



✓ **Congratulations! You passed!**
TO PASS 75% or higher

Keep Learning

GRADE
90%

Final Exam

LATEST SUBMISSION GRADE

90%

1. The first stage of the two-stage design process is _____ design.

1 / 1 point

Hint: This stage has activities like creating CRC cards, talking with the customer about their requirements, and creating mockups.

conceptual

✓ **Correct**

The correct answer is concept or conceptual design. This is the stage before technical design, where you will solicit customer requirements and use this to create a working conceptual design, using mockups and other early design techniques.

2. The second stage of the two-stage design process is _____ design.

1 / 1 point

Hint: This is when you will define the structure of the code and start turning your mockups into classes.

technical

✓ **Correct**

The correct answer is technical design. This is the stage at which the developers will start to turn the conceptual design into a more precise technical design. They could do this by using the UML design language, by specifying which methods will be coded for each class, etc.

3. Which of these conceptual design techniques will help you analyze the problem space to determine classes for your object-oriented software? **Choose the two correct answers.**

1 / 1 point

- ☐ tradeoffs
- ☐ requirements
- ☒ CRC

✓ **Correct**

Correct. CRC Cards will help you identify classes.

- ☒ mockups

✓ **Correct**

Correct. Mockups will help you visualize your problem space in the earliest stages.

4. During conceptual design, once the problem is mapped into components, what are the other two critical pieces of information that you must specify for these classes or components? **Choose the two correct answers.**

1 / 1 point

- ☒ collaborators

✓ **Correct**

Correct. Collaborators are other pieces of the software that your component will interact with to fulfill its function!

- ☒ responsibilities

✓ **Correct**

Correct. Responsibilities are what the component will do or keep track of.

- ☐ methods
- ☐ abstract data types

5. You are writing the CRC card for a Bear component. Choose the **two** responsibilities.

1 / 1 point

- ☒ hunger

✓ **Correct**

Correct. Hunger is not as obvious because it does not have a verb, but you can think of it like this: the bear component needs to keep track of its hunger.

- ☐ camper

- ☒ eat berries

✓ **Correct**

Correct. Eat berries is something bears are known to do.

- ☐ den

6. You are writing the CRC card for a Bear component. Choose the three collaborators.

1 / 1 point

☒ bear

✓ Correct

Correct. Objects can and often do interact with other objects of their class!

☐ guitar

☒ tree

✓ Correct

Correct. A tree is a component that a bear may interact with.

☐ computer

☒ den

✓ Correct

Correct. A den is a component a bear may interact with.

7. You create an object that represents a **user**, storing important information about them such as their preferences. What kind of object is this?

1 / 1 point

☐ boundary

☒ entity

☐ control

☐ client

✓ Correct

Correct! Entity objects often represent real-world objects.

8. You create an object that represents a **dialog box**. It creates buttons and text fields, etc. for the user to interact with, and it logs those interactions. What kind of object is this?

1 / 1 point

☐ display

☐ entity

☒ boundary

☐ control

☐ interaction

✓ Correct

Correct! This is a boundary object, because it interfaces with another system (the user)

9. You create an object that compares values from two different sources. It then updates the smaller value to be equal to the larger one. What kind of object is this?

1 / 1 point

☐ entity

☐ update

☒ control

☐ repository

✓ Correct

Correct! This is a control object, because it coordinates the activities of other objects.

10. Which of these is an example of a quality tradeoff?

1 / 1 point

☐ Limiting features knowing that they can be added later

☐ Adding preferences that allow users to switch some features on and off

☐ Not delivering key features so that deadlines can be met

☒ Adding security knowing it will reduce speed

✓ Correct

Correct. A tradeoff happens when to make an improvement you must sacrifice some other quality.

11. What is the term for reducing a class or object to its inputs and outputs in modelling?

1 / 1 point

☐ filter thinking

☐ pipe thinking

☒ black box thinking

☐ process thinking

✓ Correct

Correct! This is called black box thinking, because you don't care what happens inside at this point, only the inputs and outputs.

