Review Test Submission: Homework 1

User	Brittany Sifford
Course	Java Programming (Intermediate) (92759)
Test	Homework 1
Started	1/23/24 4:53 PM
Submitted	1/23/24 6:31 PM
Due Date	2/7/24 11:59 PM
Status	Completed
Attempt Score	36 out of 36 points
Time Elapsed	1 hour, 37 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

1 out of 1 points

Using public *set* methods helps provide data integrity if:

Selected Answer: Both b and c.

Answers:
- The instance variables are public.
- The instance variables are private.
- The methods perform validity checking.
- Both b and c.

Question 2

1 out of 1 points

The static method _____ of class String returns a formatted String.

Selected Answer: format.

Answers:
- printf.
- format.
- formatString.
- toFormatedString.

Question 3

1 out of 1 points

Review Test Submission: Homework 2

User	Brittany Sifford
Course	Java Programming (Intermediate) (92759)
Test	Homework 2
Started	1/29/24 10:29 AM
Submitted	1/29/24 11:26 AM
Due Date	2/15/24 11:59 PM
Status	Completed
Attempt Score	30 out of 30 points
Time Elapsed	56 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

1 out of 1 points

Which of the following statements is *false*?

Selected Answer: Anonymous methods provide a shorthand notation for creating lambdas.

Answers:

As of Java SE 8, any interface containing only one method is known as a functional interface.

There are many functional interfaces throughout the Java APIs.

Functional interfaces are used extensively with Java SE 8's new lambda capabilities.

Anonymous methods provide a shorthand notation for creating lambdas.

Question 2

1 out of 1 points

A class that implements an interface but does not declare all of the interface's methods must be declared _____.

Selected Answer: abstract.

Answers: public.

interface.

abstract.

final.

Question 3

1 out of 1 points

Interfaces can have _____ methods.

Selected Answer: any number of

Answers:
0
1
2
any number of

Question 4

1 out of 1 points

Which of the following statements about abstract superclasses is *true*?

Selected Answer: abstract superclasses may contain data.

Answers:
abstract superclasses may contain data.

abstract superclasses may *not* contain implementations of methods.

abstract superclasses must declare all methods as abstract.

abstract superclasses must declare *all* data members not given values as abstract.

Question 5

1 out of 1 points

In Java SE 7 and earlier, an interface may contain:

Selected Answer: public static final data and public abstract methods.

Answers:

private
static
data
and public
abstract
methods.

only public
abstract
methods.

public
static
final
data and public
abstract
methods.

private
static
data
and public
final
methods.

Question 6

1 out of 1 points

For which of the following would polymorphism *not* provide a clean solution?

Selected Answer: A program to compute a 5% savings account interest for a variety of clients.

Answers:

A billing program where there is a variety of client types that are billed with different fee structures.

A maintenance log program where data for a variety of types of machines is collected and maintenance schedules are produced for each machine based on the data collected.

A program to compute a 5% savings account interest for a variety of clients.

An IRS program that maintains information on a variety of taxpayers and determines who to audit based on criteria for classes of taxpayers.

Question 7

1 out of 1 points

Non-abstract classes are called _____.

Selected Answer: concrete classes.

Answers:

real classes.

instance classes.

implementable classes.

concrete classes.

Question 8

1 out of 1 points

Which of the following statements is *false*?

Selected Answer: An advantage of inheritance over interfaces is that only inheritance provides the *is-a* relationship.

Answers: An advantage of inheritance over interfaces is that only inheritance provides the *is-a* relationship.

Objects of any subclass of a class that implements an interface can also be thought of as objects of that interface type.

When a method parameter is declared with a subclass or interface type, the method processes the object passed as an argument polymorphically.

All objects have the methods of class Object.

Question 9

1 out of 1 points

Consider the abstract superclass below:

```
public abstract class Foo
{
  private int a;
  public int b;

  public Foo(int aVal, int bVal)
  {
    a = aVal;
    b = bVal;
  }

  public abstract int calculate();
}
```

Any *concrete* subclass that *extends* class Foo:

Selected Answer: Both (a) and (b).

