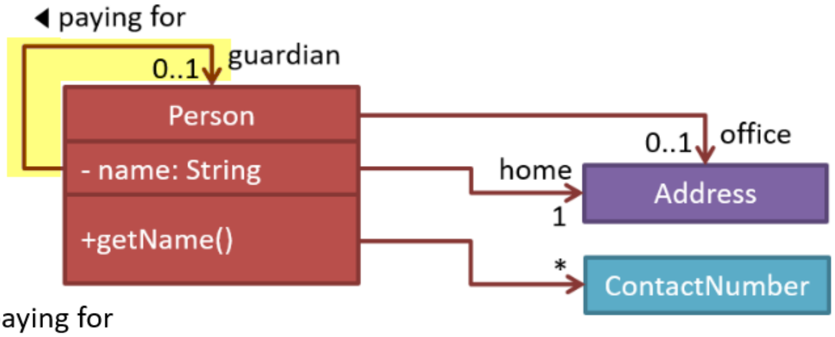
You scored 1 / 1 mark

True

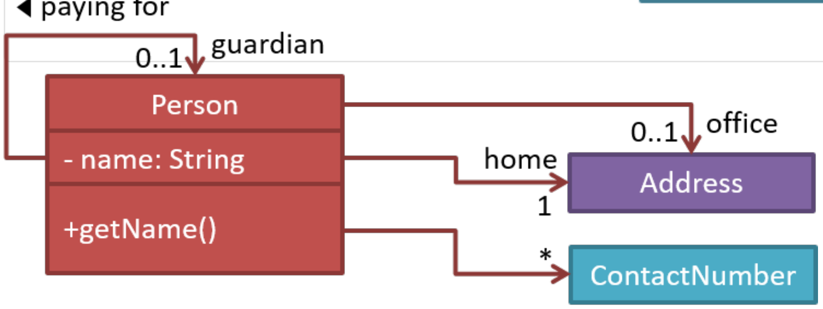
False

General Comments

True. See highlighted association link.



5.



Assume Peter is the guardian of Jake. The object representing Jake has a reference to the object representing Peter. But the object representing Peter does not have a reference to the object representing Jake.

(1 mark) ✗

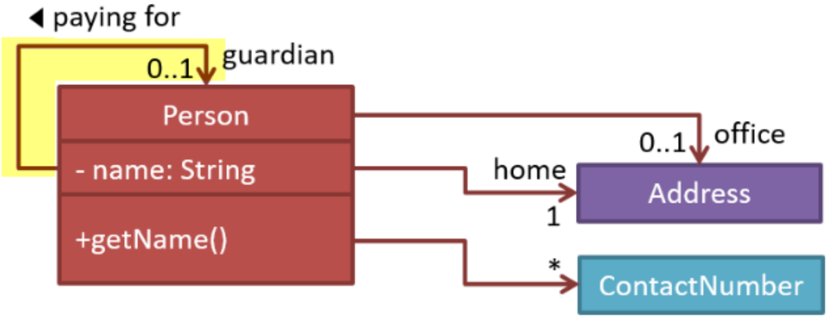
You scored 0 / 1 mark

True

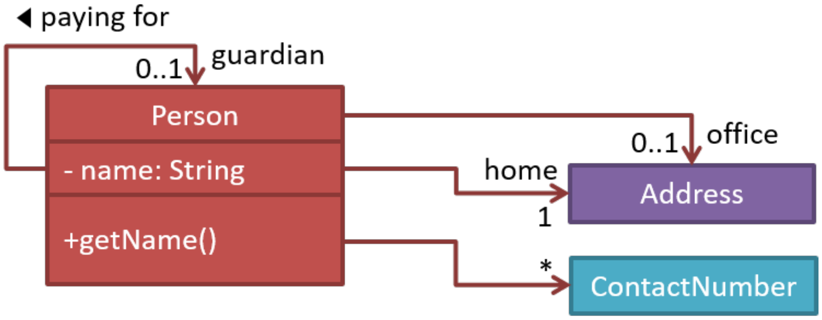
○ False

General Comments

True. Based on the navigability direction of this association.



6.



A Person object can exist without a contact number.

(1 mark) ✓

You scored 1 / 1 mark

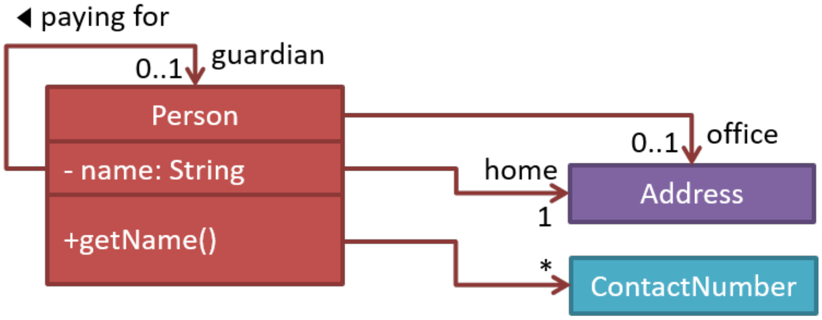
True

False

General Comments

True. The multiplicity * can mean 0 too.

7.



A person can have up to two Address objects associated with it.

(1 mark) ✓

You scored 1 / 1 mark

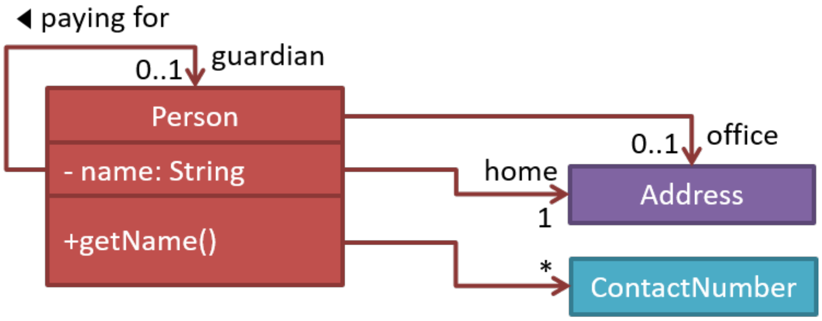
True

False

General Comments

True. A person must have one home address and at most one office address.