12. Which one of these classes is in most need of being decomposed?

1 / 1 point

☒ Store

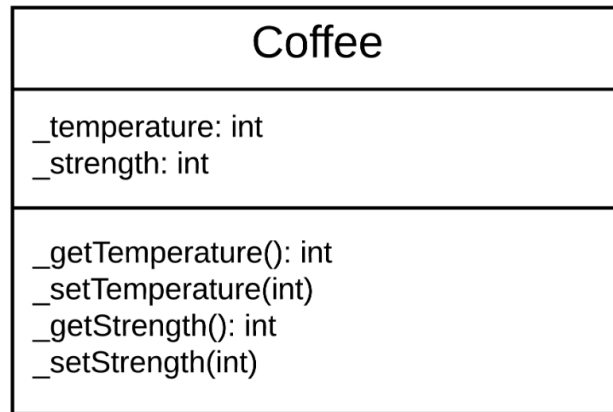
- ☐ Book
- ☐ Student
- ☐ Order

✓ Correct

Correct! A store has lots of responsibilities, including tracking orders, inventory, employees, customers, etc. This class needs to be decomposed.

13. In order to provide good encapsulation, fill-in-the-blanks on this UML class diagram: (Replace the underscores _ from top to bottom with minus signs ("-") or plus signs ("+"); your answer will be a string of six + or - signs with no spaces)

1 / 1 point



--++++

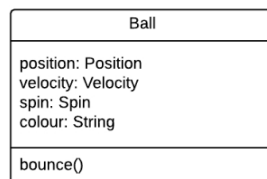
✓ Correct

The correct answer is "-+----". In other words, the variables are declared private so they cannot be seen or changed from outside the class. Instead, getter and setter methods are specified.

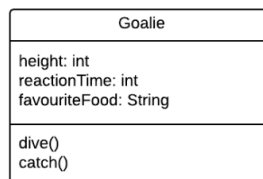
14. You are writing a simple soccer video game. Select the best example of proper abstraction:

1 / 1 point

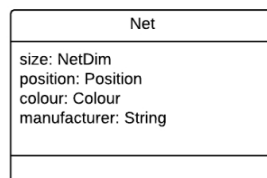
a)



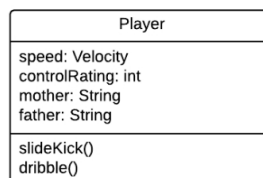
b)



c)



d)



- ☒ a)
- ☐ b)
- ☐ c)
- ☐ d)

✓ Correct

Correct! This class contains only details that are important in the context (a soccer video game).

15. Which design principle enables developers to follow the guideline D.R.Y. ("Don't Repeat Yourself"):

1 / 1 point

- ☐ encapsulation
- ☒ generalization

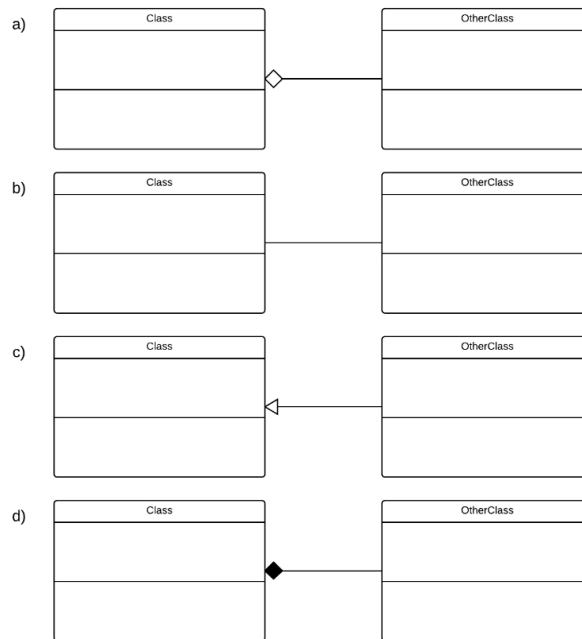
- ☐ abstraction
- ☐ decomposition

✓ Correct

Correct! Generalization (along with other object-oriented tools) allows developers to follow the D.R.Y. principle!

