

Both the methods sleep and join throws InterruptedException.

Select an answer

- *A. TRUE
- B. FALSE

This is a legal int x[]={2,3,7,9};

Select an answer

- *A. TRUE
- B. FALSE

Can you automatically convert a long to an int if the long value is small enough to fit into an int?

Select an answer

- A. Yes
- *B. No

Collection will not represent a single unit of objects.

Select an answer

- A. TRUE
- *B. FALSE

By default, all program import the java.lang package.

Select an answer

- *A. TRUE
- B. FALSE

Is it legal to access a static method using an instance of an class?

Select an answer

- A. Yes
- *B. No

A method can have more than one access modifier?

Select an answer

- *A. TRUE
- B. FALSE

A source stream initiates the flow of data, also called an input stream.

Select an answer

- *A. TRUE
- B. FALSE

ou can pass an object reference as an argument to a method call.

Select an answer

- *A. TRUE
- B. FALSE

When you pass a variable an argument to a method is called.

Select an answer

- *A. TRUE

B. FALSE

When a Java program tries to divide by zero; the runtime will create and throw an object of Arithmetic Exception.

Select an answer

- *A. TRUE
- B. FALSE

An object means, anything that can be seen or touched; material thing.

Select an answer

- *A. TRUE
- B. FALSE

Suppose that Marketing is a subclass of Department. What is casting used for in the code? `public organize (Department sales) { Marketing marketing1 = (Marketing) sales ; }`
(Please select ALL that apply)

Select an answer

- A. To cast sales to a Marketing reference
- B. To convert a Department object into a Marketing object
- C. To make the code clearer
- *D. To perform a narrowing conversion

Which constructs can you make private?

(Please select ALL that apply)

Select an answer

- A. Local variables
- *B. Member variables
- *C. Methods
- *D. Top-level classes

A byte can be of what size?

Select an answer

- *A. -128 to 127
- B. $(-2 \text{ power } 8) - 1$ to $2 \text{ power } 8$
- C. -255 to 256
- D. Depends on the particular implementation of the Java Virtual machine

What is a constructor?

Select an answer

- A. A constructor is a special kind of method that determines how a class is initialized when created
- B. A constructor is a special kind of class that determines how an object is initialized when created
- *C. A constructor is a special kind of method that determines how an object is initialized when created
- D. A constructor is a special kind of method that determines how a message is

A ref is a:

Select an answer

- *A. Method of passing the values of parameters
- B. Pointer to an object
- C. Method of compiling the PL/SQL
- D. A wrapper for encrypting the source
- E. None of the options are correct

Identify the best examples of an inheritance hierarchy.
(Please select ALL that apply)

Select an answer

- A. Animal extends Dog extends Cat
- *B. Cappuccino extends Coffee extends Beverage
- C. Director extends Manager extends Employee
- D. Vehicle extends Car

<p>What happens when you put this expression in your code? </p>
`int x=0;
int y = 4%x;`

Select an answer

- *A. Compile time error
- B. Compiles and runs
- C. Runtime exception

Can we create one or more package declaration per source file?

Select an answer

- A. Yes, specify multiple package declaration at starting of the program
- *B. No you can have only one package declaration per file
- C. Yes, you can use scope braces to show which package definition holds good for respected blocks
- D. None of the options are correct

You have a method that throws a checked exception. What is required to ensure that the code correctly compiles and any potential exceptions are handled by all callers to this method?
(Please select ALL that apply)

Select an answer

- *A. The called method must be declared with a throws clause
- *B. The calling method must either enclose the method call in a try...catch block or declare the relevant exception to be thrown
- C. The calling method must include a catch block for each specific type of exception thrown by the called method
- D. The calling method must override a method of a superclass

Suppose you have an abstract Vehicle class. The Ford class extends the Car class, which extends the Vehicle class, and overrides its getSpec method. You assign a Ford reference to a variable, vehicle, of type Vehicle. Which getSpec method is called through vehicle? `Vehicle a = new Ford(); a.getSpec();`

Select an answer

- A. The getSpec method of the Car object
- *B. The getSpec method of the Ford object
- C. The getSpec method of the Vehicle object

Suppose you needed to pass a variable number of integer arguments to the newMethod method and have to make use of the varargs feature. Which statement is syntactically correct?

Select an answer

- A. static void newMethod (intnumbr[])
- *B. static void newMethod (int ... numbr)
- C. static void newMethod (int --- numbr)
- D. static void newMethod (boolean ... numbr)

The three basic read methods are:
(Please select ALL that apply)

Select an answer

- A. int read([]char)
- B. int read()
- *C. int read(char[] cbuf)
- *D. int read(char[] cbuf, int offset, int length)

What is the work of java Runtime environment?
(Please select ALL that apply)

Select an answer

- A. Creates code
- B. Loads code
- C. Verifies code
- D. Executes code
- *E. None of the options are correct

What will be the output of the program?

```
public class Foo
{
    public static void main(String[] args)
    {
        try
        {
            return;
        }
        finally
        {
            System.out.println( "Finally" );
        }
    }
}
```

Select an answer

- *A. Finally
- B. Compilation fails.
- C. The code runs with no output.
- D. An exception is thrown at runtime.

Which three form part of correct array declarations?
(Please select ALL that apply)

Select an answer

- *A. public int a []
- *B. static int [] a
- *C. public [] int a
- D. private int a [3]
- E. private int [3] a []
- F. public final int [] a

1.Does a final member variable has to be initialized when it is declared?

Select an answer

- A. TRUE
- B. FALSE----->

2. A class can define two methods with the same name as long as the return types are different. State whether the statement is true or false.

Select an answer

- A. TRUE----->
- B. FALSE

3. When a Java program tries to divide by zero; the runtime will create and throw an object of Arithmetic Exception.

Select an answer

- A. TRUE----->
- B. FALSE

4. <p>Will this compile?</p> float f = 27.24f;

Select an answer

- A. Yes----->
- B. No

5._____ expressions are a way to describe a set of strings based on common characteristics shared by each string in the set. They can be used to search, edit, or manipulate text and data.

Select an answer

- A. Predefined----->
- B. Interface

6.A source stream initiates the flow of data, also called an input stream.

Select an answer

- A. TRUE----->
- B. FALSE

7.By default, all program import the java.lang package.

Select an answer

- A. TRUE----->
- B. FALSE

8.x=true; y=true; What is the result of x^y?

Select an answer

- A. TRUE----->
- B. FALSE

9. a method with no access modifiers CAN be overridden by a method marked protected.

Select an answer

- A. TRUE----->
- B. FALSE

10.Over loaded methods can change the access modifiers.

Select an answer

- A. TRUE----->
- B. FALSE

11. Static methods can be declared in an interface.

Select an answer

- A. TRUE
- B. FALSE----->

12. Both the methods sleep and join throws Interrupted Exception.

Select an answer

- A. TRUE----->
- B. FALSE

13.The Vector class implements a growable array of objects.

Select an answer

- A. TRUE----->
- B. FALSE

14.Is it legal to access a static method using an instance of an class?

Select an answer

- A. Yes----->
- B. No

15.Are static methods allowed to access instance variables using this.var?

Select an answer

- A. Yes

B. No----->

16.The Collection interface is the root interface for:
(Please select ALL that apply)

Select an answer

- A. Storing a classes----->
- B. Storing a collection of objects----->
- C. Processing a collection of objects
- D. Processing a block

17.Which one is a reserved keyword in java?

