Module 7 quiz on inheritance

LATEST SUBMISSION GRADE

100%

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. **1 / 1 point**

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

- ○ • super class
  - Student object
  - inherited class
- ◉ • super class
  - Student object
  - subclass
- ○ • super class
  - client
  - subclass
- ○ • sub class
  - Student object
  - super class

✓ **Correct**

2. When creating a subclass **1 / 1 point**

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

✓ **Correct**

3. Why do we aim to minimize the amount of code we have to write by not repeating code segments within a Java project? **1 / 1 point**
Check all that apply

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

  ✓ **Correct**

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

4. To make a call from a subclass to a public, overloaded method whose implementation details are in its super class, **1 / 1 point**

- ○ you must precede the call with the keyword **sub** to indicate that you are writing the method in the subclass.
- ◉ simply make the call. The unique return type and/or parameter list will enable the compiler to locate the implementation details.
- ○ simply make the call. The implementation details found in the subclass will automatically be used.
- ○ you must precede the call with the keyword **super** to indicate that the implementation details are in the super class.

✓ **Correct**

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

1 / 1 point

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

✓ **Correct**