**PluralSight -JavaFundamental**

1.What code do you need to add to a class Collection so that all the extensions of this class provide their own implementation of a method size()?s

* Your choice: correct -

abstract int size();

* Incorrect -

default int size();

* Incorrect -

@Override int size();

* Incorrect -

super int size();

* Incorrect -

I don't know yet.

2.You want to package an application composed of many class files in a single Java Archive (JAR) and make it executable. How will you proceed?

* Incorrect -

Create a single JAR file containing a MANIFEST.mf file with a reference to all the classes of that JAR.

* Your choice: incorrect -

Create one JAR file per module, put a MANIFEST.mf with a reference to the class that contains the main() method in one of them, and launch that JAR.

* Incorrect -

Package each module in a JAR, then package them in a meta-JAR, put a reference to the class that contains the main() method in this meta-JAR, and launch that JAR.

* Correct -

Create a single JAR file containing a MANIFEST.mf file with a reference to the class that contains the main() method.

* Incorrect -

I don't know yet.

3.Which of the following does not apply to Optional instances?

* Incorrect -

An optional can be empty

* Correct -

An optional cannot wrap a null object

* Incorrect -

An optional can wrap an empty string

* Incorrect -

An optional can wrap an empty list

* Incorrect -

I don't know yet.

4.How can you create a stream from the elements of a List?

* Incorrect -

By calling List.streamOf(list)

* Incorrect -

By calling Stream.of(list)

* Your choice: correct -

By calling list.stream()

* Incorrect -

By calling Arrays.stream(list.toArray())

* Incorrect -

I don't know yet.

5.How many iterations are guaranteed in a while loop?

* Incorrect -

An infinite number until a break statement is encountered

* Incorrect -

Two

* Incorrect -

One

* Your choice: correct -

Zero

* Incorrect -

I don't know yet.

6.You have the following class:

public class Holder {

   static String MAX\_SIZE = System.getProperty("max.size");

}

You execute the following code:

Holder holder1 = new Holder();

Holder holder2 = new Holder();

How many times is the getProperty("max.size") method called?

* Incorrect -

0

* Incorrect -

This code throws an exception at runtime.

* Your choice: correct -

1

* Incorrect -

2

* Incorrect -

I don't know yet.

7.How can you capitalize the first character of an instance of String?

* Incorrect -

Get access to the internal array of characters and capitalize the first one.

* Incorrect -

Call the substring() method to extract the first character and chain this call with a toUpperCase() method call.

* Correct -

Extract the first character, capitalize it, then concatenate it with the substring of the initial string, starting at character 1.

* Incorrect -

Call the toUpperCase() method and pass the index of the first character as a parameter.

* Incorrect -

I don't know yet.

8.How would you iterate over the key-value pairs from an instance of Map?

* Incorrect -

Get the Set of key value pairs with the getEntrySet() method

* Incorrect -

Get an iterator with the iterate() method

* Incorrect -

Get an iterator with the getIterator() method

* Your choice: correct -

Get the Set of key value pairs with the entrySet() method

* Incorrect -

I don't know yet.

9.Which code snippet will compile?

* Incorrect -

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

* Incorrect -

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

* Incorrect -

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

* Your choice: correct -

None of the snippets will compile

* Incorrect -

I don't know yet.

10.You have the following classes:

public class HelloHolder {

    public String get() { return "Hello"; }

    public boolean isEmpty() { return false; }

}

public class NullHolder {

    public String get() { return null; }

    public boolean isEmpty() { return true; }

}

Which interface is the most appropriate implementation for creating an abstraction of these two given classes?

* Incorrect -

public interface Holder {

   String get();

}

* Incorrect -

public interface Holder {

   default String get() { return null; }

   default boolean isEmpty() { return get() == null; }

}

* Incorrect -

public interface Holder {

   default boolean isEmpty() { return get() == null; }

}

* Your choice: correct -

public interface Holder {

   String get();

   default boolean isEmpty() { return get() == null; }

}

* Incorrect -

I don't know yet.