Answers:

Must implement a method called calculate.

Will *not* be able to access the instance variable a.

Neither (a) nor (b).

Both (a) and (b).

Question 10

1 out of 1 points

Polymorphism allows for specifics to be dealt with during:

Selected Answer: execution.

Answers:
- execution.
- compilation.
- programming.
- debugging.

Question 11

1 out of 1 points

Which interface is used to identify classes whose objects can be written to or read from some type of storage or transmitted across a network?

Selected Answer: Serializable

Answers:
- Comparable
- Runnable
- AutoCloseable
- Serializable

Question 12

1 out of 1 points

Consider classes A, B and C, where A
is an abstract superclass, B
is a concrete class that inherits from A and C is a concrete class that inherits from B.
Class A
declares abstract method originalMethod, implemented in class B. Which of the following statements is *true* of class C?

Selected Answer: None of the above.

Answers:
- Method originalMethod
cannot be overridden in class C—once it has been implemented in concrete class B, it is implicitly final.
- Method originalMethod
must be overridden in class C, or a compilation error will occur.

If method originalMethod is not overridden in class C but is called by an object of class C, an error occurs.

None of the above.

Question 13

1 out of 1 points

When a superclass variable refers to a subclass object and a method is called on that object, the proper implementation is determined at execution time. What is the process of determining the correct method to call?

Selected Answer: late binding.

Answers:
- early binding.
- non-binding.
- on-time binding.
- late binding.

Question 14

1 out of 1 points

Which of the following statements is *false*?

Selected Answer: With non-static interface methods, helper methods can now be declared directly in interfaces rather than in separate classes.

Answers:
- Prior to Java SE 8, it was common to associate with an interface a class containing static helper methods for working with objects that implemented the interface.
- Class Collections contains many static helper methods for working with objects that implement interfaces Collection, List, Set and more.
- Collections method sort can sort objects of any class that implements interface List.
- With non-static interface methods, helper methods can now be declared directly in interfaces rather than in separate classes.

Question 15

1 out of 1 points

Which of the following could be used to declare abstract method method1 in abstract class Class1 (method1 returns an int and takes no arguments)?

Selected Answer: public abstract int method1();

Answers:
public int method1();
public int abstract method1();
public abstract int method1();
public int nonfinal method1();

Question 16

1 out of 1 points

Which keyword is used to specify that a class will define the methods of an interface?

Selected Answer: implements

Answers:
uses
implements
defines
extends

Question 17

1 out of 1 points

Which of the following is *not* possible?

Selected Answer: A class that inherits from two classes.

Answers:
A class that implements two interfaces.
A class that inherits from two classes.
A class that inherits from one class, and implements an interface.
All of the above are possible.

Question 18

1 out of 1 points

Which of the following statements is *false*?

Selected Answer: References to interface types do not have access to method toString.

Answers:
References to interface types do not have access to method toString.

Method toString can be invoked implicitly on any object.

With inheritance, classes and their inherited classes tend to be very similar.

Dramatically different classes can often meaningfully implement the same interface.

Question 19

1 out of 1 points

Which of the following statements is *false*?

Selected Answer: When you enhance an existing interface with default methods—any class that implemented the original interface will break.

Answers:

In Java SE 8, an interface may declare default methods—that is, public methods with concrete implementations that specify how an operation should be performed.

When a class implements an interface, the class receives the interface's default concrete implementations if it does not override them.

When you enhance an existing interface with default methods—any class that implemented the original interface will break.

With default methods, you can declare common method implementations in interfaces (rather than abstract classes), which gives you more flexibility in designing your classes.

Question 20

1 out of 1 points

Classes and methods are declared final for all but the following reasons:

Selected Answer: final methods are static.

Answers:

final methods allow inlining the code.

final methods and classes prevent further inheritance.

final methods are static.

final methods can improve performance.

Question 21

1 out of 1 points

Assigning a subclass reference to a superclass variable is safe _____.

