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