8.



Only one Person object can keep a given Address object as its home address.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

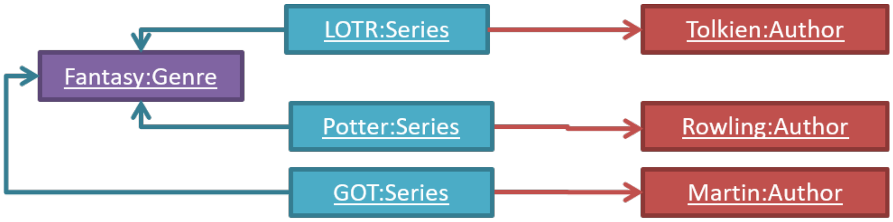
False. While a Person object must have exactly one Address object, the multiplicity at the other end is unspecified.

Object diagrams

Design → Modelling → Modelling Structure → Object Diagrams

Tools → UML → Object vs Class Diagrams

9.



All objects are underlined because they are class-level members.

(1 mark) ✗

You scored 0 / 1 mark

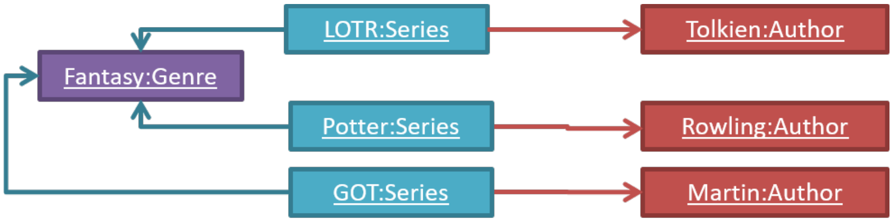
○ True

False

General Comments

False. Underlining class-level members is part of the class diagram notation, not object diagrams.

10.



The objects in the diagram belong to three classes.

(1 mark) ✓

You scored 1 / 1 mark

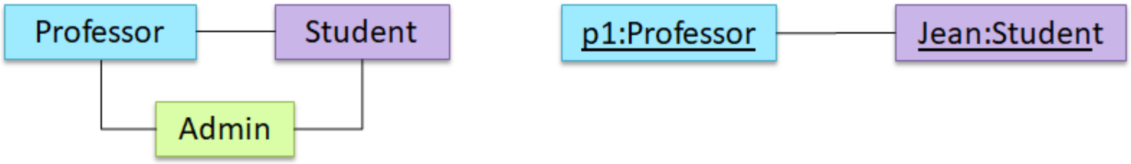
True

False

General Comments

True. Genre, Series, Author.

11.



The object diagram is compliant with the class diagram.

(1 mark) ✗

You scored 0 / 1 mark

True

○ False

General Comments

True. It does not break any rules specified in the class diagram.

12.

This object diagram is incorrect. Reason: object name not specified.

(1 mark) ✓

You scored 1 / 1 mark

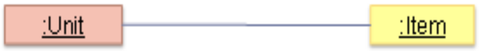
True

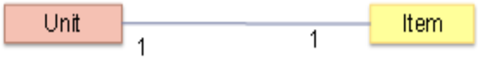
False

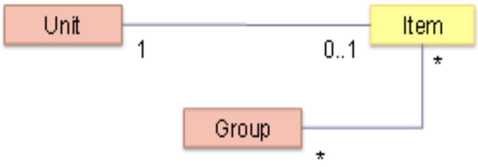
General Comments

False. It’s fine to omit the object name.

13.







Both class diagrams match the object diagram.

(1 mark) ✓

You scored 1 / 1 mark

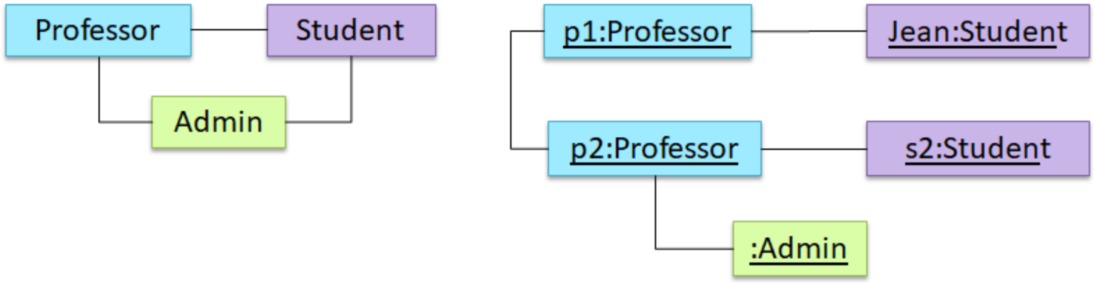
True

False

General Comments

True. Both class diagrams allow one Unit object to be linked to one Item object.

14.



The object diagram is compliant with the class diagram.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

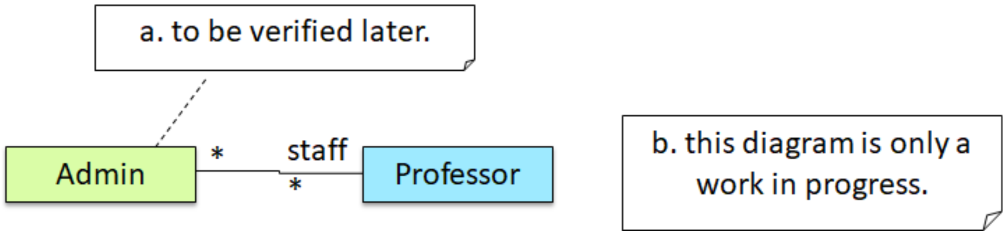
General Comments

False. There cannot be an association between to Professor objects.

UML: notes

Tools → UML → Notes

15.



Note b is not following the correct notation because it is missing the connecting line.