Select an answer

- A. Method
- B. Native----->
- C. Subclasses
- D. Reference
- E. Array

18.In order for a source code file to successfully compile which contains the public class Test, which of the following must be true?

Select an answer

- A. It must have a package statement
- B. It must be named Test.java----->
- C. It must import java.lang
- D. It must declare a public class named Test

19.<p>x = 5; z = x++; </p>What are the values of x and z?

Select an answer

- A. x = 5; z = 6
- B. x = 5; z = 6
- C. x = 6; z = 5----->
- D. x = 6; z = 2

20.Which are the features of JRE?
(Please select ALL that apply)

Select an answer

- A. The compiler tool---->
- B. The debugger tool
- C. The Java SE API-----
- D. The JVM----->

21.What are the uses of inner classes?

Select an answer

- A. Allow a class definition to be placed inside another class definition----->
B. Group classes that logically belong together
C. Used to overcome the boundaries of access specifiers
D. None of the options are correct

22.The concept of multiple inheritance is implemented in Java by:
(Please select ALL that apply)

Select an answer

- A. Extending one class and implementing one or more interfaces---
----->
B. Implementing two or more interfaces----->
C. All of these

23.Which of the following lines will compile without warning or error?

Select an answer

- A. Char c="a";
B. Byte b=257;
C. Boolean b=null;----->
D. Float f=1.3;

24.Which of the following is Java keyword?

Select an answer

- A. External
B. Implement----->
C. x = 6; z = 5
D. Integer

25.Which of the following are legal lines of codes?

Select an answer

- A. Int w = (int) 888.8;
B. Byte x = (byte)1000L;
C. Long y = (byte)100;
D. Byte z = (byte)100L;
E. All the options are correct----->

26.Given: String x = new String("xyz"); y = "abc"; x = x+y; How many String objects have been created?

Select an answer

- A. 2

- B. 3
- C. 4----->
- D. 5

27. Suppose you needed to pass a variable number of integer arguments to the newMethod method and have to make use of the varargs feature. Which statement is syntactically correct?

Select an answer

- A. static void newMethod (intnumbr[])-----
- B. static void newMethod (int ... numbr)
- C. static void newMethod (int --- numbr)
- D. static void newMethod (boolean ... numbr)

28. What will be the output of the program?

```
public class X
{
    public static void main(String [] args)
    {
        try
        {
            badMethod();
            System.out.print("A");
        }
        catch (RuntimeException ex) /* Line 10 */
        {
            System.out.print("B");
        }
        catch (Exception ex1)
        {
            System.out.print("C");
        }
        finally
        {
            System.out.print("D");
        }
        System.out.print("E");
    }
    public static void badMethod()
    {
        throw new RuntimeException();
    }
}
```

Select an answer

- A. BD
- B. BCD
- C. BDE----->
- D. BCDE

29. Select the statement that best indicates a situation with low coupling.

Select an answer

- A. The attributes of the class are all private
- B. The class refers to a small number of other objects
- C. The object contains only a small number of variables
- D. It is unlikely that changes made to one class will require any changes in another----->

30. Suppose the Furniture class contains a method called createDeliveryOrder. Each type of furniture requires a specific delivery order. You are creating Table class, which extends the Furniture class. What is the best way to handle the createDeliveryOrder method of the Table class?

Select an answer

- A. Implement constructor chaining----->
- B. Overload the createDeliveryOrder method in the Table class
- C. Override the createDeliveryOrder method

core java

What are the functions of the JVM?

Select an answer

- A. It ensures that Java is platform independent
- *B. It interprets compiled Java code
- C. It groups software into libraries of related components
- D. It provides components for the creation of applets

Question: 2 of 30

lt043991enQ0077

Choose the values that can be entered for Float Data type?

Select an answer

- A. 3.14d
- *B. 3.14e
- C. 3.14e+1d
- D. 3.14f
- E. 3.14f

Question: 3 of 30

lt043991enQ0049

The lock flag is released in the following events:
(Please select ALL that apply)

Select an answer

- A. Released data protected by synchronized should be private

- *B. Released when the thread passes the end of the synchronized code block
- *C. Released automatically when a break, return, or exception is thrown by the synchronized code block
- D. Released Share the same data and code

Question: 4 of 30

lt043991enQ0048

Multithreaded programming has following characteristics:
(Please select ALL that apply)

Select an answer

- A. Place the thread
- B. Share the data from other thread
- *C. Multiple threads are from one Runnable instance
- *D. Threads share the same data and code

Question: 5 of 30

lt043991enQ0098

<p>What happens when you put this expression in your code? </p>int x=0;
int y = 4%x;

Select an answer

- A. Compile time error
- B. Compiles and runs
- *C. Runtime exception

Question: 6 of 30

lt043991enQ0079

What are the uses of inner classes?

Select an answer

- *A. Allow a class definition to be placed inside another class definition
- B. Group classes that logically belong together
- C. Used to overcome the boundaries of access specifiers
- D. None of the options are correct

Question: 7 of 30

lt043991enQ0052

Which constructs can you make private?
(Please select ALL that apply)

Select an answer

- A. Local variables
- *B. Member variables
- *C. Methods
- D. Top-level classes

Question: 8 of 30

lt043991enQ0068

Which are keywords in Java?

Select an answer

- A. NULL
- B. Sizeof
- C. Friend
- D. Extends
- *E. Synchronized

Question: 9 of 30

lt043991enQ0099

What can be used only in conjunction with an object of a class?
(Please select ALL that apply)

Select an answer

- *A. A class method
- B. A class variable
- *C. An instance method
- D. An instance variable

Question: 10 of 30

lt043991enQ0132

Which of the following are legal lines of codes?

Select an answer

- A. Int w = (int) 888.8;
- B. Byte x = (byte)1000L;
- C. Long y = (byte)100;
- D. Byte z = (byte)100L;
- *E. All the options are correct

Question: 11 of 30

lt043991enQ0120

Given: String x = new String("xyz"); y = "abc"; x = x+y; How many String objects have been created?

Select an answer

- A. 2
- B. 3
- *C. 4
- D. 5

Question: 12 of 30

lt043991enQ0137

Given the code sample, which classes can access the Operation variable and the Calculate method? package tools.myobjects; public class Calculator { protected int Operation; protected void Calculate() { } }
(Please select ALL that apply)

Select an answer

- *A. Classes in the tools.myobjects package
- B. Classes in the samples.access package
- *C. The Calculator class
- D. Subclasses of the Calculator class in the samples.access package
- *E. Subclasses of the Calculator class in the tools.myobjects package

Question: 13 of 30

lt043991enQ0125

Which two are true?

Select an answer

- A. A finalizer may NOT be invoked explicitly
- *B. The finalize method declared in class Object takes no action

- C. `super.finalize()` is called implicitly by any overriding `finalize` method
- *D. The `finalize` method for a given object will be called no more than once by the garbage collector

Question: 14 of 30

lt043991enQ0146

Suppose the `Furniture` class contains a method called `createDeliveryOrder`. Each type of furniture requires a specific delivery order. You are creating `Table` class, which extends the `Furniture` class. What is the best way to handle the `createDeliveryOrder` method of the `Table` class?

Select an answer

- *A. Implement constructor chaining
- B. Overload the `createDeliveryOrder` method in the `Table` class
- C. Override the `createDeliveryOrder` method

Question: 15 of 30

lt043991enQ0126

Select the statement that best indicates a situation with low coupling.