11.What is the default order used to compare strings of characters?

* Your choice: incorrect -

The order compares the ASCII code of the characters, which depends on the encoding of the string.

* Incorrect -

The length of the strings.

* Incorrect -

There is no default order to compare strings of characters.

* Correct -

Strings are compared lexicographically. For example, the Unicode values of the strings are compared.

* Incorrect -

I don't know yet.

12.What functional interface is not defined in the java.util.function package?

* Your choice: correct -

The Runnable interface

* Incorrect -

The Predicate interface

* Incorrect -

The BinaryOperator interface

* Incorrect -

The Consumer interface

* Incorrect -

I don't know yet.

13.What does a class need to do when it declares that it is implementing an interface?

* Incorrect -

It must implement each of the interface’s abstract methods and override the default methods.

* Incorrect -

It must implement each of the interface's abstract methods.

* Correct -

It must be declared as abstract OR implement each of the interface's abstract methods, apart from the methods of the Object class.

* Incorrect -

It must be declared as abstract.

* Incorrect -

I don't know yet.

14.You have the following interface:

public interface Holder {

   String get();

}

This interface is implemented by a class called FieldHolder.

You need to create a factory method that returns an instance of Holder. How are you going to write it?

* Incorrect -

static FieldHolder instance() {    return new Holder(); }

* Incorrect -

static FieldHolder instance() {    return new FieldHolder(); }

* Correct -

static Holder instance() {

   return new FieldHolder();

}

* Incorrect -

static Holder instance() {    return new Holder(); }

* Incorrect -

I don't know yet.

15.Given that strings is an ArrayList with the following elements: "one", "two", and "three", what does the following code do?

for (String s: strings) {

   System.out.println("s = " + s);

   strings.remove("two");

}

* Incorrect -

The code throws an IllegalStateException.

* Incorrect -

The code prints "one" and then "three".

* Incorrect -

The code throws an ArrayOutOfBoundException.

* Your choice: correct -

The code throws a ConcurrentModificationException.

* Incorrect -

I don't know yet.

16.Why should you use the generic version of the interfaces from the Collection application programming interface (API)?

* Incorrect -

Because it brings better performances

* Correct -

Because it brings type safety to your application

* Your choice: incorrect -

Because it saves you from creating complex class hierarchy

* Incorrect -

Because it creates the interfaces you need automatically

* Incorrect -

I don't know yet.

17.Among the following Exceptions, which one must you handle by catching it or transmitting it to the calling code?

* Your choice: correct -

IOException

* Incorrect -

OutOfMemoryError

* Incorrect -

NumberFormatException

* Incorrect -

NullPointerException

* Incorrect -

I don't know yet.

18.What structure of the Collection API can guarantee that the elements are iterated over in the same order in which they were added?

* Correct -

The List

* Incorrect -

The Set

* Your choice: incorrect -

The SortedSet

* Incorrect -

The OrderedSet

* Incorrect -

I don't know yet.

1.

You have the following class:

public class Holder {

   static String MAX\_SIZE = System.getProperty("max.size");

}

You execute the following code:

Holder holder1 = new Holder();

Holder holder2 = new Holder();

How many times is the getProperty("max.size") method called?

* Incorrect -

This code throws an exception at runtime.

* Incorrect -

2

* Your choice: correct -

1

* Incorrect -

0

* Incorrect -

I don't know yet.

2.

You have the following classes:

public class HelloHolder {

    public String get() { return "Hello"; }

    public boolean isEmpty() { return false; }

}

public class NullHolder {

    public String get() { return null; }

    public boolean isEmpty() { return true; }

}

Which interface is the most appropriate implementation for creating an abstraction of these two given classes?

* Incorrect -

public interface Holder {

   default boolean isEmpty() { return get() == null; }

}

* Your choice: correct -

public interface Holder {

   String get();

   default boolean isEmpty() { return get() == null; }

}

* Incorrect -

public interface Holder {

   String get();

}

* Incorrect -

public interface Holder {

   default String get() { return null; }

   default boolean isEmpty() { return get() == null; }

}

* Incorrect -

I don't know yet.