(1 mark) ✓

You scored 1 / 1 mark

True

False

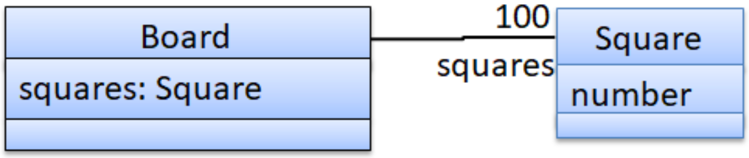
General Comments

False. The connecting line is optional.

Class diagrams: associations as attributes

Tools → UML → Class Diagrams → Associations as Attributes

16.



This diagram is showing the association correctly.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. The same association is shown as an attribute.

9/16 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

16

You have completed

Week 9 Quiz - Part I

You scored **15/17**

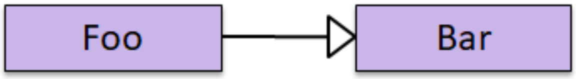
Have not received the confirmation Email? Click here to re-send.

Display chosen by module coordinator

SCROLL DOWN TO VIEW DETAILED RESULTS

Class Diagrams: Intermediate level

1.



Bar inherits from Foo.

(1 mark) ✓

You scored 1 / 1 mark

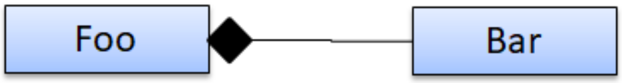
True

False

General Comments

False. Foo inherits from Bar. The triangle is placed at the parent class.

2.



Foo forms a part of Bar.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. Bar forms a part of Foo.

3. Aggregation is a whole-part relationship.

(1 mark) ✓

You scored 1 / 1 mark

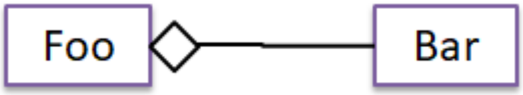
True

False

General Comments

False. It is a container-containee relationship.

4.



Foo contains Bar.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. The aggregation symbol appears near the container class.

5. The book UML Distilled (by Martin Fowler) advocates omitting the aggregation symbol altogether as it does not add value.

(1 mark) ✓

You scored 1 / 1 mark

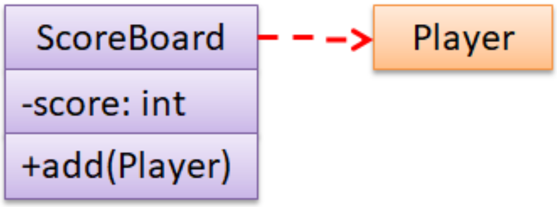
True

False

General Comments

True. As given in the textbook.

6.



The ScoreBoard class depends on the Player class.

(1 mark) ✓

You scored 1 / 1 mark

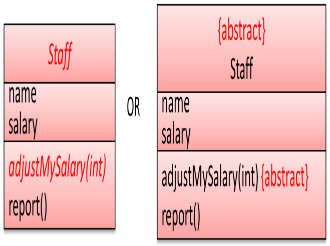
True

False

General Comments

True. The arrow points to the class that is being depended on.

7.



Staff is an abstract class.

(1 mark) ✓

You scored 1 / 1 mark

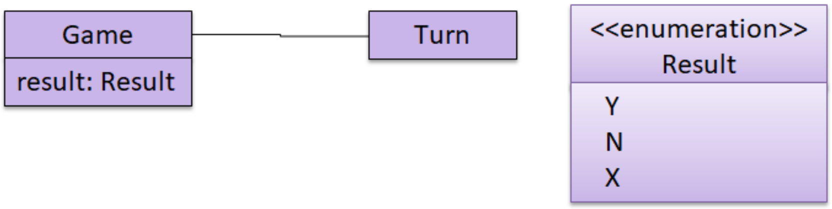
True

False

General Comments

True. Italics can be used to indicate abstract classes (but you are recommended to use {abstract} instead)

8.



Game is an enumeration.

(1 mark) ✓

You scored 1 / 1 mark

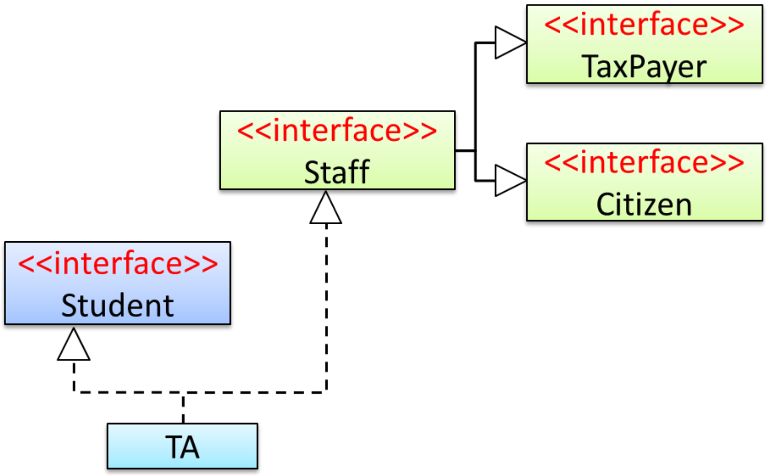
True

False

General Comments

False. The result variable of Game is of type Result, which is an enumeration.

9.



Staff interface *implements* the TaxPayer interface. TA *implements* the Staff interface.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. Staff *inherits*, not *implements*, the TaxPayer interface.

Logging

10. Java programmers need to use third-party logging libraries as Java does not have a default

logging mechanism.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. Java has a default logging facility.

11. Logging systems have features such as the following:

- the ability to enable and disable logging
- the ability to change the logging intensity

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As given in the textbook.

12. As per the textbook, a log file is like the auto-pilot system of an airplane.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It’s like the black box.

13. Logs are typically written to a log file but it is also possible to log information in other ways e.g. into a database or a remote server.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As given in the textbook.

Assertions

14. Java disables assertions by default.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

15. Assertions can be used to describe assumptions about the program state.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

16. In Java, assertions can be disabled without modifying the code.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

17. Java assertions can be used for exception handling.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False, They have a different purpose.

15/17 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

16 17

You have completed

Week 9 Quiz - Part II

You scored **15/19**

Have not received the confirmation Email? Click here to re-send.

