Reminder about the terminology of inheritance.

- A class which extends another class is called a subclass, child class, or derived class.
- Such classes inherit all public attributes and methods from the super class, and can override or add new attributes or methods.
- Such classes can access the attributes and methods in the super class, as well as their own.
- A class which is extended by another class is called a super class, parent class, or base class.
- Such classes have no knowledge of (or access to) the attributes and methods in the subclass.

Unified Modeling Language (UML) diagrams show inheritance relationships.

- B is a subclass of A.
- A is the super class of B.
- C is a subclass of B.
- B is the super class of C.
- No multiple inheritance!

Which statement about inheritance in Java is correct?

Please select the single correct answer.
Inheritance is ...

A. when a class accesses variables or methods from another class.
B. when a class implements an interface.
C. when a subclass extends a super class.
D. when a super class extends a subclass.
E. when Java inherits features from another language, e.g. C++.

What is the main purpose of inheritance in the Java programming language?

A. To allow programmers to easily duplicate (copy) code and data from an existing class to a new class.
B. To create relationships in which classes share some attributes and code, but define their own specialized behavior.
C. To make complicated relationships between classes that are difficult to understand.
D. To specify consistent attributes and behavior throughout a set of related classes.

When a subclass extends an abstract class, which methods must be overridden?

Please select the single correct answer.

A. All concrete and abstract methods must be overridden.
B. Only the concrete methods must be overridden.
C. Only the abstract methods must be overridden.
D. No concrete or abstract methods have to be overridden.
E. Any number of concrete and abstract methods can be overridden.
Which is the best explanation of what it means to override a method?

A. A super class overrides a method in a subclass by writing a new implementation with the same signature.
B. A subclass overrides a method in a super class by writing a new implementation with the same signature.
C. A super class overrides a method in a subclass by writing a modified implementation with different parameters.
D. A subclass overrides a method in a super class by writing a modified implementation with different parameters.

Which is the best explanation of what it means to overload a method?

A. Only a subclass is allowed to overload a super class method, by adding a method of the same name but different signature.
B. Only a super class is allowed to overload a subclass method, by adding a method of the same name but different signature.
C. Any class can overload any method by adding a new method of the same name with different parameters.
D. Any class can extend the interface it implements by adding methods of the same name with different signatures.

Which statement is correct about an attribute or method that is overridden?

A. When a derived class overrides an attribute or method in a base class, the base class can no longer access the base class version.
B. When a derived class overrides an attribute or method, the derived class can no longer access the base class version.
C. When a derived class overrides an attribute or method, the base and derived classes can still access the base version.
D. When a derived class overrides an attribute or method, the base and derived classes can access both versions.

What is the difference between private and public method inheritance?

A. Derived classes can only inherit public methods, not private.
B. Derived classes can inherit private or public methods.
C. Derived classes can only inherit private methods, not public.
D. Public versus private makes no difference w.r.t. inheritance.
E. None of the above.

Which of the following statements about constructor chaining are correct?

1. Implicit constructor chaining is done by the Java compiler for default constructors.
2. Explicit constructor chaining must be done by the programmer if the programmer wants to invoke an overloaded constructor (give it arguments).
3. Constructor chaining is optional, a programmer can choose to initialize all attributes instead.
4. Constructor chaining requires each subclass to call its superclass constructor.

A. 1, 2, 3
B. 2, 3, 4
C. 1, 2, 4
D. 1, 3, 4