Chap 10 Questions

Multiple Choice

1. When one object is a specialized version of another object, there is an \_\_\_\_\_ between them. **Answer: A. “is a” relationship.**
2. In an inheritance relationship, the \_\_\_\_\_ is the general class. **Answer: B. Base class.**
3. In an inheritance relationship, the \_\_\_\_\_ is the specialized class. **Answer: C. derived class.**
4. Base classes are sometimes called \_\_\_\_\_. **Answer:** **D. Superclasses.**
5. Derived classes are sometimes called \_\_\_\_\_. **Answer: B. Subclasses.**
6. The \_\_\_\_\_ refers to the base class. **Answer: C. base keyword.**
7. The term \_\_\_\_\_ refers to an object’s ability to take different forms. **Answer: C. Polymorphism.**
8. When a derived class method has the same name as a base class method, it is often said that the derived class method \_\_\_\_\_ the base class method. **Answer: C. Overrides.**
9. The \_\_\_\_\_ declares that a derived class is allowed to override a method.

**Answer: D. virtual keyword.**

1. The \_\_\_\_\_ declares that this method overrides a method in the base class.

**Answer: A. override keyword.**

1. A class that is not intended to be instantiated, but used only as a base class, is called a(n) \_\_\_\_\_. **Answer: D. Abstract class.**
2. To declare a class as abstract, you use the \_\_\_\_\_ in the class header.

**Answer: A. abstract keyword.**

1. A regular, non-abstract class is sometimes called a \_\_\_\_\_. **Answer: C. Concrete class.**
2. A(n) \_\_\_\_\_ is a method that appears in a base class but expects to be overridden in a derived class. **Answer: A. Abstract method.**
3. A(n) \_\_\_\_\_ is a property that appears in a base class but expects to be overridden in a derived class. **Answer: D. Abstract property.**
4. \_\_\_\_\_ allows a base class reference variable to reference a derived class object.

**Answer: B. Inheritance.**

True or False

1. The base class inherits fields, properties, and methods from the derived class. **Answer: False.**
2. Polymorphism allows a class variable of the base class type to reference objects of either the base class or the derived class types. **Answer: False**
3. Properties in a base class cannot be overridden in the same way that methods can be overridden. **Answer: False.**
4. A base class reference variable can reference an object of any class that is derived from the base class. **Answer: True.**
5. A statement that tries to use the new operator to instantiate an abstract class will not compile. **Answer: True.**
6. A class that is not intended to be instantiated, but used only as a base class, is called a concrete class. **Answer: False.**
7. When an abstract property appears in a class, it must be overridden in any class that is derived from the class. Answer:

Short Answer

1. What does a derived class inherit from its base class? **Answer: A derived class inherits the base class’s methods, properties, and fields.**
2. Look at the following code, which is the first line of a class declaration. What is the name of the base class? What is the name of the derived class? Code: ”*class Tiger : Felis ”*

**Answer: “Felis” is the base class, and “Tiger” is the derived class.**

1. Can methods in the derived class directly access the base class’s private members? **Answer: No, they cannot. The derived class can access public properties of the base class though, which could access the base class’s private members.**
2. When you create an instance of a derived class, which constructor is called first? **Answer: The base class constructer is called first, obviously followed by the derived class’s constructer.**
3. In what kind of situation would you want to use an abstract class instead of a base class? **Answer: If every derived class has to have certain characteristics, but the specifics of those characteristics are going to vary for pretty much every single derived class. An example would be a base “Airplane” class and a bunch of derived classes for types of airplanes. Every airplane has a wingspan, which could show up in the base, abstract Airplane class, but said wingspan will vary most of the time.**
4. What is the primary difference between an abstract class and a regular class? **Answer: The main difference is that an abstract class can NOT be instantiated, while a regular class obviously can.**
5. Can abstract classes also contain abstract properties? **Answer: Yes they can; this will ensure that the abstract class will definitely have certain properties set in place for derived classes to inherit, without giving any details on those properties.**