Display chosen by module coordinator

SCROLL DOWN TO VIEW DETAILED RESULTS

Design Principles: Abstraction

1. Data abstraction is removing all data-related information from the view to create a higher level abstraction.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False.

Not *all data-related information*. Quoting the textbook:

> Data abstraction: abstracting away the lower level data items and thinking in terms of bigger entities

2. Abstraction can be applied repeatedly to obtain progressively higher levels of abstractions.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.
As per the textbook.

3. The textbook explains how abstraction can be divided into three sub-types: data abstraction, logic abstraction, and control abstraction.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False.
There is no mention of logic abstraction.

4. An OOP class is an abstraction over related data and behaviors.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.
As per the textbook.

Design Principles: Coupling

5. Coupling is a measure of the degree of dependence between components, classes, methods, etc.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.
As per the textbook.

6. Strong coupling is encouraged.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It’s discouraged.

7. If `Foo` is coupled to `Bar`, a change to `Bar` will require a change in `Foo`.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. As per the textbook, X is coupled to Y if a change to Y can potentially (but not necessarily always) require a change in X.

8. In one of the cases below, A is not coupled to B.

- A has access to the internal structure of B
- A and B depend on the same global variable
- A receives an object of B as a parameter or a return value
- A inherits from B

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. All causes A to be coupled to B. These are examples given in the textbook.

9. Tight coupling can make maintenance harder.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

10. As coupling increases, testability increases.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It's the reverse. When there is high-coupling, testing one component independent of the others become harder.

Design Principles: Cohesion

11. Weak cohesion lowers maintainability because a less cohesive module may need to be modified even due to unrelated causes.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

12. High cohesion is better.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.
As per the textbook.

13. Cohesion is a measure of how strongly-related and focused the various responsibilities of a component are.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.
As per the textbook.

14. All these are types of cohesion:

- code related to a single concept is kept together
- code that is invoked close together in time is kept together
- code that manipulates the same data structure is kept together

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

Design Principles: Single Responsibility Principle

15. As per SRP, a method should have only one responsibility.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. The SRP is about classes, not methods.

16. Single Responsibility Principle (SRP) states that a class should have one, and only one, reason to change.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. This is the definition given in the textbook.

Design Principles: Separation of Concerns

17. The principle of SoC leads to higher cohesion.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

18. Applying SoC increases functional overlaps among code sections that limits the ripple effect when changes are introduced to a specific part of the system.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It *reduces* such overlaps.

19. The principle of SoC states that the code which we are more concerned about should be separated out.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. That is not the definition of SoC.

15/19 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19

You have completed

Week 10 - Quiz Part I

You scored **19/20**

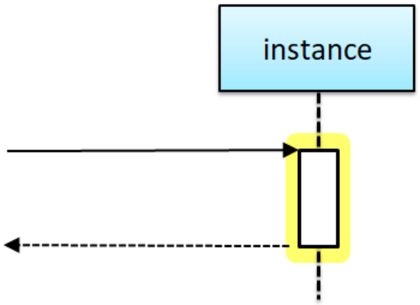
Have not received the confirmation Email? Click here to re-send.

Display chosen by module coordinator

SCROLL DOWN TO VIEW DETAILED RESULTS

Sequence Diagrams: Basics

1.



The highlighted part (in yellow) is called an ***active box***. It represents the period during which the method is actively executing.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It is called an *activation bar*.

2. A UML sequence diagram can capture the interactions between multiple objects for a given scenario, for example, an object calling a method of another object.

(1 mark) ✓

You scored 1 / 1 mark

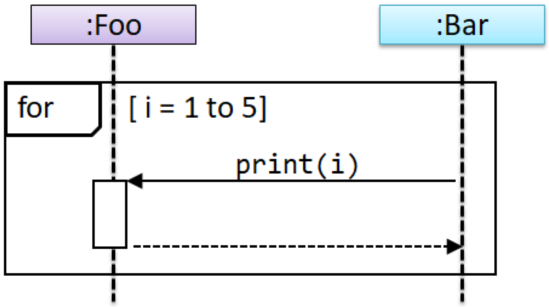
True

False

General Comments

True. As given in the textbook.

3.



This is the correct way to show a for-loop.

(1 mark) ✓

You scored 1 / 1 mark

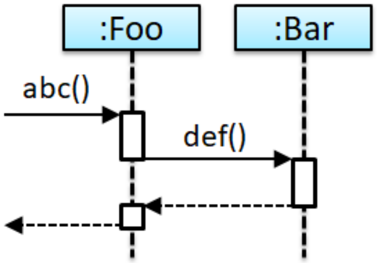
True

False

General Comments

False. The keyword to use is ‘loop’, not ‘for’.

4.



This sequence diagrams has a notation problem.

(1 mark) ✓

You scored 1 / 1 mark

True

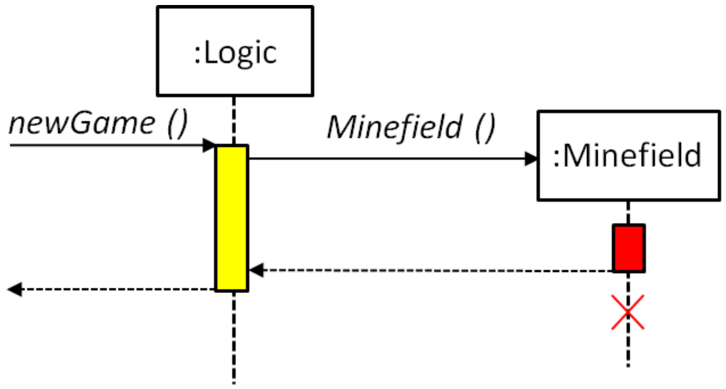
False

General Comments

True.

The activation bar of the abc() method is broken in the middle.

5.



The yellow box correctly represents a constructor.

(1 mark) ✓

You scored 1 / 1 mark

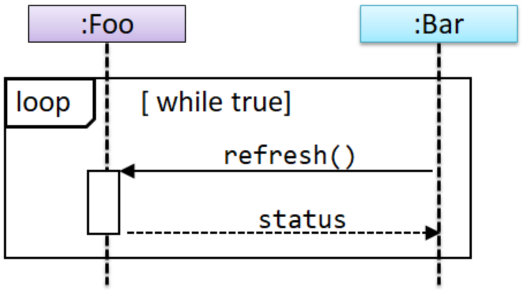
True

False

General Comments

False. That is the method calling the constructor.

