CS 1331 Introduction to Object-Oriented Programming: Abstract classes and interfaces

1. What is an abstract class?

A. A class that cannot be instantiated
B. A class that can only be subclassed
C. A class that can only be used as an interface
D. A class that can be used as an interface or a superclass

2. What is an interface?

A. A class that cannot be instantiated
B. A class that can only be subclassed
C. A class that can only be used as an interface
D. A class that can be used as an interface or a superclass

3. What is the difference between an abstract class and an interface?

A. An abstract class cannot be instantiated, while an interface can
B. An abstract class can only be subclassed, while an interface can be implemented
C. An abstract class can only be used as an interface, while an interface can be implemented
D. An abstract class can be used as an interface or a superclass, while an interface can only be implemented

4. Which of the following is true about abstract classes?

A. Abstract classes cannot be instantiated
B. Abstract classes can only be subclassed
C. Abstract classes can only be used as interfaces
D. Abstract classes can be used as interfaces or superclasses

5. Which of the following is true about interfaces?

A. Interfaces cannot be instantiated
B. Interfaces can only be subclassed
C. Interfaces can only be used as interfaces
D. Interfaces can be used as interfaces or superclasses

6. Which of the following is not a benefit of using abstract classes?

A. Abstract classes can provide a default implementation of an interface
B. Abstract classes can be used to group related classes
C. Abstract classes can be used to hide implementation details
D. Abstract classes can be used to enforce object-oriented principles

7. Which of the following is not a benefit of using interfaces?

A. Interfaces can be used to group related classes
B. Interfaces can be used to hide implementation details
C. Interfaces can be used to enforce object-oriented principles
D. Interfaces can be used to define contracts between classes

8. Which of the following is not a disadvantage of using abstract classes?

A. Abstract classes can be difficult to understand
B. Abstract classes can be difficult to maintain
C. Abstract classes can be difficult to test
D. Abstract classes can be inflexible

9. Which of the following is not a disadvantage of using interfaces?

A. Interfaces can be difficult to understand
B. Interfaces can be difficult to maintain
C. Interfaces can be difficult to test
D. Interfaces can be inflexible

10. Which of the following is the best reason to use an abstract class?

A. To provide a default implementation of an interface
B. To group related classes
C. To hide implementation details
D. To enforce object-oriented principles

11. Which of the following is the best reason to use an interface?

A. To group related classes
B. To hide implementation details
C. To enforce object-oriented principles
D. To define contracts between classes

12. Which of the following is not a reason to use abstract classes?

A. To provide a default implementation of an interface
B. To group related classes
C. To hide implementation details
D. To enforce object-oriented principles

13. Which of the following is not a reason to use interfaces?

A. To group related classes
B. To hide implementation details
C. To enforce object-oriented principles
D. To define contracts between classes

14. What is the best reason to use an abstract class?

A. To provide a default implementation of an interface
B. To group related classes
C. To hide implementation details
D. To enforce object-oriented principles

15. What is the best reason to use an interface?

A. To group related classes
B. To hide implementation details
C. To enforce object-oriented principles
D. To define contracts between classes

Answer Key: 1-D, 2-C, 3-D, 4-D, 5-D, 6-D, 7-D, 8-A, 9-A, 10-A, 11-D, 12-A, 13-A, 14-A, 15-D