Selected Answer: because the subclass object is an object of its superclass.

Answers:
- because the subclass object has an object of its superclass.
- because the subclass object is an object of its superclass.
- only when the superclass is abstract.
- only when the superclass is concrete.

Question 22

1 out of 1 points

Declaring a method final means:

Selected Answer: it cannot be overridden.

Answers:
- it will prepare the object for garbage collection.
- it cannot be accessed from outside its class.
- it cannot be overloaded.
- it cannot be overridden.

Question 23

1 out of 1 points

Which of the following is *false*?

Selected Answer: It's OK to any of a class's methods from its constructors.

Answers:
- You should not call overridable methods from constructors—when creating a subclass object, this could lead to an overridden method being called before the subclass object is fully initialized.
- It's OK to any of a class's methods from its constructors.
- When you construct a subclass object, its constructor first calls one of the direct superclass's constructors. If the superclass constructor calls an overridable method, the subclass's version of that method will be called by the superclass constructor.
- It's acceptable to call a static method from a constructor.

Question 24

1 out of 1 points

A(n) _____ class cannot be instantiated.

Selected Answer: abstract.

Answers:
final.
concrete.
abstract.
polymorphic.

Question 25

1 out of 1 points

If the superclass contains only abstract method declarations, the superclass is used for _____.

Selected Answer: interface inheritance.

Answers:
implementation inheritance.
interface inheritance.
Both.
Neither.

Question 26

1 out of 1 points

Every object in Java knows its own class and can access this information through method _____.

Selected Answer: getClass.

Answers:
getClass.
getInformation.
objectClass.
objectInformation.

Question 27

1 out of 1 points

All of the following methods are implicitly final except:

Selected Answer: a method in an abstract class.

Answers:
a method in an abstract class.
a private method.
a method declared in a final class.

static
method.

Question 28

1 out of 1 points

Which statement *best* describes the relationship between superclass and subclass types?

Selected Answer: A subclass reference *can* be assigned to a superclass variable, but a superclass reference *cannot* be assigned to a subclass variable.

Answers:

A subclass reference *cannot* be assigned to a superclass variable and a superclass reference *cannot* be assigned to a subclass variable.

A subclass reference *can* be assigned to a superclass variable and a superclass reference *can* be assigned to a subclass variable.

A superclass reference *can* be assigned to a subclass variable, but a subclass reference *cannot* be assigned to a superclass variable.

A subclass reference *can* be assigned to a superclass variable, but a superclass reference *cannot* be assigned to a subclass variable.

Question 29

1 out of 1 points

Polymorphism enables you to:

Selected Answer: program in the general.

Answers: program in the general.

program in the specific.

absorb attributes and behavior from previous classes.

hide information from the user.

Question 30

1 out of 1 points

Which of the following does *not* complete the sentence correctly?

An interface _____.

Selected Answer: can be instantiated.

Answers: forces classes that implement it to declare all the abstract interface methods.

can be used in place of an abstract class when there is no default implementation to inherit.

is declared in a file by itself and is saved in a file with the same name as the interface followed by the .java extension.

can be instantiated.

Monday, March 4, 2024 10:17:51 AM CST

← OK

Review Test Submission: Homework 3

User	Brittany Sifford
Course	Java Programming (Intermediate) (92759)
Test	Homework 3
Started	2/6/24 1:52 PM
Submitted	2/6/24 3:12 PM
Due Date	2/25/24 11:59 PM
Status	Completed
Attempt Score	40 out of 40 points
Time Elapsed	1 hour, 19 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

1 out of 1 points

Which of the following is not an abstract implementation provided by the collections framework?

Selected Answer: AbstractTree.

Answers:
- AbstractCollection.
- AbstractTree.
- AbstractMap.
- AbstractList.

Question 2

1 out of 1 points

Which statement is false?

Selected Answer: The only operation permitted on the view returned by Arrays method asList is delete, which deletes the value from the view and the backing array.

