Review Test Submission: Quiz 3

User	Doung Lan Cheung
Course	CIS.351.M001.SPRING20.Data Structures
Test	Quiz 3
Started	2/27/20 11:24 PM
Submitted	2/27/20 11:25 PM
Due Date	2/29/20 11:59 PM
Status	Completed
Attempt Score	100 out of 100 points
Time Elapsed	0 minute out of 25 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers, Feedback, Incorrectly Answered Questions

Question 1

10 out of 10 points

Which of the following statements is true?

Selected Answer: Abstract classes may include both abstract and non-abstract methods.

Answers: All methods in an abstract class must also be abstract.

Abstract classes may be instantiated, but it is only possible to call non-abstract methods of those classes.

Abstract classes may include both abstract and non-abstract methods.

Abstract classes may not include instance variables.

Question 2

10 out of 10 points

How do we indicate in a UML diagram that a class is abstract?

Selected Answer: The class name is in italics.

Answers: The word "abstract" is placed in parentheses after the class name.

The boundary of the UML box is a dotted line instead of a solid line.

The class name is bold.

The class name is in italics.

Question 3

5 out of 5 points

Which of the following statements is true?

Selected Answer: Abstract classes may have constructors, even though they cannot be instantiated.

Answers: Abstract classes don't have constructors, because they cannot be instantiated.

Abstract classes may have constructors, even though they cannot be instantiated.

Abstract classes have abstract constructors.

Question 4

5 out of 5 points

Which of the following best describes an interface in Java?

Selected Answer: An interface is similar to an abstract class that has only abstract methods.

Answers: An interface is similar to an abstract class that has only abstract methods.

An interface is a specialized class used to connect two other classes.

Interface is the default access specifier. It applies whenever an instance variable or method is NOT declared to be public, private or protected.

The interface keyword is used to indicate that one class is a subclass of another.

Question 5

5 out of 5 points

Which of the following is true?

Selected Answer: A class may implement any number of interfaces.

Answers: A class may inherit from a superclass or implement an interface, but not both.

A class may implement at most one interface.

A class may implement any number of interfaces.

One class may inherit from multiple superclasses.

Question 6

5 out of 5 points

Assuming that A is an interface, which of the following is true?

Selected Answer: It is possible to declare a variable of type A.

Answers:
- It is possible to instantiate an object of type A.
- It is possible to declare a variable of type A.
- Both are true.
- Neither are true.

Question 7

5 out of 5 points

Which of the following Java keywords are most closely related to exception handling?

Selected Answer: try, catch

Answers:
- attempt, except
- try, catch
- static, final
- first, last

Question 8

5 out of 5 points

In a try ... catch ... finally statement,

Selected Answer: The finally block is always executed.

Answers:
- The finally block is always executed.

  The finally block is only executed if an exception is raised that does not have a catch block.
- The finally block is only executed if no exception is raised.
- The finally block will never be executed.

Question 9

5 out of 5 points

What is the purpose of the throws keyword?

Selected

Answer: It informs the compiler that the method may throw the indicated exception type.

Answers:
- It informs the compiler that the method may throw the indicated exception type.
- It instantiates and throws, an exception object.
- throws is not a Java keyword.

The throws keyword indicates to the Java compiler that the method will save information to disk.

Question 10

5 out of 5 points

A method must always catch any exception that is raised when that method is executed.

Selected Answer: False

Answers:
- True
- False

Question 11

5 out of 5 points

Which of the following accurately describe the relationship between a subclass and a superclass.

Check all that are true.

Selected Answers:
- The subclass is a specialized version of the superclass.

  There is an "is a" relationship between the subclass and the superclass. An instance of the subclass "is a" instance of the superclass as well.

Answers:
- The subclass is a specialized version of the superclass.
- The superclass is a specialized version of the subclass.

  There is an "is a" relationship between the subclass and the superclass. An instance of the subclass "is a" instance of the superclass as well.

  There is an "is a" relationship between the superclass and the subclass. An instance of the superclass "is a" instance of the subclass as well.

Question 12

5 out of 5 points

Which of the following is true?

Selected Answer: The subclass inherits, and may access, all public members of the superclass, except the constructor.

Answers:

The superclass inherits, and may access, all public members of the subclass, except the constructor.

The subclass inherits, and may access, all public members of the superclass, except the constructor.

The superclass inherits, and may access, all members (public and private) of the subclass, except the constructor.

The subclass inherits, and may access, all members (public and private) of the superclass, except the constructor.

Question 13 5 out of 5 points

Which of the following is true of superclass constructors? Check all that are true.

Selected Answers: A superclass constructor is always called before the subclass constructor.

The "super" keyword may be used to explicitly call a superclass constructor from the subclass.

Answers: A superclass constructor is always called before the subclass constructor.

A subclass constructor is always called before the superclass constructor.

The "super" keyword may be used to explicitly call a superclass constructor from the subclass.

The "sub" keyword may be used to explicitly call a subclass constructor from the superclass.

It is not possible to explicitly call a superclass constructor from the subclass.

Question 14 5 out of 5 points

Which of the following best describes the difference between overloading and overriding a method?

Selected Answer: When one method overrides another, both methods will have the same signature.

Answers: There is no difference. They mean the same thing.

When one method overrides another, both methods will have the same signature.

When one method overloads another, both methods will have the same signature.

Overloading only makes sense in the context of inheritance.

Question 15 5 out of 5 points

What is the role of the protected keyword in Java?

Selected Answer: It is an access specifier that provides a level of access somewhere between public and private.

Answers: It indicates that the contents of a variable cannot be modified.

It is an access specifier that provides a level of access somewhere between public and private.

It indicates that a particular class may not be subclassed.

Question 16 5 out of 5 points

Which of the following statements is true?

Selected Answer: A reference variable my hold object references of non-matching types, as long as the object type is a subclass of the variable type.

Answers:

A reference variable may only hold an object reference if there is an exact match between the type of the object and the type of the variable.

A reference variable my hold object references of non-matching types, as long as the object type is a subclass of the variable type.

A reference variable my hold object references of non-matching types, as long as the object type is a superclass of the variable type.

A reference variable my hold object references of any reference type, though a cast may be required.

Question 17 5 out of 5 points

Which of the following is described as “dynamic binding”?

Selected Answer: The JVM dynamically determines which method to call based on the object's type at run-time.

Answers: The contents of variables may change dynamically at run-time.

An object may dynamically change its superclass at run-time.

The JVM dynamically determines which method to call based on the object's type at run-time.

The JVM dynamically determines which method to call based on the variable's type at run-time.

Question 18

5 out of 5 points

Assume that class A does not define a toString method, and does not explicitly extend any other class. What will happen if we call toString on an object of class A?

Selected Answer: All classes implicitly inherit from Object, so the toString method defined by Object will be called.

Answers: The code will not compile because the method is not defined.

All classes implicitly inherit from Super, so the toString method defined by Super will be called.

All classes implicitly inherit from Object, so the toString method defined by Object will be called.

Every Java class has a default toString method that prints the contents of the instance variables. That method will be executed.

Tuesday, May 5, 2020 1:03:55 AM EDT

← OK