3.

You need to call a method that throws IOException and FileNotFoundException. Knowing that FileNotFoundException is an extension of IOException, among the following catch clauses, which one is NOT correct?

* Incorrect -

catch (FileNotFoundException e) {} catch (IOException e) {}

* Your choice: correct -

catch (FileNotFoundException | IOException e) {}

* Incorrect -

catch (IOException e) {}

* Incorrect -

catch (Exception e) {}

* Incorrect -

I don't know yet.

4.

You must test an Android application written in Java on a development machine running Linux. What tool do you need?

* Incorrect -

A Java Runtime Environment running in Android Studio

* Incorrect -

A real phone, connected to your development machine through the USB interface

* Your choice: correct -

An Android Virtual Device, such as the ones available through Android Studio

* Incorrect -

A Java Development Kit running in Android Studio

* Incorrect -

I don't know yet.

5.

You have a class Person with an age property of type int and its corresponding getter getAge(). You have a stream of instances of this class and want to divide this stream into two lists: the first containing the people younger than 20, and the other the people older than 20. Which collector can you use?

* Incorrect -

Collectors.groupingBy(Person::getAge, comparing(age -> age > 20))

* Your choice: correct -

Collectors.partitioningBy(p -> p.getAge() > 20, toList())

* Incorrect -

Collectors.groupingBy(Person::getAge, comparing(p -> p.getAge() > 20))

* Incorrect -

Collectors.partitioningBy(age -> age > 20, toList())

* Incorrect -

I don't know yet.

6.

You have written a method that takes an index (of type int) in a list as a parameter. Which of the following properly documents this parameter?

* Your choice: correct -

@param index the index in the list

* Incorrect -

@parameter int index the index in the list

* Incorrect -

@parameter index the index in the list

* Incorrect -

@param int index the index in the list

* Incorrect -

I don't know yet.

7.

Which of the following does not apply to a stream?

* Incorrect -

A stream can act as a buffer of elements.

* Incorrect -

A stream must connect to a source of data.

* Your choice: correct -

A stream can only process objects.

* Incorrect -

A stream must not modify the source it is connected to.

* Incorrect -

I don't know yet.

8.

What is the compile-time error for the following code?

public class Holder {

   static enum Civility {

      MRS("Mrs."), MR("Mr.");

      static Map<String, Civility> map = new HashMap<>();

      String label;

      Civility(String label) {

          this.label = label;

          map.put(this.label, this);

      }

      public static Civility of(String label) {

         return map.get(label);

      }

   }

}

* Your choice: correct -

You cannot access static members from an enum constructor.

* Incorrect -

The constructors of an enum must be private.

* Incorrect -

An inner enum is static by default.

* Incorrect -

All the fields of an enum should be final.

* Incorrect -

I don't know yet.

9.

You need to create a Function that searches a string of characters for a specific word. You then create a BiFunction that takes the string and the searched word. How can you make it a Function that takes only the string?

* Correct -

By using partial application on this BiFunction with the searched word.

* Incorrect -

By composing the search function and this BiFunction.

* Your choice: incorrect -

By chaining the search function and this BiFunction.

* Incorrect -

By creating a function that calls this one with both parameters.

* Incorrect -

I don't know yet.

10.

You have a Person class with two properties: name and age. You have the following code:

Comparator<Person> cmp =

   Comparator.comparing(Person::getName)

             .thenComparing(Person::getAge);

Given that Comparator<T> is a functional interface with the following abstract method:

int compare(T t1, T, t2);

What is the declaration of the thenComparing() method?

* Incorrect -

default <U extends Comparable<? super U>> Comparator<T> thenComparing(Predicate<? extends T> p)

* Correct -

default <U extends Comparable<? super U>> Comparator<T>  thenComparing(Function<? super T, ? extends U> f)

* Incorrect -

default <U extends Comparable<? super U>> Comparator<T>  thenComparing(Predicate<? super T> p)

* Your choice: incorrect -

default <U extends Comparable<? super U>> Comparator<T>  thenComparing(Function<? extends T, ? super U> f)

* Incorrect -

I don't know yet.