Answers:
- When a List is backed by an array, any modifications made through the List view change the array.
- When a List is backed by an array, any modifications made to the array change the List view.

The only operation permitted on the view returned by Arrays method asList is delete, which deletes the value from the view and the backing array.

Adding elements to the view returned by Arrays method asList results in an UnsupportedOperationException.

Question 3

1 out of 1 points

Maps allocate keys to values and cannot contain duplicate keys, i.e., the key-to-value mapping is a _____ mapping.

Selected Answer: one-to-one.

Answers:
- many-to-many.
- many-to-one.
- one-to-many.
- one-to-one.

Question 4

1 out of 1 points

The classes and interfaces which comprise the collections framework are members of package _____.

Selected Answer: java.util.

Answers:
- java.util.
- javax.swing.
- java.collections.
- java.collection.

Question 5

1 out of 1 points

Which statement is false?

Selected Answer: Unmodifable wrappers throw ModificationExceptions if attempts are made to modify the collection.

Answers:
- The Collections API provides a set of public static methods for converting collections to unmodifiable versions.
- Unmodifable wrappers throw ModificationExceptions if attempts are made to modify the collection.

You can use an unmodifiable wrapper to create a collection that offers read-only access to others while allowing read-write access to yourself.

You can create the kind of collection mentioned in part (c) simply by giving others a reference to the unmodifiable wrapper while you also retain a reference to the wrapped collection itself.

Question 6

1 out of 1 points

Which statement is false?

Selected Answer: A List cannot contain duplicate elements.

Answers:
- A List is a Collection.
- A List cannot contain duplicate elements.
- A List is sometimes called a sequence.
- Lists use zero-based indices.

Question 7

1 out of 1 points

Objects of many classes can now be output and input with Java's object _____.

Selected Answer: serialization.

Answers:
- encapsulation.
- overloading.
- serialization.
- reflection.

Question 8

1 out of 1 points

Which statement is false?

Selected Answer: Class SortedSet implements TreeSet.

Answers:
- SortedSet extends Set.
- Class SortedSet implements TreeSet.
- When a HashSet is constructed, it removes any duplicates in the Collection.
- By definition, a Set object does not contain any duplicates.

Question 9

1 out of 1 points

Which of the following does not implement interface List?

Selected Answer: ListIterator.

Answers:
- ArrayList.
- LinkedList.
- Vector.
- ListIterator.

Question 10

1 out of 1 points

LinkedList method listIterator returns a(n) _____.

Selected Answer: bidirectional iterator.

Answers:
- Iterator.
- List.
- sub list.
- bidirectional iterator.

Question 11

1 out of 1 points

Method shuffle is a member of _____.

Selected Answer: class Collections.

Answers:
- class Arrays.
- class Collections.
- interface Collection.
- Interface List.

Question 12

1 out of 1 points

Which statement is false?

Selected Answer: PriorityQueue orders elements in increasing order, so that smallest value will be the first element removed from PriorityQueue.

Answers:
- Queue is a new collection interface introduced in J2SE 5.0.

Queue and PriorityQueue are included in the java.util package.

PriorityQueue orders elements in increasing order, so that smallest value will be the first element removed from PriorityQueue.

Queue extends interface Collection.

Question 13

1 out of 1 points

Which of the following performs a boxing conversion?

Selected Answer: Integer x = 7;

Answers:
int x = 7;
Integer x = 7;
Neither of the above.
Both of the above.

Question 14

1 out of 1 points

Which statement is false?

Selected Answer: If the search key is found, method binarySearch returns the List index (position relative to 1) of the element containing the search key.

Answers:
Java does not guarantee which item will be found first when a binarySearch is performed on a List containing multiple elements equivalent to the search key.
If the search key is found, method binarySearch returns the List index (position relative to 1) of the element containing the search key.
The binary search algorithm is fast.
Method binarySearch takes a List as the first argument.

Question 15

1 out of 1 points

The collections framework provides various _____ collection interfaces from which the programmer can quickly "flesh out" complete customized implementations.