6.



As per the diagram, a Foo object is calling the refresh() method of a Bar object in a loop.

(1 mark) ✓

You scored 1 / 1 mark

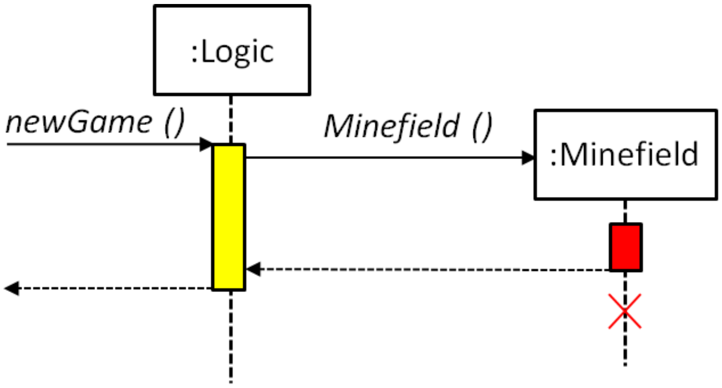
True

False

General Comments

False. it is the Bar object that is calling the Foo object.

7.



The red box correctly represents a constructor.

(1 mark) ✗

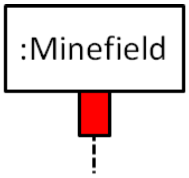
You scored 0 / 1 mark

○ True

False

General Comments

False. It should be touching the box above it, as shown below:



8. In a sequence diagram, time goes from left-to-right.

(1 mark) ✓

You scored 1 / 1 mark

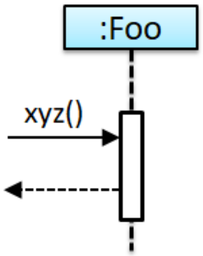
True

False

General Comments

False. It goes from top-to-bottom.

9.



This sequence diagram has a notation problem.

(1 mark) ✓

You scored 1 / 1 mark

True

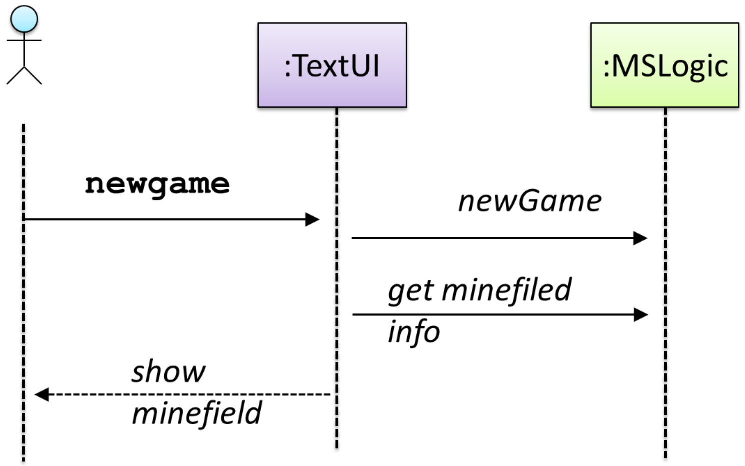
False

General Comments

True.

The activation bar is too long. As per the diagram, the method become active before it is being called; similarly, it stays active even after the method has returned.

10.



The notations missing from this diagram are optional, and therefore, the diagram is acceptable.

(1 mark) ✓

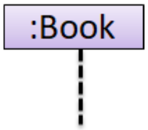
You scored 1 / 1 mark

True

False

General Comments

True. This is the example given in the textbook.

11. 

This sequence diagram correctly represents an unnamed Book object.

(1 mark) ✓

You scored 1 / 1 mark

True

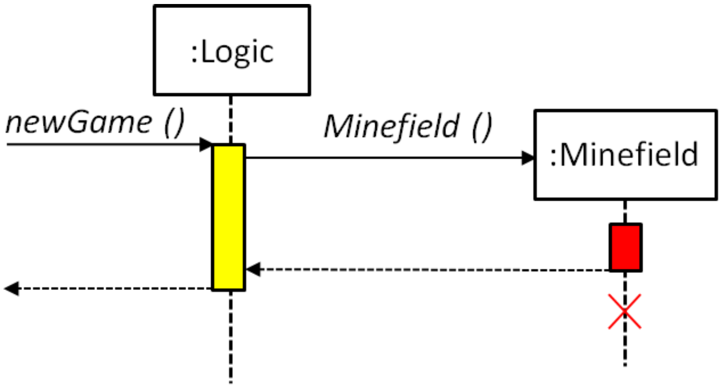
False

General Comments

True. As per the textbook.

Sequence Diagrams: Intermediate-Level

12.



The red X correctly shows that the object is deleted or it is no longer referenced.

(1 mark) ✓

You scored 1 / 1 mark

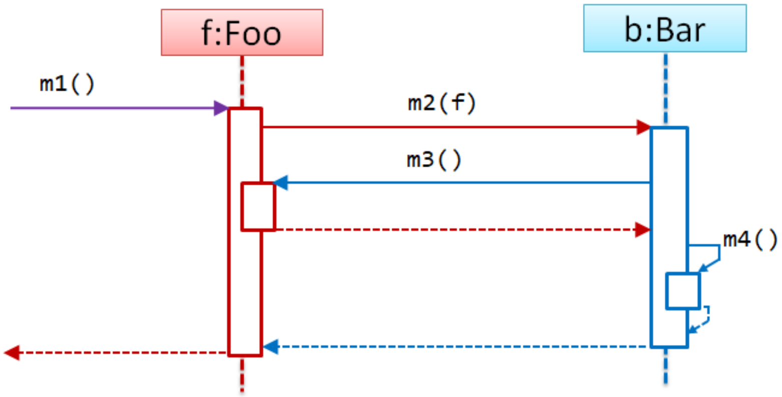
True

False

General Comments

False. The life line should not go beyond the X.