Select an answer

- A. The attributes of the class are all private
- B. The class refers to a small number of other objects
- C. The object contains only a small number of variables
- *D. It is unlikely that changes made to one class will require any changes in another

Question: 17 of 30

lt043991enQ0006

The `Vector` class implements a growable array of objects.

Select an answer

- *A. TRUE
- B. FALSE

Question: 18 of 30

lt043991enQ0027

JAVA identifiers is case sensitive.

Select an answer

- *A. TRUE
- B. FALSE

Question: 19 of 30

lt043991enQ0032

JVM is Platform Independent.

Select an answer

- A. TRUE
- *B. FALSE

Question: 20 of 30

lt043991enQ0030

<p>Will this compile?</p> float f = 27.24f;

Select an answer

- *A. Yes
- B. No

Question: 21 of 30

lt043991enQ0020

The do-while loop repeats a set of code atleast once before the condition is tested.

Select an answer

- *A. TRUE
- B. FALSE

Question: 22 of 30

lt043991enQ0007

A source stream initiates the flow of data, also called an input stream.

Select an answer

- *A. TRUE
- B. FALSE

Question: 23 of 30

lt043991enQ0038

Is it legal to access a static method using an instance of an class?

Select an answer

- *A. Yes
- B. No

Java cannot run on windows 3.1 as Windows 3.1 does not support multi-threading.

Select an answer

- A. TRUE
- *B. FALSE

Question: 25 of 30

lt043991enQ0021

By default, all program import the java.lang package.

Select an answer

- *A. TRUE
- B. FALSE

Question: 26 of 30

lt043991enQ0036

Static variables are implicitly final. State whether the statement is true or false.

Select an answer

- A. TRUE
- *B. FALSE

Question: 27 of 30

lt043991enQ0004

The user can also create her own exception classes by inheriting from the class Exception.

Select an answer

- *A. TRUE

B. FALSE

Question: 28 of 30

lt043991enQ0041

Can you automatically convert a long to an int if the long value is small enough to fit into an int?

Select an answer

A. Yes

*B. No

Question: 29 of 30

lt043991enQ0040

Are static methods allowed to access instance variables using this.var?

Select an answer

A. Yes

*B. No

Question: 30 of 30

lt043991enQ0033

Variables can also be declared after their used.

Select an answer

A. TRUE

*B. FALSE

Question: 1 of 30

lt043991enQ0140

What will be the output of the program?

```
public class RTExcept
{
    public static void throwit ()
    {
        System.out.print("throwit ");
        throw new RuntimeException();
    }
    public static void main(String [] args)
    {
        try
        {
            System.out.print("hello ");
            throwit();
        }
        catch (Exception re )
        {
            System.out.print("caught ");
        }
        finally
        {
            System.out.print("finally ");
        }
        System.out.println("after ");
    }
}
```

}

Select an answer

- A. hellothrowit caught
- B. Compilation fails
- C. hellothrowitRuntimeException caught after
- D. hellothrowit caught finally after

D

Question: 2 of 30

lt043991enQ0118

Choose which definitions holds good for variables.

Select an answer

- A. Variables that are defined inside a method and are called local, automatic, temp, or stack variables
- B. Variables created when the method is executed are destroyed when the method is exited
- C. Variables are created by the JVM
- D. Variables cannot be de-referenced
- E. None of the options are correct

A,B

Question: 3 of 30

lt043991enQ0120

Given: String x = new String("xyz"); y = "abc"; x = x+y; How many String objects have been created?

Select an answer

- A. 2
- B. 3
- C. 4
- D. 5

C

Question: 4 of 30

lt043991enQ0136

The three basic read methods are:

(Please select ALL that apply)

Select an answer

- A. `int read([]char)`
- B. `int read()`
- C. `int read(char[] cbuf)`
- D. `int read(char[] cbuf, int offset, int length`

B,C,D

Question: 5 of 30

lt043991enQ0133

What can you assign to an interface reference type?

(Please select ALL that apply)

Select an answer

- A. A class reference
- B. A primitive type reference
- C. An array reference
- D. An interface reference

A,D

Question: 6 of 30

lt043991enQ0138

How do you prevent any external code from instantiating a class that is not abstract?

(Please select ALL that apply)

Select an answer

- A. Declare a static factory method
- B. Declare at least one constructor
- C. Declare the class private
- D. Make all constructors private

C

Question: 7 of 30

lt043991enQ0019

The switch statement does not require a break.

Select an answer

- A. TRUE
- B. FALSE

B

Question: 8 of 30

lt043991enQ0024

Over loaded methods can change the access modifiers.

Select an answer

- A. TRUE
- B. FALSE

A

Question: 9 of 30

lt043991enQ0033

Variables can also be declared after their used.

Select an answer

- A. TRUE
- B. FALSE

B

Question: 10 of 30

lt043991enQ0016

Procedures with common functionality are grouped into modules.

Select an answer

- A. TRUE
- B. FALSE

A

Question: 11 of 30

lt043991enQ0006

The Vector class implements a growable array of objects.

Select an answer

- A. TRUE
- B. FALSE

A

Question: 12 of 30

lt043991enQ0017

An object means, anything that can be seen or touched; material thing.

Select an answer

- A. TRUE
- B. FALSE

A

Question: 13 of 30

lt043991enQ0018

Character literals are stored as unicode characters.

Select an answer

- A. TRUE
- B. FALSE

A

Question: 14 of 30

lt043991enQ0027

JAVA identifiers is case sensitive.

Select an answer

- A. TRUE
- B. FALSE

A

Question: 15 of 30

lt043991enQ0001

A base class reference can be used to access the sub class methods.

Select an answer

- A. TRUE
- B. FALSE

B

Question: 16 of 30

lt043991enQ0009

Java cannot run on windows 3.1 as Windows 3.1 does not support multi-threading.

Select an answer

- A. TRUE
- B. FALSE

B

Question: 17 of 30

lt043991enQ0030

<p>Will this compile?</p> float f = 27.24f;

Select an answer

- A. Yes
- B. No

A

Question: 18 of 30

lt043991enQ0028

A method can have more than one access modifier?

Select an answer

- A. TRUE
- B. FALSE

B

Question: 19 of 30

lt043991enQ0031

Is this legal ? public static void main(String []hari)

Select an answer

- A. Yes
- B. No

A

Question: 20 of 30

lt043991enQ0040

Are static methods allowed to access instance variables using `this.var`?

Select an answer

- A. Yes
- B. No

B

Question: 21 of 30

lt043991enQ0007

A source stream initiates the flow of data, also called an input stream.

Select an answer

- A. TRUE
- B. FALSE

A

Question: 22 of 30

lt043991enQ0045

Certain subclasses of exceptions must be declared for code to compile successfully. Which events can cause such exceptions to be thrown?
(Please select ALL that apply)

Select an answer

- A. A method tries to divide an integer by zero
- B. An application fails to open a file
- C. An inactive thread is interrupted by another thread
- D. An index to a string or an array is out of range

A, C, D

Question: 23 of 30

lt043991enQ0055

Suppose that Marketing is a subclass of Department. What is casting used for in the code? `public organize
(Department sales) { Marketing marketing1 = (Marketing) sales ; }`
(Please select ALL that apply)

Select an answer

- A. To cast sales to a Marketing reference
- B. To convert a Department object into a Marketing object
- C. To make the code clearer
- D. To perform a narrowing conversion

A, B

Question: 24 of 30

lt043991enQ0090

Which will legally declare, construct, and initialize an array?