16. Which of these UML class diagrams shows an association relationship?

1 / 1 point



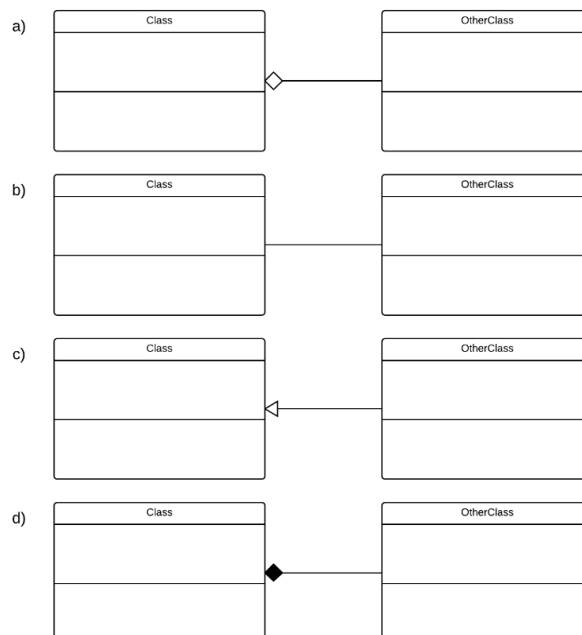
- ☐ a)
- ☒ b)
- ☐ c)
- ☐ d)

✓ Correct

Correct! A simple association relationship is shown with a plain line, often with numbers indicating how many of each object can be associated.

17. Which of these UML class diagrams depicts an aggregation ("has-a") relationship between the two classes?

1 / 1 point



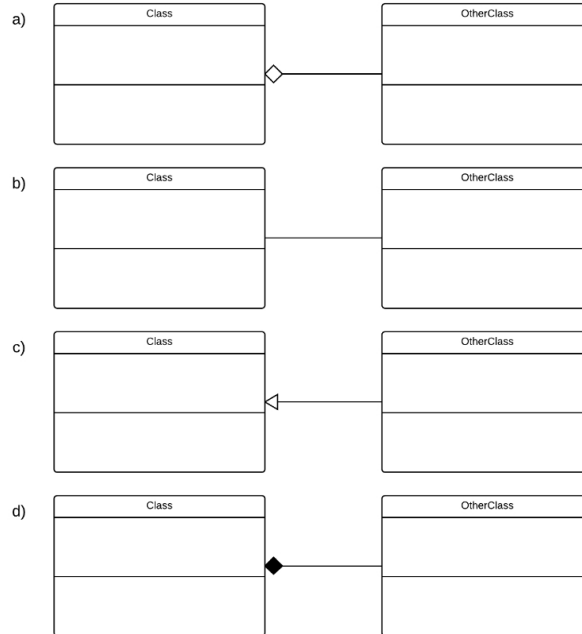
- ☒ a)
- ☐ b)
- ☐ c)
- ☐ d)

✓ Correct

Correct! An open diamond indicates a weak "has a" or aggregation relationship.

18. Which of these UML class diagrams depicts a composition, or a strong "has-a" relationship?

1 / 1 point



- ☐ a)
- ☐ b)
- ☐ c)
- ☒ d)

✓ Correct

Correct! A filled diamond indicates a composition - or strong "has a" - relationship.

19. Select the object pairing that has an **association** relationship:

1 / 1 point

- ☐ Tree - Root
- ☒ Hiker - Trail
- ☐ Coffee - Water
- ☐ Book - Page

✓ Correct

Correct! The hiker and trailer are associated but not dependent on each other.

20. Select the object pairing that has an **aggregation** relationship:

0 / 1 point

- ☐ Pie - Crust
- ☒ Car - Road
- ☐ Stapler - Staple
- ☐ Book - Page

! Incorrect

Incorrect. These two objects are more independent than in an aggregation relationship.

21. Select the object pairing that has a **composition** relationship:

1 / 1 point

- ☐ Bear - Forest
- ☐ Tea - Sugar
- ☐ Record Player- Record -
- ☒ Book - Page

✓ Correct

Correct. A book must have pages!

