Congratulations! You passed!
Grade received 100%Latest Submission Grade 100%To pass 75% or higher
Go to next item

1. 1 / 1 point

Some of the earliest computing languages supported:

- (●) main program and subroutines
- ( ) objects and classes
- ( ) abstract data types
- ( ) local variables

Correct
Correct. Early languages supported software that was a tree of routines.

2. What are some advantages of object-oriented programming in a language like Java? Choose the three correct answers. 1 / 1 point

- [x] abstract data types

Correct
Correct! Abstract data types are implemented in natural manner through classes.

- [x] mimic the real-world structure of the problem

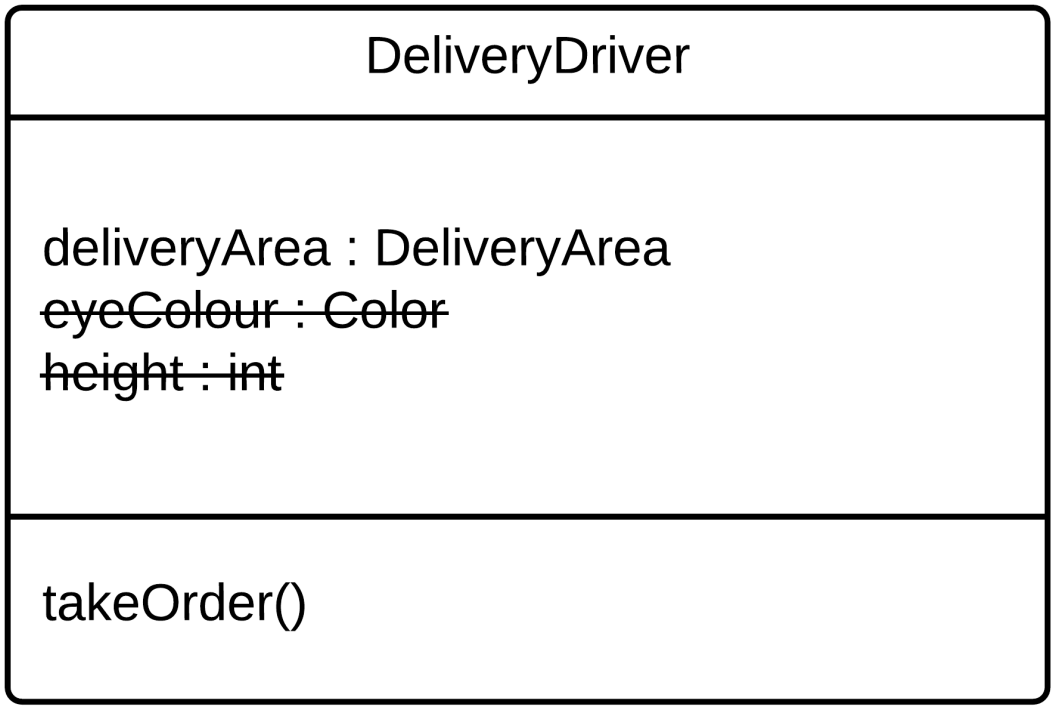
Correct
Correct! In object oriented languages, it can be simpler to represent the problem space.

- [x] data management

Correct
Correct. Access to attributes of objects can be controlled.

- [ ] computing efficiency

3. Sam was asked to create a DeliveryDriver class. Sam thought about the problem, and reduced it to its most essential aspects… things like takeOrder, DeliveryArea, etc. She ignored things that were not important in the context, like the driver's height or eye colour. She just applied an important object-oriented design principle. Which of these concepts best describes what she just did? 1 / 1 point



- generalization
- decomposition
- abstraction
- encapsulation

Correct
Correct! Abstraction is bringing an object into the software by identifying its most important aspects and eliminating those that are unnecessary in the context.

4. Sam identified the important attributes and behaviours of a delivery driver and put them all into a DeliveryDriver class, like "takeOrder", "deliverOrder", and "DeliveryArea." She exposed some of these to other classes. She just demonstrated an important part of object-oriented design. Which of these concepts best describes what you did? 1 / 1 point

- decomposition
- generalization
- abstraction
- encapsulation

Correct
Correct! A well-structured class should only expose those variables and methods that the developer intends, hiding away any information that outside classes do not need. This is called encapsulation.

5. Sam decided that the DeliveryDriver class was getting too complex, so she split it up, moving its behaviour into several related classes, like DeliveryCar and DeliveryOrder. She just demonstrated which important object-oriented design principle? 1 / 1 point

- ○ abstraction
- ◉ decomposition
- ○ encapsulation
- ○ generalization

✓ Correct
Correct! If an object is too complex, it can be divided into several classes with related functionality. These classes can still work together. This is called decomposition.

6. Sam realized that her DeliveryDriver class had some behaviour and attributes that could be shared by other classes, like BusDriver and TaxiDriver, so she made a class called Driver that these classes inherited behaviour from. Which object-oriented design principle did she use? 1 / 1 point

- ○ decomposition
- ○ abstraction
- ◉ generalization
- ○ encapsulation

✓ Correct
Correct! As its name suggests, generalization involves separating out some general behaviour. Then this behaviour can be shared, for example, through inheritance.

7. Which keywords allow your classes to achieve polymorphism in Java? Choose the two correct answers. 1 / 1 point

- ☑ extends

✓ Correct
Correct! The is the keyword for implementation inheritance in Java.