Select an answer

- A. `int [] myList = {"1", "2", "3"};`
- B. `int [] myList = (5, 8, 2);`
- C. `intmyList [] [] = {4,9,7,0};`
- D. `intmyList [] = {4, 3, 7};`

D

Question: 25 of 30

lt043991enQ0088

To move control to the start of next repetition, which statement should be used?

Select an answer

- A. While statement
- B. If-else statement
- C. Break statement
- D. Continue statement

D

Question: 26 of 30

lt043991enQ0083

System.out.println(0.0/0.0); What is the result?

Select an answer

- A. Infinity
- B. ∞Infinity
- C. NaN
- D. Runtime Exception

C

Question: 27 of 30

lt043991enQ0052

Which constructs can you make private?
(Please select ALL that apply)

Select an answer

- A. Local variables
- B. Member variables
- C. Methods
- D. Top-level classes

B,C

Question: 28 of 30

lt043991enQ0084

How do we call The Garbage Collection?

Select an answer

- A. Finalize
- B. System.gc()
- C. System.delete()
- D. Finally

B

Question: 29 of 30

lt043991enQ0091

Which of the following are correct methods for initializing the array dayhigh with 7 values?

Select an answer

- A. Intdayhigh = { 24, 23, 24, 25, 25, 23, 21 };
- B. Intdayhigh[] = { 24, 23, 24, 25, 25, 23, 21 };
- C. intdayhigh = new [24, 23, 24, 25, 25, 23, 21];
- D. int[] dayhigh = { 24, 23, 24, 25, 25, 23, 21 };
- E. intdayhigh [] = new int[24, 23, 24, 25, 25, 23, 21];

D

Question: 30 of 30

lt043991enQ0095

What is a constructor?

Select an answer

A. A constructor is a special kind of method that determines how a class is initialized when created

B. A constructor is a special kind of class that determines how an object is initialized when created

C. A constructor is a special kind of method that determines how an object is initialized when created

D. A constructor is a special kind of method that determines how a message is

C

Question:

All Java keywords are written in lower case.

Answer:

true

Explanation:

No keywords in Java start with a capital letter.

=====

Question:

A method name can start with a number.

Answer:

false

Explanation:

Method and variable names can't start with a number, although they can contain numbers. myMethod2() is legal; 2myMethod() is not.

=====

Question:

A variable name can start with \$.

Answer:

true

Explanation:

Methods and variables can start with a \$ character, but its not recommended.

=====

Question:

Which of these are legal values for a boolean?

Answer:

true, false

Explanation:

Unlike C/C++, booleans in Java are represented with the literals true and false. You can't substitute integers such as 1 and 0 where a boolean value is expected.

=====

Question:

What is the bit-depth of a char?

Answer:

16 bits

Explanation:

Chars in Java are all 16-bit Unicode.

=====

Question:

What is the bit-depth of a byte?

Answer:

8 bits

Explanation:

Bytes are always 8 bits. The left-most bit is used to represent the sign.

=====

Question:

A floating point literal in Java is treated as a:

Answer:

double

Explanation:

Floating point literals (example: 2.27) in Java are automatically typed as

doubles unless you add an "f" (example: 2.27f).

=====

Question:

When you pass a reference to an array as an argument to a method call, you pass a new reference to the original array.

Answer:

true

Explanation:

An array is an object, and Java always passes a copy of the reference to the original object. Both references now point to the same array object.

=====

Question:

Does a call to System.gc() force garbage collection to take place? (assuming there are objects to be collected)

Answer:
Not always

Explanation:
Think of a call to `System.gc()` as a... suggestion. Garbage collection may or may not actually occur.

=====

Question:
Which of these operators would cause the bit pattern 00111100 to become 11000011?

Answer:
~

Explanation:
The ~ operator performs a bitwise inversion (flips the bits!).

=====

Question:
Assume `x = 2` and `y = 4`. What is the result of `!(y > x)`

Answer:
false

Explanation:
This test asks if the expression (`y` is greater than 2) is NOT true... and that's false. `y` IS greater than 2, so the result is false.

=====

Question:
Assume `x = 6` and `y = 7`. What is the result of `!(y == x)`

Answer:
true

Explanation:
The test asks if the primitive values of `y` and `x` are NOT equal in value, which is true.

=====

Question:
Assume: `byte a = 3; byte b = 2;`
What is the result of:
`byte c = a + b ;`

Answer:
compiler error

Explanation:
The result of arithmetic is always at LEAST an `int`, so the compiler will ask for an explicit cast: `byte c = (byte)(a + b);` even though `a` and `b` are bytes.

=====

Question:
Can you compare a boolean to an integer?

Answer:
No

Explanation:

Not in Java. A boolean has a value of true or false, not 1 and 0, or 0 and anything else. You just can't do it (watch out C folks)

=====

Question:

What is "instanceof" ?

Answer:

an operator

Explanation:

The "instanceof" operator -- if (A instanceof B) -- tests if an object A is either an instance of B, an instance of a subclass of B, or implements the B interface.

=====

Question:

Integer a = new Integer(5);
Integer b = new Integer(5);
What is the result of running:
if (a == b)

Answer:

false

Explanation:

The == operator looks ONLY for a match of bits. For two different objects, the only way == can be true is if both reference variables point to the SAME object.

=====

Question:

Can a top-level class be marked as private?

Answer:

No

Explanation:

If a class was private... who could use it? For a normal, top-level class, the only access modifiers that make sense are public and default (package). Only nested inner classes can be private.

=====

Question:

Can a top-level class be marked as protected?

Answer:

No

Explanation:

A top-level class can ONLY be marked public or default (package). The protected access is just for member variables and methods, and allows subclasses outside the superclass package to inherit the protected members.

=====

Question:

methods which are marked protected can be called on any subclass of the class in which the method is declared.

Answer:

true

Explanation:

Protected members (methods and data) can be inherited by subclasses, including those outside the package. The only difference between default (no modifier) and protected, is that protected extends visibility outside the package... but ONLY to subclasses.

=====

Question:

Can a private method of a superclass be declared within a subclass?

Answer:

Yes

Explanation:

Trick question! Private methods can NOT be overridden, but they CAN be re-declared/redefined in the subclass, and called directly on the subclass object. But polymorphism will not apply.

=====

Question:

Can an abstract class have non-abstract methods?

Answer:

Yes

Explanation:

Unlike interfaces, abstract classes CAN provide implementations of methods.

=====

Question:

Can a non-abstract class have abstract methods?

Answer:

No

Explanation:

Even a single abstract method tells the compiler that the whole class MUST be declared abstract.

=====

Question:

If a non-abstract class X inherits abstract methods from its abstract parent class Y, is the X class required to implement the inherited abstract methods?

Answer:

Yes

Explanation:

Any concrete (non-abstract) class MUST provide bodies (implementations) for the abstract methods it inherits. Otherwise, the compiler will insist that the new subclass itself be marked as abstract.

=====

Question:

If a class declares that it implements an interface (class A implements B), must class A implement the methods of interface B?

Answer:
Unless A is abstract

Explanation:
If a class is abstract, and it implements an interface, it can leave the interface implementations up to a concrete (non-abstract) subclass.
=====

Question:
TRUE or FALSE -
during arithmetic, when the operands are of different types, the resulting type is ALWAYS the widest of the two types.