Selected Answer: abstract.

Answers: abstract.

concrete.

structured.

unstructured.

Question 16

1 out of 1 points

Which of the following performs an unboxing conversion? Assume x refers to an Integer object.

Selected Answer: int y = x;

Answers: int y = x;

Integer y = x;

Neither of the above.

Both of the above.

Question 17

1 out of 1 points

Which statement is false?

Selected Answer: Class ArrayList is a fixed-size array.

Answers: A ListIterator accesses the elements of a List.

Class ArrayList is a fixed-size array.

A LinkedList is a linked list implementation of a List.

ArrayLists execute faster than Vectors because they are not thread safe.

Question 18

1 out of 1 points

Which statement is false?

Selected Answer: All built-in collections are synchronized.

Answers: All built-in collections are synchronized.

Concurrent access to a Collection by multiple threads could cause indeterminate results or fatal errors.

To prevent potential threading problems, synchronization wrappers are used around collection classes that might be accessed by multiple threads.

A synchronization wrapper class receives method calls, adds some functionality for thread safety and then delegates the calls to the wrapped class.

Question 19

1 out of 1 points

A Properties object is a _____ Hashtable object.

Selected Answer: persistent.

Answers:
- transient.
- persistent.
- polymorphic.
- protected.

Question 20

1 out of 1 points

If the desired Object is not found, binarySearch returns _____.

Selected Answer: a negative value

Answers:
- a positive value
- zero
- a negative value
- an ObjectNotFoundError.

Question 21

1 out of 1 points

_____ methods enable a program to view a portion of a collection.

Selected Answer: Range-view.

Answers:
- Focus-view.
- Range-view.
- Delimiter-view.
- Subset-view.

Question 22

1 out of 1 points

Which statement is false?

Selected Answer: Collections discourage software reuse because they are non-portable.

Answers: A collection is an object that can hold references to other objects.

The collection interfaces declare the operations that can be performed on each type of collection.

Collections discourage software reuse because they are non-portable.

Collections are carefully constructed for rapid execution and efficient use of memory.

Question 23

1 out of 1 points

Collections method _____ returns true if two Collections have no elements in common.

Selected Answer: disjoint.

Answers: shuffle.

contains.

hasCommon.

disjoint.

Question 24

1 out of 1 points

Iterator method _____ determines whether the Collection contains more elements.

Selected Answer: hasNext.

Answers: hasNext.

next.

contains.

containsNext.

Question 25

1 out of 1 points

Interface Collection contains _____ operations (i.e., operations performed on the entire collection).

Selected Answer: bulk.

Answers:
- aggregate.
- composite.
- integral.
- ✓ bulk.

Question 26

1 out of 1 points

PriorityQueue method _____ inserts an element at the appropriate location in the queue.

Selected Answer: ✓ offer.

Answers:
- ✓ offer.
- push.
- poll.
- peek.

Question 27

1 out of 1 points

Collections method _____ returns a Comparator object that orders the collection's elements in reverse order.

Selected Answer: ✓ reverseOrder.

Answers:
- rotate.
- shuffle.
- reverse.
- ✓ reverseOrder.

Question 28

1 out of 1 points

Algorithm _____ randomly orders a List's elements.

Selected Answer: ✓ shuffle.

Answers:
- randomShuffle.
- randomPlacement.
- fiftyTwoCardPickup.
- ✓ shuffle.

Question 29

1 out of 1 points

Which of these is not an example of a "real-life" collection?

Selected Answer: The number of pages in a book.

Answers:
- The cards you hold in a card game.
- Your favorite songs stored in your computer.
- The players on a soccer team.
- The number of pages in a book.

Question 30

1 out of 1 points

Stack method _____ looks at the top element of a stack without removing that element.

Selected Answer: peek.

Answers:
- glance.
- peek.
- look.
- sample.

Question 31

1 out of 1 points

Map method _____ is used to determine whether a map contains a mapping for the specified key.