11.

What is the type of a map created from a stream of strings of characters with the following collector: Collectors.groupingBy(String::length, counting())?

* Incorrect -

Map<Integer, List<Long>>

* Correct -

Map<Integer, Long>

* Incorrect -

Map<Integer, Integer>

* Your choice: incorrect -

Map<Integer, List<String>>

* Incorrect -

I don't know yet.

12.

You have a stream of integers. You call reduce(Integer::max) on this stream. What does it return?

* Incorrect -

It does not compile. You must pass Integer.MIN\_VALUE as the first parameter of this call.

* Correct -

An optional containing the value of the biggest integer.

* Incorrect -

The value of the biggest integer.

* Incorrect -

It does not compile. You must pass 0 as the first parameter of this call.

* Incorrect -

I don't know yet.

13.

What is the difference between a domain model built using class composition and a domain model built using class extension?

* Correct -

It uses delegation instead of inheritance.

* Incorrect -

It uses abstract classes instead of inheritance.

* Your choice: incorrect -

It uses dependency injection instead of inheritance.

* Incorrect -

It uses factory methods instead of inheritance.

* Incorrect -

I don't know yet.

14.

Suppose you have the two following interfaces: A and B.

interface A {

   default int size() {

      return 10;

   }

}

interface B extends A {

  int size();

}

You then create the AB class:

public class AB implements A, B {

}

And then call the size() method on an instance of AB. What does it print out?

* Your choice: correct -

Nothing, because this code does not compile.

* Incorrect -

null

* Incorrect -

0

* Incorrect -

10

* Incorrect -

I don't know yet.

15.

You have the following functional interface:

public interface Converter {

   int convert(int i);

}

What code would allow for the composition of two converters?

* Incorrect -

default Converter compose(Converter other) {

   return  i -> other.convert(i);

}

* Your choice: correct -

default Converter compose(Converter other) {

   return  i -> convert(other.convert(i));

}

* Incorrect -

default Converter compose(Converter other) {

   return  i -> convert(other, convert(i));

}

* Incorrect -

default Converter compose(Converter other) {

   return  i -> convert(other, i);

}

* Incorrect -

I don't know yet.

16.

You have old source code in your application, written and compiled using Java 6. You have single method interfaces in this code. How can you implement the method of the interfaces using lambdas?

* Your choice: incorrect -

You need to add the @FunctionalInterface annotation on those interfaces and recompile this code with a Java 8 compiler.

* Incorrect -

You need to move those interfaces to a package named function.

* Incorrect -

You just need to recompile this old code with a Java 8 compiler.

* Correct -

In a given class, you can implement the interface methods by simply using the syntax of the lambda expressions.

* Incorrect -

I don't know yet.

17.

What types of exceptions must be specified in a method's throws clause?

* Your choice: incorrect -

All exceptions, except ones that extend RuntimeException

* Correct -

Only the exceptions that do not extend RuntimeException and do not extend Error

* Incorrect -

Only exceptions that extend RuntimeException

* Incorrect -

All the exceptions, except the ones that extend Error

* Incorrect -

I don't know yet.

18.

Why can't you call the following code in a class parameterized by the type E?

E e = new E();

* Your choice: incorrect -

Because the Java virtual machine (JVM) cannot verify the runtime value of the parameter E

* Incorrect -

Because it would lead to unacceptable performance hits

* Correct -

Because the type E is erased at compile time and replaced with Object

* Incorrect -

Because it could throw exceptions at runtime that cannot be seen at compile time

* Incorrect -

I don't know yet.

1.

You need to use an exception specific to your application instead of the standard IOException. Every time an IOException is thrown by some code in your application, you need to catch this exception and rethrow it, wrapped in your custom application exception. How can you preserve the IOException as the root cause of your custom exception?