Answer:
false

Explanation:
The result of an arithmetic operation on any two primitive integer operands will be at least an int -- even if the operands are byte and short.
=====

Question:
class Y is a subclass of class X. Will this compile?
Y myY = new X();

Answer:
No

Explanation:
You can never have a reference type of a subclass refer to an actual instance of the superclass. Because an object of type X (the superclass) wouldn't have the new methods that an object of type Y (the subclass) is supposed to have.
=====

Question:
Class Y is a subclass of class X. Will this compile?
X myX = new Y();

Answer:
Yes

Explanation:
A reference variable can always refer to any object which is a subclass of the reference type, since the instance methods of the reference variable type (X) are guaranteed to be callable on the actual subclass object (class Y).
=====

Question:
Can you compile an if construct without using curly braces?

Answer:
Yes

Explanation:
If you do not use curly braces for an if block, then only the first statement following the if () will run if the if condition is true.
=====

Question:

What goes in the first part of a for loop?

Answer:

init statement

Explanation:

The init statement must be the first part of the for declaration --

example:

```
for (int i = 7;;)
```

=====

Question:

Is this legal?

```
int i ;
```

```
for ( i = 4, int t = 6; i < 0; i++ )
```

Answer:

No

Explanation:

You can declare or initialize more than one variable in a for loop, but you

can't mix declarations with initializations.

=====

Question:

In a for loop, the continue statement will cause execution to jump out of the loop and continue on at the first statement below the loop.

Answer:

false

Explanation:

Continue really means "continue with the next iteration", so execution of the current iteration stops, and execution continues at the next iteration.

=====

Question:

An inner class must be defined within the opening and closing braces of a class, but outside any method -- just like data members.

Answer:

false

Explanation:

An inner class can be defined inside a method, or even inside a parameter to a method! (with anonymous inner classes)

=====

Question:

A class defined within a parameter to a method is:

Answer:

anonymous

Explanation:

A class defined within a method parameter (typically used with event listener classes) is considered anonymous. The object is created and alive,

but the new class type has no name, so you can't create more instances of it
except by going through that same method.

=====

Question:

when an object is created using new (), its constructor will not complete until all constructors higher on the inheritance tree have completed.

Answer:

true

Explanation:

Remember the rule: the parents must exist before the children, and this goes all the way up the ancestral tree.

=====

Question:

the compiler will always insert a default, no-argument constructor into any class definition.

Answer:

false

Explanation:

The compiler will ONLY insert a default constructor if you don't provide any constructor at all.

=====

Question:

when you override a method, the overriding method in the subclass must not declare ANY new exceptions not declared by the overridden method.

Answer:

false

Explanation:

This rule only applies to "checked" exceptions -- the exceptions which are subclasses of Exception but not subclasses of RuntimeException.

=====

Question:

Will this compile?
float f = 27.24f;

Answer:

Yes

Explanation:

Without the "f", Java will consider a floating point number to be a double, and the compiler will force you to do an explicit cast.

=====

Question:

As a literal, what is 27.24?

Answer:

a double

Explanation:

All floating point literals are assumed to be doubles, not floats!

=====

Question:

Will this compile?
`int[] x = [27];`

Answer:
No

Explanation:
In Java you must create the array using: `int[] x = new int[27]` or by explicitly filling in the values: `int[] x = {4,2,7,3,9,8};`
=====

Question:
When an array is constructed, are its elements automatically initialized?

Answer:
Yes

Explanation:
All array elements are ALWAYS given default values, whether the array is a class, instance, or local variable. Primitives initialize to zero, booleans to false, chars to `'\u0000'` and object references to null.
=====

Question:
Is this legal?
`int x[] = {2,7,4,9};`

Answer:
Yes

Explanation:
This instantiation/initialization is allowed as a shortcut to creating a new array and assigning a value for each index.
=====

Question:
Is this legal?
`public static void main (String[] fred)`

Answer:
Yes

Explanation:
You can call the command-line argument variable whatever you like! And you can put the brackets `[]` before or after the variable name.
=====

Question:
Assume `x = 0` and `y = 3`.
What is the value of `x` after: `x = y++;`

Answer:
3

Explanation:
`x` gets the value of `y`, THEN `y` is incremented. The result would have been 4 if the expression was: `x = ++y ;`
=====

Question:
Given `x = y--;`
Which of these will be true AFTER execution?

Answer:

`x > y`

Explanation:

`y` is assigned to `x`, then `y` is decremented.

So by the end of the statement, `x` is greater than `y`.

=====

Question:

Integer literals are always what size?

Answer:

int - 32 bits

Explanation:

integer literals are always treated as 32-bit ints

=====

Question:

Can you use an increment unary operator on a byte?

(example: `byte b = 4; b++;`)

Answer:

Yes

Explanation:

You don't need an explicit cast to use the increment unary operator.

=====

Question:

What happens when you divide an integer by a variable

with a value of 0?

Answer:

Runtime exception

Explanation:

At runtime you will get an `ArithmeticException` which is not checked for at compile time. Floats and doubles will NOT throw an exception if divided by 0.

=====

Question:

What happens when you put this expression in your code:

`int x = 0;`

`int y = 4 % x ;`

Answer:

Runtime exception

Explanation:

The modulo operator (%) is performing a divide, so you get a divide-by-zero

`ArithmeticException` at runtime. Even using a literal 0 (`y = 4 % 0;`)

instead

of a variable would not cause a compiler error.

=====

Question:

What happens when you have this in your code:

`double x;`

`x = 24.0/0;`

Answer:

Compiles and runs

Explanation:

Floating point numbers don't produce a divide-by-zero
ArithmeticException.
They will give a result which is a Not a Number value.
=====

Question:

What happens when you bit shift by a number greater
than or equal to the number of bits in the result?
(eg. `int c = 270; c >> 33`)

Answer:

You get `270 >> 1`

Explanation:

Shifting bits by a number greater than or equal to the number of bits
in the result (eg. 32 bits for an int) will cause the value to be shifted
by the number modulo the number of bits in the value (you shift by `33%32`
which is 1).
=====

Question:

What happens when you compare two primitives of different
numerical types?

Answer:

Smaller type is promoted

Explanation:

Java promotes the smaller type to the wider of the types being compared.
So when comparing a byte to an int, the byte is first promoted to an int.
=====

Question:

Which of these is the Exclusive-OR (XOR) operator?

Answer:

^

Explanation:

The ^ is the Exclusive-OR operator.
=====

Question:

`x = false; y = true;`
What is the result of `x && y`?

Answer:

false

Explanation:

"&&" is the short-circuit AND operator. If both values
are true, the result is true, but Java won't bother to check the second
condition if the first one is false.
=====

Question:

Can a method have more than one access modifier?

Answer:

No

Explanation:

Methods CAN have more than one modifier, but not more than one ACCESS
modifier. Access modifiers are public, protected, private (and of course,

default/package). Early versions of the JDK allowed a private protected access, but current versions prohibit this.

=====

Question:

Can a subclass access the private member variables of its own superclass?

Answer:

No

Explanation:

Private means "private to the class". Period. Kids (subclasses) get no special treatment.

=====

Question:

Can a method with no access modifier be overridden by a method marked protected?

Answer:

Yes

Explanation:

Overridden methods are allowed to have LESS restriction, and since protected is less restricted than default (package), this is allowed.

=====

Question:

Does a final member variable have to be initialized at the time it's declared?