13.



Method m3 calls method m4 of self.

(1 mark) ✓

You scored 1 / 1 mark

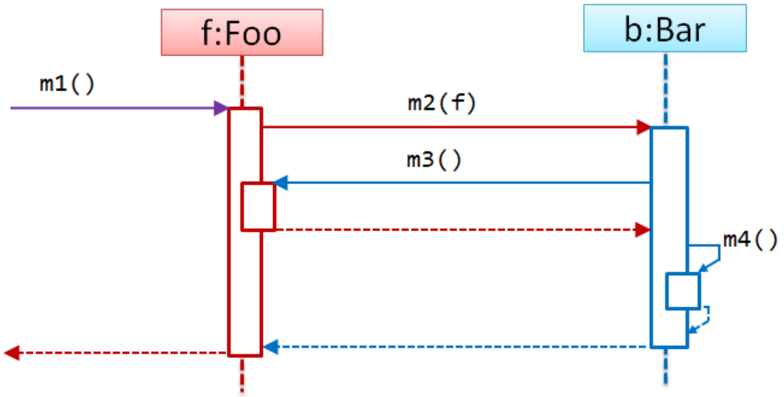
True

False

General Comments

False. It’s the method m2 that calls method m4 of self.

14.



Object b is calling method m3 of the object f.

(1 mark) ✓

You scored 1 / 1 mark

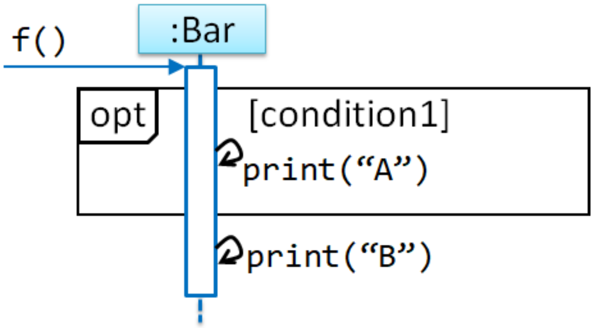
True

False

General Comments

True. It's a call back.

15.



A call to f() can result in any of the following:

- "A"
- "B"
- "AB"

(1 mark) ✓

You scored 1 / 1 mark

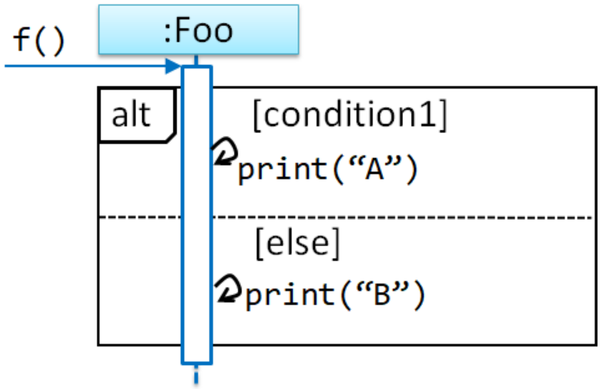
True

False

General Comments

False. It can result in “AB” or “B” as print(“A”) is in the optional path. But it cannot result in just “A”.

16.



A call to f() can result in printing “A” or “B”, but not both.

(1 mark) ✓

You scored 1 / 1 mark

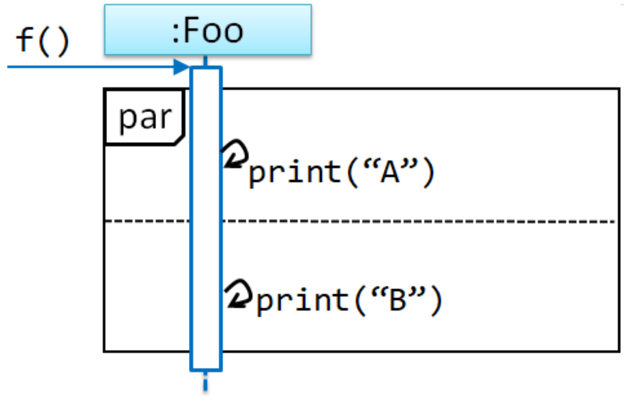
True

False

General Comments

True. They are alternatives.

17.



A call to f() can result in any of the following:

- "BA"
- "AB"

(1 mark) ✓

You scored 1 / 1 mark

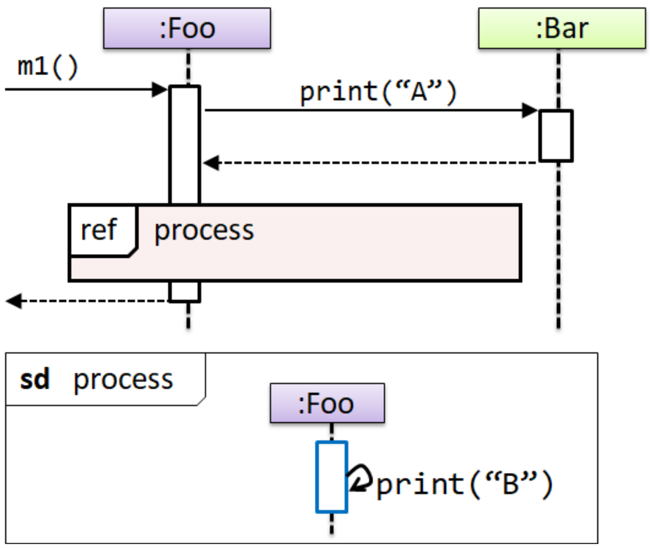
True

False

General Comments

True. The two methods are called in parallel, but we can't predict the exact sequence.

18.