22. Choose the **two answers** that correctly complete the following sentence:

1 / 1 point

"We say that a class has low cohesion if..."

☒ ...It tries to encapsulate too many unrelated responsibilities.

✓ **Correct**

Correct. Cohesion is the degree to which a class is directed toward one purpose. Giving it unrelated responsibilities reduces cohesion.

☐ ...connects to many other classes.

☐ ...It does not have all the necessary parts, i.e. it is incomplete.

☒ ...its purpose is unclear.

✓ **Correct**

Correct. Cohesion is how well a class is directed toward a clear, singular purpose.

23. Two classes are tightly coupled. What are some ways you might be able to tell? **Choose the two correct answers.**

1 / 1 point

☒ In order to understand one class, you need to open up the other to look at the implementation

✓ **Correct**

Correct. This is usually a sign that the coupling is too tight; instead, the interfaces should be clear and interactions limited.

☐ They can easily be swapped with different implementations of the same class

☐ Their interactions are limited and controlled

☒ They are very highly reliant on each other

✓ **Correct**

Correct. Coupling refers to how deeply integrated different components are. Tight coupling means the components are deeply integrated, which is not desirable because it makes it more difficult to make changes.

24. How can you apply the principle of Separation of Concerns in object-oriented programming?

0 / 1 point

- ☐ Separate objects or components according to their role in the software
- ☒ Separate data and actions (methods) into different classes
- ☐ Split developers into teams that each deal with different parts of the software
- ☐ Ensure classes are only concerned with their own data

! **Incorrect**

Incorrect. In fact, data and methods to manipulate that data are usually part of the same class.

25. Which of these violates **Liskov's Substitution Principle**?

1 / 1 point

- ☐ the subclass adds behaviour
- ☐ the superclass is too general
- ☒ an operation in the superclass is replaced by a different operation in the subclass
- ☐ subclasses specify the abstract methods of the superclass

✓ **Correct**

Correct! This directly violates Liskov's substitution principle, which is a useful test to identify poor uses of inheritance.

26. For which of these situations would you use a sequence diagram?

1 / 1 point

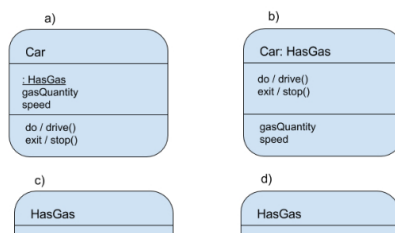
- ☒ To show the collaborative behaviour of objects in your program.
- ☐ To show the relationship between classes
- ☐ To show all of the different processes of your program.
- ☐ To show the different modes that your program can be in.

✓ **Correct**

Correct! This is the best use of a sequence diagram.

27. Choose the correct state diagram for a car which has a state called "HasGas:"

1 / 1 point









- ☐ a)
- ☐ b)
- ☒ c)
- ☐ d)

✓ Correct

Correct! The state goes at the top, variables in the middle, and activities (including exit and entry activities) in the bottom.

28. Which of these elements represents a termination in a UML State diagram?

1 / 1 point

- a)  b) 
- c)  d) 

- ☒ a)
- ☐ b)
- ☐ c)
- ☐ d)

✓ Correct

Correct! This represents a termination

29. What is the purpose of model checking?

0 / 1 point

- ☐ To check the software for errors before release
- ☐ To test for user-reported bugs
- ☒ To verify that the technical implementation matches conceptual mockups
- ☐ To verify that the conceptual model of your software matches the customer's requirements.

! Incorrect

Incorrect. The technical implementation cannot be programmatically tested against conceptual mockups, which are informal representations.

30. What is an abstract data type?

1 / 1 point

- ☐ variables that are assigned a type (i.e. integer, double) but does not yet have a value assigned.
- ☒ a type of data defined by the developer rather than the language.
- ☐ a data type that cannot be used directly but must be implemented as an interface
- ☐ a data-centric class

✓ Correct

Correct! Abstract data types are structured by the developer. They eventually evolved into classes.