Answer:

No

Explanation:

Although early versions of the JDK enforced this, current versions let you initialize a final variable in the constructor, but no later.

=====

Question:

If a final variable is a reference to an object, can that object's data members be modified while the program is running?

Answer:

Yes

Explanation:

An reference variable marked final does NOT mean that the object itself can not be changed... final means that the reference can never be changed to point to some other object.

=====

Question:

Can a final method be overridden?

Answer:

No

Explanation:

Final methods can't be overridden... they're FINAL! The compiler will catch you.

=====

Question:

Can you automatically convert a long to an int if the long value is small enough to fit into an int?

Answer:

No

Explanation:

A long has 64 bits while an int has 32. You can't implicitly squish a *potentially* big thing into a little thing. The compiler doesn't care if the long variable is holding a tiny value. It just looks for the possibility of trouble.

=====

Question:

Can you automatically/implicitly convert a char to a short?

Answer:

No

Explanation:

They're the same bit-depth, but since chars are unsigned, they might have a higher positive value than a short can accept.

=====

Question:

Can you pass a short into a method call when an int is expected?

Answer:

Yes

Explanation:

Since a short will definitely fit into an int, the short is implicitly promoted. No cast needed.

=====

Question:

Given this code, what prints?

```
boolean b = false;
if ( b = true )
{
    System.out.println("yes");
}
```

Answer:

"yes" prints

Explanation:

"Yes" prints because the expression (b = true) uses the assignment operator =, not the equality operator ==. So b is set to true, and the result is true so the if condition runs the code.

=====

Question:

If you do not use curly braces for a while loop body, what will execute if the while condition is true?

Answer:

First statement after while ()

Explanation:

If there are no curly braces to set off a loop block, then just

the first statement is executed as part of the condition and any statements following that are executed no matter what.

=====

Question:

using a break in a for loop causes the loop to break out of the current iteration and jump to the next iteration of the loop.

Answer:

false

Explanation:

The break statements means "break out of this structure completely!". So the current iteration and any other remaining iterations of the loop will not run. Execution moves to the first statement below the loop.

=====

Question:

An exception can be caught by a catch block with an argument type that is a superclass of the thrown exception.

Answer:

true

Explanation:

If there is no catch(ExceptionArgument) which matches the thrown exception, then an argument which is a superclass of the exception will catch it.

=====

Question:

In a try/catch, an exception will not be considered caught if the catch (Exception) block has no code within the braces { }

Answer:

false

Explanation:

Its perfectly legal to catch the exception but do nothing about it.

=====

Question:

if an exception is not caught, the finally block will run and the rest of the method is skipped.

Answer:

true

Explanation:

The finally block will always run if an exception is thrown, and then the exception is immediately passed to the calling method.

=====

Question:

An overloaded method MUST change:

Answer:

Argument list

Explanation:

An overloaded method must change the argument list, but the return

type is optional.

=====

Question:

Overloaded methods are free to change the access modifier.

Answer:

true

Explanation:

As long as the argument list is different, the overloaded method is also free to change both the return type and the access visibility.

=====

Question:

constructors can not be overloaded.

Answer:

false

Explanation:

Constructors can be overloaded to allow the calling code to provide more (or less) information when the object is created.

=====

Question:

An inner class has free access to private member data of the outer classes.

Answer:

true

Explanation:

Inner classes have a special relationship with the their outer (enclosing) classes, including access to private data of the outer class. For event listeners implemented as inner classes, this means you do NOT have to pass a reference to the outer class in to the constructor of the inner class. The inner class has something like an "outer this" ("outer this" is not a keyword, just the way inner classes behave).

=====

Question:

Member (instance) variables are always assigned a default value if not explicitly initialized.

Answer:

true

Explanation:

Member variables are always given a default value -- numerical primitives get zero, chars get '\u0000', and object references get null.

=====

Question:

Automatic (local) variables are always assigned a default value if not explicitly initialized.

Answer:

false

Explanation:

You don't have to initialize a local variable, but as soon as you try to use it, you will get a compiler error because local variables aren't given default values.

=====

Question:

When you pass a variable as an argument to a method call, what are you passing?

Answer:

a copy of the value

Explanation:

You always get a copy of whatever is in the variable -- either a primitive or a reference. So for objects, you get a copy of the reference.

=====

Question:

When you pass an object reference as an argument to a method call, what gets passed?

Answer:

a copy of the reference

Explanation:

You always get a copy of whatever is in the variable -- either a primitive or a reference. So for objects, you get a copy of the reference.

=====

Question:

When using the right-shift operator (>>), what happens to the bits which move off the right end?

Answer:

They're gone

Explanation:

Bits moved off the end of any shift operation are just gone. History. Vaporized.

=====

Question:

x = true;

y = true;

What is the result of x ^ y?

Answer:

false

Explanation:

" ^ " is the Exclusive-OR operator -- it only returns true if just ONE of the values is true.

=====

Question:

x = true;

y = false;

What is the result of x & y?

Answer:

false

Explanation:

"&" is the AND operator. For a logical expression, the result will be true only if BOTH values are true. (For bitwise AND, both bits must be 1)

=====

Question:

Can one object access a private variable of another object of the same class?

Answer:

Yes

Explanation:

Private means "private to the class", NOT "private to the object". So two objects of the same class could access each other's private data.

=====

Question:

For object x to access a method of object y, when the method has no access modifier, object x MUST be an instance of a class which is:

Answer:

In the same package as y

Explanation:

No access modifier really means "default" access which really means "only to others in the package". But you'll never see a keyword/modifier "default".

=====

Question:

static variables are implicitly final.

Answer:

false

Explanation:

Static variables CAN be marked final, which makes them essentially constants, but they certainly don't have to be. Static just means "one per class... not one per instance"

=====

Question:

Is it legal to access a static method using an instance of the class?

Answer:

Yes

Explanation:

Its legal to say anObject.aStaticMethod() but the preferred way is to use the class name - MyClass.aStaticMethod(), to make it obvious in your code that you're calling a static method.

=====

Question:

Are static methods allowed to access instance variables using this.var?

Answer:

No

Explanation:

Within a static method, there is no 'this'! Since this refers to the current object -- the object whose method is executing -- 'this' makes no sense in a static method, since it doesn't belong to any object.

=====

Question:

The only things which can be defined inside a class and outside a method are instance variables and inner classes.

Answer:

false

Explanation:

Besides instance variables and inner classes, you can also define static code blocks. The compiler combines static blocks and static variable initialization into one big method which runs when the class is loaded.

=====

Question:

Can a constructor be declared private?

Answer:

Yes

Explanation:

So how do you make an object...? Obviously you'd need a less restrictive method (say, public) which can be called to create and return a new object.

=====

Question:

Can a constructor be declared abstract?

Answer:

No

Explanation:

Since a constructor is used solely to create a new instance, and an abstract class can NEVER be instantiated, it makes no sense.

=====

Question:

Can a constructor be declared static?

Answer:

No

Explanation:

Constructors are for creating new instance objects. Static methods are for non-instance code, so it makes no sense to have a static constructor.

=====

Question:

Is this legal?

```
float f = 2;
```

Answer:

Yes

Explanation:

The 2 is treated as an int literal, and since an int fits in a float, no explicit cast or 'f' is needed. If the literal was 2.0 (instead of 2), it would be treated as a double and would not compile.

=====

Question:

Can you automatically convert an int to a double?

