Module 7 quiz on inheritance

Quiz, 5 questions

5/5 points (100.00%)

Congratulations! You passed!

Next Item

1 / 1 point

1.
Consider a class Student whose state includes name and age, Joanne who is a Student, and PartTime whose state includes all of the fields of Student and also maximumHours.

- Student is a _____
- Joanne is a _____
- PartTime is a _____

- sub class
- Student object
- super class

- super class
- client
- subclass

- super class
- Student object
- subclass

Correct

- super class
- Student object
- inherited class

Module 7 quiz on inheritance

Quiz, 5 questions

5/5 points (100.00%)

1 / 1 point

2.
When creating a subclass

- write both the inherited and the additional state and behavior of the subclass.
- only write the methods that are public.
- only write the instance variables and methods that are inherited by the subclass.
- only write the instance variables and methods that are added to the subclass.

Correct

---

1 / 1 point

3.
Why do we aim to minimize the amount of code we have to write by not repeating code segments within a Java project? *Check all that apply*

- [x] It makes projects unique. An object that contains all of its state and behavior in a single file is not easily reused.

Un-selected is correct

- [x] It makes debugging easier. Implementation details that only appear once in a project can be corrected or improved by editing in only one place.

Correct

- [x] It makes a program run faster.

Un-selected is correct

Module 7 quiz on inheritance

Quiz, 5 questions

5/5 points (100.00%)

- [x] It facilitates code reuse. An object that contains all of its state and behavior in a single file is "portable" and can be included in many different Java projects.

Correct

- [x] It improves readability, code is easier to trace when implementation details are found in only on location.

Correct

---

1 / 1 point

4.
To make a call from a subclass to a public, overloaded method whose implementation details are in its super class,

- you must precede the call with the keyword **super** to indicate that the implementation details are in the super class.
- simply make the call. The unique return type and/or parameter list will enable the compiler to locate the implementation details.

Correct

- simply make the call. The implementation details found in the subclass will automatically be used.
- you must precede the call with the keyword **sub** to indicate that you are writing the method in the subclass.

---

1 / 1 point

5.

Module 7 quiz on inheritance

Quiz, 5 questions

5/5 points (100.00%)

To make a call from a subclass to a public, overridden method whose implementation details are in its super class

- A subclass can not call a method in its super class.
- simply make the call. The unique return type and/or parameter list will allow the compiler to locate the implementation details.
- you must precede the call with the keyword **super** to indicate which version of the method you are calling.

Correct

- you must precede the call with the keyword **this** to indicate which version of the method you are calling.