A call to m1() can result in any of the following:

- “A”
- “B”

(1 mark) ✓

You scored 1 / 1 mark

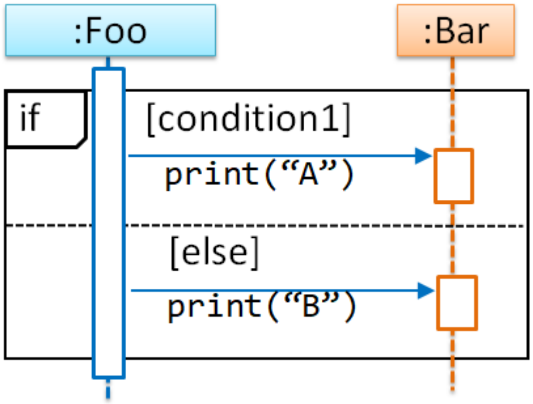
True

False

General Comments

False. It should result in “AB”.

19.



This is the correct way to show an if-else branch.

(1 mark) ✓

You scored 1 / 1 mark

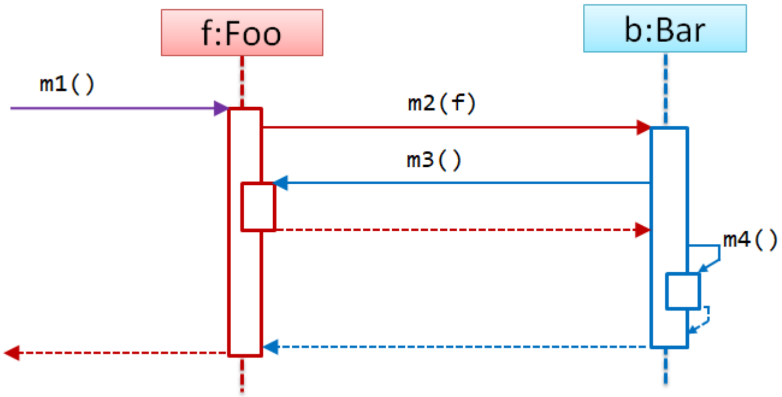
True

False

General Comments

False. The keyword is 'alt', not 'if'

20.



Method m1() calls its own method m3.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It’s the method m2 of the Bar class that calls the m3.

19/20 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

16 17 18 19 20

You have completed

Week 10 - Quiz Part II

You scored **14/14**

Have not received the confirmation Email? Click here to re-send.

Display chosen by module coordinator

SCROLL DOWN TO VIEW DETAILED RESULTS

Design Patterns

1. The singleton pattern can reduce testability.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

2. The singleton pattern can increase the coupling across the code base.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

3. Anti-patterns are commonly used solutions, which are usually incorrect or sub-optimal.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

4. The Facade pattern increase the amount of code.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. It adds an extra class.

5. A singleton class typically has at least one static method and at least one static variable. Both are private.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. The method is not private but the instance can be.

6. Facade pattern facilitates access to the functionality of a component without exposing its internal details.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True

7. The term Design Patterns was popularized by a book whose authors are also known as the 'Three Amigos'.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. The three amigos came up with UML.

8. Design patterns are elegant and reusable design solutions.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the definition in the textbook.

9. A singleton class typically has a private constructor.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True

10. The common format to describe a design pattern consists contains a context, a

problem, and a solution.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. These are the three compulsory items stated.

Test Coverage

11. Coverage analysis can be useful in improving the quality of test cases.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the text book.

12. 100% condition coverage can require more test cases than 100% branch coverage of the same code.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. A branch may require more than one condition to be true.

13. Test coverage is a metric used to measure the extent to which testing exercises the SUT.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

14. 100% path coverage can require more test cases than 100% statement coverage of the same code.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. A test suite can execute all statements without necessarily executing all possible paths of the code.

14/14 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11 12 13 14

You have completed

Week 11 Quiz

You scored **24/28**

Have not received the confirmation Email? Click here to re-send.

Display chosen by module coordinator

SCROLL DOWN TO VIEW DETAILED RESULTS

Architecture

1. UML is the standard notation used for architecture diagrams.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. Quoting the textbook:
>Architecture diagrams are free-form diagrams.

2. The software architecture shows the overall organization of the system. It can be viewed as a very high-level design of the software.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As given in the textbook.

3. In the layered architecture style, higher layers make use of services provided by lower layers.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

4. LumiNUS uses the client-server architecture style.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. All Web applications use the client-server style.

5. Operating systems often use the n-tier style.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

Types of Testing

6. Usability testing and portability testing is part of system testing. So is performance testing.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. These are three of many aspects that the textbook states as part of system testing.

7. In unit testing, we use stubs in place of dependencies. Integration testing is similar to unit testing except we use the actual dependencies instead of stubs.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False.

Quoting the textbook:

> Integration testing is not simply a case of repeating the unit test cases using the actual dependencies (instead of the stubs used in unit testing). Instead, integration tests are additional test cases that focus on the interactions between the parts.

8. System testing is the testing of a system to verify that it conforms to the specified external behavior of the system. Therefore, by definition, system tests do not go beyond the bounds defined in the specification.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False.

Quoting the textbook:

> Sometimes, system tests go beyond the bounds defined in the specification. This is useful when testing that the system fails 'gracefully' having pushed beyond its limits.

9. Acceptance testing focus more on the negative test cases while system testing focus on both positive and negative test cases.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. Acceptance testing focus more on positive test cases.

10. Integration testing is the type of done to confirm that different *parts* of the software work together as expected.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

11. Ideally, acceptance testing is to be done on the deployment site or on a close simulation of the deployment site.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

12. Beta testing is performed by the users, under controlled conditions set by the software development team.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. That’s the definition of the alpha testing.

13. Acceptance testing is done before system testing.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It comes *after*.

14. Acceptance testing (aka User Acceptance Testing (UAT) is always done by users, as opposed to professional testers.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. Not always. It can also be done by professional testers representing the users.

15. System Testing is done against the *system specification*. Acceptance Testing is done against the *requirements specification*.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

16. In some projects, one document serves as both the requirements specification and the system specification.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.

Quoting the textbook:

> However, in many cases one document serves as both a requirement specification and a system specification.

17. As system testing has a wider scope than acceptance testing, if system testing is adequately done, we need not do acceptance testing.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. The two have different purpose

Test Case Design

18. Grey-box test case design is a mixture of specification-based and implementation-based approaches.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. In other words, a mixture of black-box and white-box.

19. Consider this test case used to test a CLI program:

- Input: incorrect command
- Expected: error message

This is a negative test case.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. It test for error conditions.