Selected Answer: containsKey

Answers:
- containsKey
- hasKey
- containsMapping
- hasMapping

Question 32

1 out of 1 points

Which statement about hashing is false?

Selected Answer: A load factor of 1.0 usually results in good hashing performance, but less efficient utilization of memory.

Answers:

Hashing facilitates high-speed storing and retrieval of data.

Two different data items can hash to the same cell; this is called a collision.

A load factor of 0.5 usually results in good hashing performance, but less efficient utilization of memory.

A load factor of 1.0 usually results in good hashing performance, but less efficient utilization of memory.

Question 33

1 out of 1 points

Java supports type inferencing with the <> notation in statements that declare and create generic type variables and objects. For example, the following line:

```
List<String> list = new ArrayList<String>();
```

can be written as:

Selected Answer: List<String> list = new ArrayList<>();

Answers:

List<> list = new ArrayList<>();

List<> list = new ArrayList<String>();

List<String> list = new ArrayList<>();

List<String> list = new ArrayList();

Question 34

1 out of 1 points

Class Collections provides algorithms for reversing, filling and copying _____.

Selected Answer: Lists.

Answers: Lists.

Collections.

Arrays.

Stacks.

Question 35

1 out of 1 points

If no elements are in the Stack, method pop throws an _____.

Selected Answer: EmptyStackException.

Answers:

OutOfMemoryError.

OutOfMemoryException.

EmptyStackError.

EmptyStackException.

Question 36

1 out of 1 points

A(n) _____ allows a program to walk through the collection and remove elements from the collection.

Selected Answer: Iterator.

Answers:
Set.
Queue.
Iterator.
List.

Question 37

1 out of 1 points

Comparator method compare should return _____ if the first argument is greater than the second argument.

Selected Answer: a positive int value.

Answers:
a positive int value.
zero.
a negative int value.
a String.

Question 38

1 out of 1 points

Collections method sort that accepts a List as an argument. It sorts the List elements, which must implement the _____ interface.

Selected Answer: Comparable.

Answers:
Comparable.
Comparator.
Compare.
Ordering.

Question 39

1 out of 1 points

Which statement is false?

Selected Answer: The methods for primitive types correspond to the methods for the corresponding type-wrapper classes.

Answers:
- Each primitive type has a corresponding type-wrapper class.
- The type-wrapper classes enable you to manipulate primitive-type values as objects.
- Type-wrapper classes are final, so you cannot extend them.
- The methods for primitive types correspond to the methods for the corresponding type-wrapper classes.

Question 40

1 out of 1 points

To find the smallest and largest element of a Collection, use Collections methods _____ and _____.

Selected Answer: min, max.

Answers:
- least, greatest.
- smallest, largest.
- first, last.
- min, max.

Monday, March 4, 2024 10:18:13 AM CST

← OK

combining

Static class variables:

Selected Answer: are shared by all objects of a class.

Answers:
- are final.
- are public.
- are private.
- are shared by all objects of a class.

Question 4

1 out of 1 points

Instance variables declared final do not or cannot:

Selected Answer: Be modified after they are initialized.

Answers:
- Cause syntax errors if used as a left-hand value.
- Be initialized.
- Be modified after they are initialized.
- None of the above.

Question 5

1 out of 1 points

Which of the following statements is *true*?

Selected Answer: Methods and instance variables can both be either public or private.

Answers:
- Methods and instance variables can both be either public or private.
- Information hiding is achieved by restricting access to class members via keyword public.
- The private members of a class are directly accessible to the clients of a class.
- None of the above is true.

Question 6

1 out of 1 points

When must a program *explicitly* use the *this* reference?

Selected Answer: Accessing an instance variable that is shadowed by a local variable.

Answers:
- Accessing a private variable.
- Accessing a public variable.
- Accessing a local variable.
- Accessing an instance variable that is shadowed by a local variable.

Question 7

1 out of 1 points

A constructor *cannot*:

Selected Answer: specify return types or return values.

Answers:
- be overloaded.
- initialize variables to their defaults.
- specify return types or return values.
- have the same name as the class.