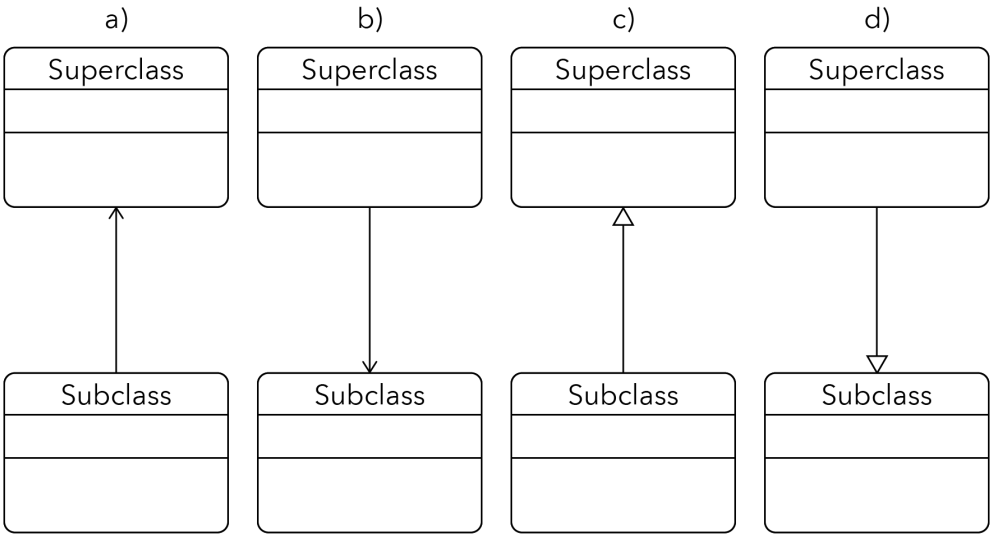
- ☐ interfaces
- ☑ implements

✓ Correct
Correct! This is the keyword for interface inheritance in Java.

- ☐ inherits

8. 1 / 1 point

Which is the proper way to show inheritance in a UML class diagram? Choose one of the four options below:



- a)
- b)
- c) ✓
- d)

Correct
Correct! Inheritance is shown in UML with an open arrowhead pointing to the superclass.

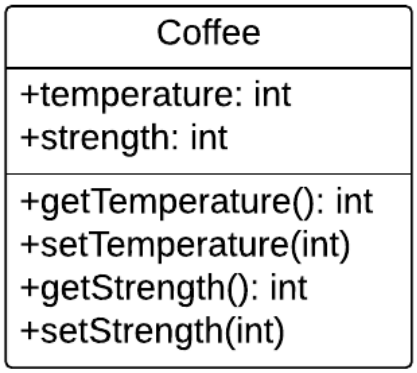
9. Which is the proper way to show an abstract method in a UML class diagram? 1 / 1 point

- *anOperation()* ✓
- #anOperation()
- «anOperation()»
- abstract anOperation()

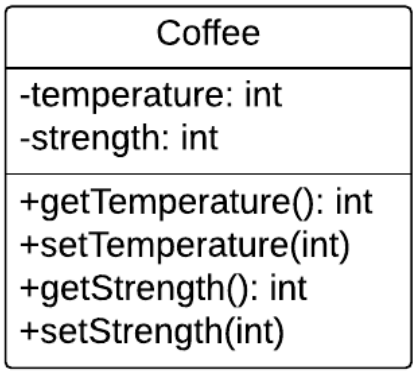
Correct
Correct! An abstract method, variable, or class name is denoted in UML with italics.

10. Which UML class diagram is a good example of encapsulation? Choose one of the four options below: 1 / 1 point

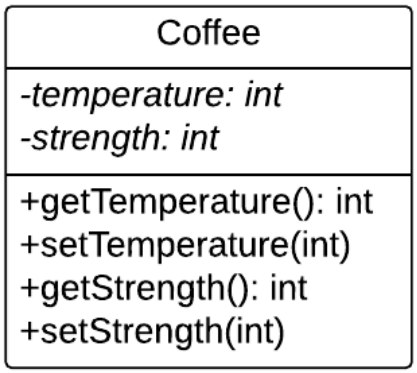
a)



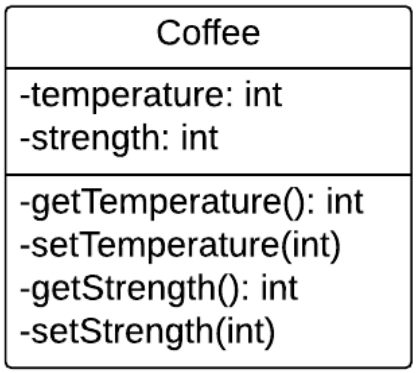
b)



c)



d)



- ○ a)
- ◉ b)
- ○ c)
- ○ d)

✓ Correct
Correct! This class has hidden the variables, then used getters and setters to allow access to them (and potentially add gatekeeping or value checking).

11. 1 / 1 point

Add the necessary keyword to complete this instance variable declaration in a class if you are using the encapsulation design principle to hide this variable from all other classes:
_____ String message;

private

Correct

Correct answer is 'private'. This String should be private, and if other classes do need to change or read this variable, you can specify a setter or a getter method.

12. There are three different relationships that objects can have. Which of these best describes the relationship between a Chair and its Legs? 1 / 1 point

- association
- aggregation
- composition
- formation

Correct
Legs are a crucial part of a chair being a chair. This is a strong "has-a" relationship, so it is composition.