20. Test case design can be divided into the following two categories:

- specification-based
- Responsibility-based

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. Both of these mean the same thing.

21. Consider a Java method `isPrime(int i)` that returns true if i is a prime number.

‘All non-int values’ is a possible EP for testing this method.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. As Java is strongly-typed, it is not even possible to use non-int values to test the method.

22. When deciding EPs of a method of an object, the state of the target object should be considered too.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As given in the textbook.

23. As per the EP technique, only one test case from each partition should be tested.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. There is no such restriction.

24. Equivalence partitioning can help make tests more efficient but not more effective.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It can improve both.

25. *Exploratory* testing is driven by observations during testing.

(1 mark) ✗

You scored 0 / 1 mark

True

○ False

General Comments

True. As per the textbook..

26. Consider a Java method isValidMonth(int m) that returns true if m is in the range [1..12] (both inclusive).

'Integers from 13 to highest possible int value' is an acceptable EP for testing this method.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. This is an example given in the textbook.

27. An *equivalence class* is a group of test inputs that are likely to be processed by the SUT in the same way.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the definition in the textbook.

28. More test cases is always better.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. We need to consider cost of testing as well.

24/28 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28

You have completed

Week 12 Quiz

You scored **16/32**

Have not received the confirmation Email? Click here to re-send.

Display chosen by module coordinator

SCROLL DOWN TO VIEW DETAILED RESULTS

Boundary Value Analysis

1. Boundary Value Analysis is based on the observation that bugs often occur near boundaries of equivalence partitions.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As given in the textbook.

2. BVA suggests that test inputs near boundaries are more likely to find bugs.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As given in the textbook.

3. It is possible that a SUT does not have clear boundary values.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. Some test inputs cannot be divided into partitions, and hence, do not have boundaries.

4. BVA suggests that testing one boundary value per partition is enough.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. It does not specify how many test inputs to use.

Combining Test Inputs

5. As per the textbook, at least once and mix and match are two strategies used when combining test inputs.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. The textbook does not mention a strategy called mix and match.

6. When an SUT takes multiple inputs, testing all possible combinations is effective but may not be efficient.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As given in the textbook.

Other QA Techniques

7. Formal verification can be used to prove the absence of errors. In contrast, testing can only prove the presence of error, not their absence.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

8. *Linters* are a subset of dynamic (not static) analysis tools.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. As per the textbook.

9. Code review is a QA activity.

(1 mark) ✗

You scored 0 / 1 mark

True

○ False

General Comments

True.

Quoting the textbook:

> While testing is the most common activity used in QA, there are other complementary techniques such as static analysis, code reviews, and formal verification.

10. A system crash is more likely to be a verification failure than a validation failure.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. A system crash is more likely to be a bug in the code, not in the requirements.

11. As per the article (i.e., 10 tips for reviewing code you don't like) given in the resources section of the textbook, it is not recommended to phrase your comments as questions.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. In fact it recommends doing exactly that.

12. Static analysis can find bugs.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. Quoting the textbook:
>Higher-end static analyzer tools can perform more complex analysis such as locating potential bugs, memory leaks, inefficient code structures etc.

13. Verification: are we building the right system?
Validation: are we building the system right?

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. It’s the other way around.

14.

As per the textbook, PR reviews is one of the two ways of reviewing code.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. There are three ways given.

15. IDEs have built-in static analysis capabilities.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. Quoting the textbook:
>Most modern IDEs come with some inbuilt static analysis capabilities. For example, an IDE can highlight unused variables as you type the code into the editor.

16. It is very important to clearly distinguish between validation and verification.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False.

Quoting the textbook:

> Whether something belongs under validation or verification is not that important. What is more important is both are done, instead of limiting to verification.

17. Formal methods, when compared to testing, is more expensive in general.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.

Quoting the textbook:

It requires highly specialized notations and knowledge which makes it an expensive technique to administer.

SDLC Process Models

This quiz also covers the first part of this topic covered earlier in the semester.

18. The agile processes value customer collaboration over contract negotiation.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

19. XP programmers constantly communicate with their customers.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

20. XP stands for eXperience programming.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It stands for eXtreme programming.

21. Scrum is designed to accommodate *requirements churn*.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True.

Quoting the textbook:

> A key principle of Scrum is its recognition that during a project the customers can change their minds about what they want and need (often called requirements churn), and that unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner.

22. eXtreme Programming (XP) is a well-known non-agile process while Scrum is a well-known agile process.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. Both are agile processes.

23. SDLC stands for Software Development Lifecycle Categories.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

False. It stands for Software Development Life Cycle

24. XP can be divided into twelve simple rules.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. Not twelve.

25. The daily scrum meeting is used as a mechanism for solving problems encountered by the project (or project members).

(1 mark) ✖

You scored 0 / 1 mark

○ True

False

General Comments

False.

Quoting the textbook:

> The daily scrum meeting is not used as a problem-solving or issue resolution meeting.

26. In the SCRUM process, iterations are called scrums.

(1 mark) ✖

You scored 0 / 1 mark

○ True

False

General Comments

False. They are called sprints.

27. The waterfall model can be either breadth-first or depth-first.

(1 mark) ✗

You scored 0 / 1 mark

[✓ True]

False

General Comments

False. That applies to the iterative model, not the waterfall model.

28. The *sequential* model is also called the *waterfall* model.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

29. In SCRUM, the *Scrum Master* represents the stakeholders and the business.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. It is the *Product Owner* who represents the stakeholders and the business.

30. The agile processes value working software over individuals and interactions.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. Both are valued.

31. The waterfall model is more suitable for projects that have well-understood and stable requirements as compared to fuzzy and evolving requirements.

(1 mark) ✓

You scored 1 / 1 mark

True

False

General Comments

True. As per the textbook.

32. Ideally, a daily scrum meeting is held at the end of the day, as it helps wrap up the day's work.

(1 mark) ✗

You scored 0 / 1 mark

○ True

False

General Comments

False. The morning is preferred.

16/32 QUESTIONS ANSWERED CORRECTLY

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31 32