Question 8

1 out of 1 points

A programmer-defined constructor that has *no* arguments is called a(n) _____.

Selected Answer: no-argument constructor.

Answers:
- empty constructor.
- no-argument constructor.
- default constructor.
- null constructor.

Question 9

1 out of 1 points

What happens when this is used in a constructor's body to call another constructor of the same class if that call is not the first statement in the constructor?

Selected Answer: A compilation error occurs.

Answers:
- A compilation error occurs.
- A runtime error occurs.

A logic error occurs.

Nothing happens. The program compiles and runs.

Question 10

1 out of 1 points

The _____ of a class are also called the public services or the public interface that the class provides to its clients.

Selected Answer: public methods.

Answers:
public constructors.
public instance variables.
public methods.
All of the above.

Question 11

1 out of 1 points

A final field should also be declared _____ if it is initialized in its declaration.

Selected Answer: static.

Answers:
private.
public.
protected.
static.

Question 12

1 out of 1 points

When implementing a method, use the class's *set* and *get* methods to access the class's _____ data.

Selected Answer: private.

Answers:
public.
private.
protected.
All of the above.

Question 13

1 out of 1 points

Constructors:

Selected Answer: Both (a) and (c).

Answers:
- Initialize instance variables.
- When overloaded, can have identical argument lists.
- When overloaded, are selected by number, types and order of types of parameters.
- Both (a) and (c).

Question 14

1 out of 1 points

Which of the following class members should usually be private?

Selected Answer: Variables (or fields).

Answers:
- Methods.
- Constructors.
- Variables (or fields).
- All of the above.

Question 15

1 out of 1 points

Set methods are also commonly called _____ methods and *get* methods are also commonly called _____ methods.

Selected Answer: mutator, accessor.

Answers:
- query, mutator.
- accessor, mutator.
- mutator, accessor.
- query, accessor.

Question 16

1 out of 1 points

When no access modifier is specified for a method or variable, the method or variable:

Selected Answer: Has package access.

Answers:
- Is public.
- Is private.

Has package access.

Is static.

Question 17

1 out of 1 points

Which method returns an array of the enum's constants?

Selected Answer: values.

Answers: values.

getValues.

constants.

getConstants.

Question 18

1 out of 1 points

Having a *this* reference allows:

Selected Answer: All of the above.

Answers: a method to refer explicitly to the instance variables and other methods of the object on which the method was called.

a method to refer implicitly to the instance variables and other methods of the object on which the method was called.

an object to reference itself.

All of the above.

Question 19

1 out of 1 points

Which statement is *true* when a superclass has protected instance variables?

Selected Answer: All of the above.

Answers: A subclass object can assign an invalid value to the superclass's instance variables, thus leaving an object in an inconsistent state.

Subclass methods are more likely to be written so that they depend on the superclass's data implementation.

We may need to modify all the subclasses of the superclass if the superclass implementation changes.

All of the above.

Question 20

1 out of 1 points

Private fields of a superclass can be accessed in a subclass

Selected Answer: by calling public or protected methods declared in the superclass.

Answers:
- by calling private methods declared in the superclass.
- by calling public or protected methods declared in the superclass.
- directly.
- All of the above.

Question 21

1 out of 1 points

Which of the following statements is false?

Selected Answer: If the class you're inheriting from declares instance variables as private, the inherited class can access those instance variables directly.

Answers:
- A class can directly inherit from class Object.
- It's often much more efficient to create a class by inheriting from a similar class than to create the class by writing every line of code the new class requires.
- If the class you're inheriting from declares instance variables as private, the inherited class can access those instance variables directly.
- A class's instance variables are normally declared private to enforce good software engineering.

Question 22

1 out of 1 points

Inheritance is also known as the

Selected Answer: *is-a* relationship.

Answers:
- *knows-a* relationship.
- *has-a* relationship.
- *uses-a* relationship.

is-a relationship.

Question 23

1 out of 1 points

Which of the following statements is *false*?