Answer:

Yes

Explanation:

Any int can fit into a double-sized variable, so no cast is needed.

=====

Question:

Can you pass a double to a method when the method's argument is a float?

Answer:

No

Explanation:

Regardless of the double value, a double-sized variable is too big to fit in a float, so it can't be implicitly cast.

=====

Question:

Class Y is a subclass of class X. Will this compile?

```
Y myY = new Y();
```

```
X myX = myY;
```

Answer:

Yes

Explanation:

Any object can always be referenced by a variable which is of a superclass

type... because the child/subclass object is guaranteed to have the methods

of the superclass.

=====

Question:

using a break in a for loop causes the loop to break out of the current iteration and jump to the next iteration of the loop.

Answer:

false

Explanation:

The break statements means "break out of this structure completely!".

So the current iteration and any other remaining iterations of the loop will not run. Execution moves to the first statement below the loop.

=====

Question:

Is this legal?

```
int i = 0;
```

```
if ( i ) {}
```

Answer:

No

Explanation:

The only thing that can be an if (expression) is a boolean expression, and since in Java ints cannot be cast to booleans, if (anInteger) will not compile.

=====

Question:

In a switch statement, the argument to the case label (case: argument) can be any variable which can fit within an int.

Answer:

false

Explanation:

The case argument must be either an int literal, or an int-compatible variable which is a constant (i.e. static final).

=====
Question:

In a switch construct, once a matching case value is found, execution will continue through other case labels until it reaches a break statement.

Answer:

true

Explanation:

Code executing as part of a case statement will just keep on running through other cases until/unless it reaches a break statement. This is called "fall through".

=====
Question:

In a switch construct, the default statement must be placed after all the case statements.

Answer:

false

Explanation:

The default can be placed anywhere.

=====
Question:

All exceptions inherit from:

Answer:

java.lang.Throwable

Explanation:

The Exception hierarchy begins at java.lang.Throwable.

=====
Question:

If method A declares that it throws an exception and method B calls method A, then method B must ALWAYS declare the same exception.

Answer:

false

Explanation:

Method B does not have to declare the exception if it handles it (by providing an appropriate catch(theException), OR if the exception declared by A is a runtime exception.

=====
Question:

Class A inherits from B. B includes a method with the same name, return type, and arguments as a method in A.

This is an example of:

Answer:

Overriding

Explanation:

An overriding method must have the same name, return type, and arguments as the method in the superclass. It can't declare new or broader checked exceptions, and it also can't have more restrictive access.

=====

Question:

a class can define two methods with the same name as long as the return types are different.

Answer:

false

Explanation:

An overloaded method can change the return type, but return type alone is not enough -- it also must change the argument list.

=====

Question:

an overriding method of a subclass can throw an ArithmeticException not thrown by the original (overridden) method.

Answer:

true

Explanation:

ArithmeticException is a RuntimeException, so the rules about overriding methods not throwing new or broader exceptions does not apply.

=====

Question:

overloaded methods must not throw new checked exceptions not thrown by the original method.

Answer:

false

Explanation:

This would be true for overriding methods, but overloaded methods are free to throw new checked exceptions.

=====

Question:

an inner class has free access to ALL member data of its enclosing (outer) class.

Answer:

true

Explanation:

An inner class has a special relationship and can see even private member data of the outer class.

=====

Question:

It is not possible to have an instance of a non-static inner class before any instances of the outer class have been created.

Answer:
true

Explanation:

A non-static inner class can ONLY exist as part of a relationship with an instance of its enclosing class.

=====

Question:

This is a legal way to create an instance of an inner class:

```
OuterClass.InnerClass x =  
    new OuterClass().new InnerClass();
```

Answer:
true

Explanation:

It may look strange, but that syntax is how you can create an instance of an inner class when you don't yet have an instance of the outer class. An inner class must ALWAYS be tied to an instance of the outer class.

=====

Question:

```
public class Outer  
{  
    int x ;  
    static class Inner  
    {  
        void foo()  
        {  
            x = 5 ;  
        }  
    }  
}
```

Answer:
does not compile

Explanation:

A static inner class is treated as a top-level nested class, and has no reference -- no special relationship -- to any instance of the enclosing class.

=====

Question:

Class B extends A. B overrides the doStuff() method inherited from A. Which methods gets called if you say:
A myA = new B();
myA.doStuff();

Answer:
doStuff() in B

Explanation:

Java's polymorphism causes instance methods to be bound (chosen) at runtime, not during compilation. At runtime, the overriding method of the 'actual' object (which is a B) is dynamically chosen, even though the reference type is A.

=====

Question:

Are you allowed to have more than one top-level (non-inner) class definition per source file?

Answer:
Yes

Explanation:
You can have as many top-level classes in one source file as you like, but only one can be declared public and the source file name must match the public class name.

=====

Question:
Each source file must have a public class in it.

Answer:
No

Explanation:
You don't have to have any public classes in your source files.

=====

Question:
Assume the bit value of byte x is 00101011.
What is the result of x >> 2;

Answer:
00001010

Explanation:
The right shift operator fills from the left with the value of the sign bit (in this case, 0).

=====

Question:
Assume the bit pattern of byte x is: 10110001.
What will the sign of x be after x >> 2;

Answer:
negative

Explanation:
The right-shift operator fills all bits from the left with the same value as the original sign bit -- in this case 1.

=====

Question:
What does the unsigned right-shift operator do to a negative number?

Answer:
change its sign

Explanation:
The unsigned right-shift operator always fills from the left with zero, so the result will always become positive.

=====

Question:
What is the result of 00110011 & 11110000 ?

Answer:
00110000

Explanation:

The bitwise AND operator, &, will result in bit values of 1 only if BOTH bits being compared are 1.

=====

Question:

The short-circuit operators work on boolean types, not bits.

Answer:

true

Explanation:

Short-circuit operators work only on logical expressions; they make boolean tests more efficient by not testing both operands unless its necessary.

=====

Question:

Methods which have no access modifier can only be accessed by classes within the same package.

Answer:

true

Explanation:

No access modifier means "default" which means "only to others in the package".

=====

Question:

methods which are marked protected can be accessed only by classes within the same package.

Answer:

false

Explanation:

Protected means "any subclass, regardless of package, can inherit these members -- methods or data"

=====

Question:

Can a public method be overridden by a protected method?

Answer:

No

Explanation:

Overriding methods can NOT be more restrictive than the original method.

=====

Question:

methods and variables can be declared as final, but a class cannot be declared final.

Answer:

false

Explanation:

A final class is a class which can NOT be subclassed. Its the end-of-the-line in the inheritance tree.

=====

Question:

Can a final class be subclassed?

Answer:

No

Explanation:

A final class means the end-of-the-line for the inheritance tree. No subclasses

can be made from a final class, only instances.

=====

Question:

Can you have a final abstract class?

Answer:

No

Explanation:

final and abstract are opposites: a final class can NOT be subclassed, while an abstract class MUST be subclassed... so final abstract makes no sense.

=====

Question:

Can a member variable be declared abstract?

Answer:

No

Explanation:

Only methods and classes can be declared abstract.

=====

Question:

Can an abstract method be overridden?

Answer:

Yes

Explanation:

An abstract method has no choice! It MUST be overridden.

=====

Question:

Can a class be declared native?

Answer:

No

Explanation:

Methods can be native, not classes or variables.

=====

Question:

Can a member variable be declared native?