Selected Answer: A superclass object is a subclass object.

Answers:

A subclass is often larger than its superclass.

A superclass object is a subclass object.

The class following the extends keyword in a class declaration is the direct superclass of the class being declared.

Java uses interfaces to provide the benefits of multiple inheritance.

Question 24

1 out of 1 points

Which of the following keywords allows a subclass to access a superclass method even when the subclass has overridden the superclass method?

Selected Answer: super.

Answers:

base.

this.

public.

super.

Question 25

1 out of 1 points

An advantage of inheritance is that:

Selected Answer: Objects of a subclass can be treated like objects of their superclass.

Answers:

All methods can be inherited.

All instance variables can be uniformly accessed by subclasses and superclasses.

Objects of a subclass can be treated like objects of their superclass.

None of the above.

Question 26

1 out of 1 points

Which superclass members are inherited by all subclasses of that superclass?

Selected Answer: protected instance variables and methods.

Answers:
- private instance variables and methods.
- protected instance variables and methods.
- private constructors.
- protected constructors.

Question 27

1 out of 1 points

The default implementation of method clone of Object performs a _____.

Selected Answer: shallow copy.

Answers:
- empty copy.
- deep copy.
- full copy.
- shallow copy.

Question 28

1 out of 1 points

Overriding a method differs from *overloading* a method because:

Selected Answer: Overridden methods have the same signature.

Answers:
- Overloaded methods have the same signature.
- Overridden methods have the same signature.
- Both of the above.
- Neither of the above.

Question 29

1 out of 1 points

Every class in Java, except _____, extends an existing class.

Selected Answer: Object.

Answers:
- Integer.

Object.

String.

Class.

Question 30

1 out of 1 points

When a subclass constructor calls its superclass constructor, what happens if the superclass's constructor does not assign a value to an instance variable?

Selected Answer: The program compiles and runs because the instance variables are initialized to their default values.

Answers:

A syntax error occurs.

A compile-time error occurs.

A run-time error occurs.

The program compiles and runs because the instance variables are initialized to their default values.

Question 31

1 out of 1 points

Consider the classes below, declared in the same file:

```
class A
{
   int a;
   public A()
   {
        a = 7;
   }
}

class B extends A
{
     int b;
     public B()
    {
         b = 8;
    }
}
```

Which of the statements below is *false*?

Selected Answer: A reference of type A can be treated as a reference of type B.

Answers:

Both variables a and b are instance variables.

After the constructor for class B executes, the variable a will have the value 7.

After the constructor for class B executes, the variable b will have the value 8.

A reference of type A can be treated as a reference of type B.

Question 32

1 out of 1 points

Using the protected keyword also gives a member:

Selected Answer: package access.

Answers:
- public access.
- package access.
- private access.
- block scope.

Question 33

1 out of 1 points

Which of the following is the superclass constructor call syntax?

Selected Answer: keyword super, followed by a set of parentheses containing the superclass constructor arguments.

Answers:
- keyword super, followed by a dot (.) .
- keyword super, followed by a set of parentheses containing the superclass constructor arguments.
- keyword super, followed by a dot and the superclass constructor name.
- None of the above.

Question 34

1 out of 1 points

When overriding a superclass method and calling the superclass version from the subclass method, failure to prefix the superclass method name with the keyword super and a dot (.) in the superclass method call causes _____.

Selected Answer: infinite recursion.

Answers:
- a compile-time error.
- a syntax error.
- infinite recursion.

a runtime error.

Question 35

1 out of 1 points

Superclass methods with this level of access *cannot* be called from subclasses.

Selected Answer: ✓ private.

Answers: ✓ private.

public.

protected.

package.

Question 36

1 out of 1 points

The default equals implementation of class Object determines:

Selected Answer: ✓ whether two references refer to the same object in memory.

Answers: ✓ whether two references refer to the same object in memory.

whether two references have the same type.

whether two objects have the same instance variables.

whether two objects have the same instance variable values.

Monday, March 4, 2024 10:16:27 AM CST

← OK