Answer:

No

Explanation:

Only methods can be native.

=====

Question:

Can a method be declared transient?

Answer:

No

Explanation:

The transient modifier is used to keep variable data from being saved if an object is "serialized" (saved as an object). Since methods don't change their state for an individual object, methods are never serialized, so transient is not needed for methods.

=====

Question:

Can a class be declared transient?

Answer:

No

Explanation:

Only data members can be declared as transient... meaning their value won't be saved when an object is "serialized" (saved).

=====

Question:

Can a member variable be declared synchronized?

Answer:

No

Explanation:

The word synchronized marks method code so that it can't be run by more than one thread at a time.

=====

Question:

Can a class be declared synchronized?

Answer:

No

Explanation:

The keyword synchronized marks method code so that only one thread can access it at a time.

=====

Question:

Can a method be declared synchronized?

Answer:

Yes

Explanation:

The synchronized modifier means that only one thread at a time can run the synchronized code.

=====

Question:

What happens when you compile this:

byte a = 2;

a++;

Answer:

Compiles and runs

Explanation:

The ++operator increments a as a byte, so there's no problem. You WOULD have a problem with a = a + 1; because the a + 1 would result in an int

which would then have to be cast back to the byte variable a.

=====

Question:

Class Y is a subclass of class X. Will this compile?

X myX = new X();

Y myY = (Y)myX;

Answer:

Yes

Explanation:

It will compile, because the myX object might really have been a Y object all along (as far as the compiler knows), but at runtime it will fail.

=====

Question:

In a switch statement, the argument to the switch () can be a variable as long as it is compatible with an int.

Answer:

true

Explanation:

The switch can obviously hold a variable, because that's what you're testing! But the variable cannot be anything except an int-compatible primitive.

=====

Question:

in a switch construct, the default statement will always execute.

Answer:

false

Explanation:

The default statement will only execute if no matching case is found, OR if there is "fall through" from a case statement executed above.

=====

Question:

in a switch construct, the default statement will execute if no case values match the switch() argument.

Answer:

true

Explanation:

If there are no matching cases found, the default will run.

=====

Question:

in a try/catch construct, the catch blocks may be written in any order and the most narrow/specific catch argument is chosen at runtime.

Answer:

false

Explanation:

The catch blocks must be written in order from most specific to least (e.g. IOException comes above Exception) or the compiler will complain.

=====

Question:

a finally block will only be executed if the exception is not caught.

Answer:

false

Explanation:

The finally block will ALWAYS run... unless the try/catch calls System.exit().

=====

Question:

a MalformedURLException is a checked exception.

Answer:

true

Explanation:

A MalformedURLException is checked by the compiler to be sure that the "handle or declare" rule is followed.

=====

Question:

an ArithmeticException is a checked exception.

Answer:

false

Explanation:

ArithmeticExceptions are considered programmer problems, and are NOT checked by the compiler. A divide-by-zero error, for example, should be checked for by your own code.

=====

Question:

if a try/catch construct calls System.exit(), the finally block will execute first.

Answer:

false

Explanation:

System.exit() means stop the program now! Even a finally block can't stop that from happening.

=====

Question:

an overriding method can change the access modifier from default to protected.

Answer:

true

Explanation:

Since the protected modifier is less restrictive than default, an overriding method is allowed to make that change.

=====

Question:

To invoke an overridden method from the method which overrides it, use:

Answer:

```
super.theMethod();
```

Explanation:

The super keyword is like a reference to the superclass.

=====

Question:

constructors are not inherited.

Answer:

true

Explanation:

Constructors are different from methods in several ways, and inheritance is one of them. Subclasses do NOT get a constructor from the superclass.

=====

Question:

It is legal to declare an array with an interface as the type (instead of a class or primitive) -- example:
PetInterface []p;

Answer:

true

Explanation:

One of the best uses of polymorphism in Java is to declare an array using an interface type, then add objects which can be from any class, as long as they implement the interface type of the array.

=====

Question:

Can a method be declared volatile?

Answer:

No

Explanation:

Only variables can be declared as volatile.

=====

Question:

Is cast a keyword?

Answer:

No

Explanation:

It may have been in c++, but wasn't carried into Java.

=====

Question:

What is the result of the following code
System.out.println(Math.round(-3.22));

Answer:

-3

Explanation:

round returns the closest int to the argument.

=====

Question:

What is the result of the following code?
System.out.println(Math.ceil(-3.22));

Answer:
-3.0

Explanation:
ceil returns the smallest (closest to negative infinity) double value that is not less than the argument and is equal to a mathematical integer.

=====

Question:
What is the result of the following code?
System.out.println(Math.floor(-3.22));

Answer:
-4.0

Explanation:
floor returns the largest (closest to positive infinity) double value that is not greater than the argument and is equal to a mathematical integer.

=====

Question:
What is the result of the following code?
System.out.println(Math.sqrt(-4));

Answer:
NaN

Explanation:
If the argument is NaN or less than zero, then the result is NaN.

=====

Question:
Will this compile?
int iArray[] = new int[] {1,2,3,4,5};

Answer:
Yes

Explanation:
This is one way to initialize an array when it is declared.

=====

Question:
Will this compile?
int iArray[5]= new int[]{1,2,3,4,5};

Answer:
No

Explanation:
An array declaration may not have the size on the left hand side of the equals sign.

=====

Question:
Assuming all variables are declared and initialized properly, will this compile?
for (i = 1, j = 2; i < 3; i++, j++){};

Answer:
Yes

Explanation:

Multiple initializers and multiple incrementers are permitted.

=====

Question:

Assuming all variables are declared and
initialized properly, will this compile?
for (i = 1; k < 5; j++) {};

Answer:

Yes

Explanation:

Multiple variables are permitted within the construct of a
for loop (although this is not recommended)

=====

Question:

if you do not provide any constructor for
your class, the compiler will insert
a default constructor with public access modifier

Answer:

false

Explanation:

The generated constructor is given the access modifier
"public" if the class is public, or else the constructor
is given the default access implied by no modifier

=====

Question:

An interface can extend at most one interface.

Answer:

false

Explanation:

An interface may extend multiple interfaces. A class is,
however, limited to extending only one class.

=====

Question:

An abstract class can have a constructor.

Answer:

Yes

Explanation:

An abstract class may have a constructor. Since an abstract
class cannot be instantiated, you cannot call its constructor
directly. However, a constructor will be called implicitly
when an abstract class is extended

=====

Question:

If there is a return statement in the catch block,
the finally block will not be executed.

Answer:

false

Explanation:

The ONLY case when the finally block will not be executed
is if there is a call to System.exit() in the try/catch blocks.

=====

Question:

what will be the output of this code:

```
class Exam
{
    protected String difficultyLevel = "easy";

    public void printDifficultyLevel()
    {
        System.out.println( difficultyLevel );
    }
}
class SCJPEXam extends Exam
{
    private String difficultyLevel = "killing";
}
SCJPEXam myExam = new SCJPEXam();
myExam.printDifficultyLevel();
```

Answer:

easy

Explanation:

Methods can be overridden. Attributes cannot.

=====

Question:

An anonymous inner class cannot have a constructor

Answer:

true

Explanation:

A constructor has the same name as the class. If the class is anonymous (has no name), how could you name the constructor?

=====

Question:

A method cannot be declared as throwing more than one exception

Answer:

false

Explanation:

Any number of exceptions, separated by commas, can be coded in a throws clause

=====