* Your choice: correct -

Pass the root IOException as a parameter to the constructor of your custom application exception

* Incorrect -

Preserve the message by passing it as a parameter to the constructor of your custom application exception, but not the stack trace

* Incorrect -

Set the message and the stack trace as a property of your custom application exception

* Incorrect -

Log the message and the stack trace using the standard JVM logging mechanism

* Incorrect -

I don't know yet.

2.

You have a Person class with a name property of type String. You have a stream of instances of this class and collect this stream with the following collector: Collectors.mapping(Person::getName, joining(", ")).

What do you get as a result?

* Incorrect -

A string with the concatenation of the names of the people if they are not empty, separated by commas

* Incorrect -

A string with the concatenation of the names of the people if they are not null, separated by commas

* Incorrect -

A list of the names of the people

* Your choice: correct -

A string with the concatenation of the names of the people, separated by commas

* Incorrect -

I don't know yet.

3.

You have a map of Integer and String. You need to put all the values to uppercase. How are you going to proceed?

* Incorrect -

Call map.replaceAll(value -> value.toUpperCase()).

* Incorrect -

Call map.computeIfPresent((key, value) -> value.toUpperCase()).

* Correct -

Call map.replaceAll((key, value) -> value.toUpperCase()).

* Your choice: incorrect -

Call map.computeIfPresent(value -> value.toUpperCase()).

* Incorrect -

I don't know yet.

4.

Given that strings is an ArrayList with the following elements: "one", "two", and "three", what does the code below do?

for (String s: strings) {

   System.out.println("s = " + s);

   strings.remove("two");

}

* Incorrect -

The code throws an IllegalStateException.

* Incorrect -

The code throws an ArrayOutOfBoundException.

* Your choice: correct -

The code throws a ConcurrentModificationException.

* Incorrect -

The code prints "one" and then "three".

* Incorrect -

I don't know yet.

5.

You have a char c, an int i and a long l. All three variables have been initialized. Which one of the following affectations does not compile?

* Incorrect -

l = c

* Your choice: correct -

c = i

* Incorrect -

i = c

* Incorrect -

l = i

* Incorrect -

I don't know yet.

6.

Which of the following type is NOT valid within a switch statement's test expression?

* Incorrect -

enum

* Incorrect -

int

* Incorrect -

String

* Your choice: correct -

boolean

* Incorrect -

I don't know yet.

7.

You are designing a system in which several threads are going to add messages to a single wait list. Several consumers will take one message at a time and process it. If your wait list is empty, you want a consumer to wait until a message is available. What implementation are you going to choose?

* Incorrect -

LinkedList

* Incorrect -

ConcurrentLinkedQueue

* Your choice: correct -

ArrayBlockingQueue

* Incorrect -

ArrayDeque

* Incorrect -

I don't know yet.

8.

You have a class Person with an age property of type int and its corresponding getter getAge(). You have a stream of instances of this class and want to divide this stream into two lists: the first containing the people younger than 20, and the other the people older than 20. Which collector can you use?

* Incorrect -

Collectors.groupingBy(Person::getAge, comparing(age -> age > 20))

* Incorrect -

Collectors.groupingBy(Person::getAge, comparing(p -> p.getAge() > 20))

* Your choice: correct -

Collectors.partitioningBy(p -> p.getAge() > 20, toList())

* Incorrect -

Collectors.partitioningBy(age -> age > 20, toList())

* Incorrect -

I don't know yet.

9.

You have the following functional interface:

public interface Converter {

   int convert(int i);

}

What code would allow you to create a converter from Celsius to Fahrenheit degrees?

* Incorrect -

static Converter toFahrenheit(int celsius) {   return 9\*celsius/5 + 32;}

* Your choice: incorrect -

static Converter toFahrenheit() {   return celsius -> convert(9\*celsius/5 + 32);}

* Incorrect -

static Converter toFahrenheit() {   return celsius -> 9\*convert(celsius)/5 + 32;}

* Correct -

static Converter toFahrenheit() {   return celsius -> 9\*celsius/5 + 32;}

* Incorrect -

I don't know yet.

10.

You have written a method that takes an index (of type int) in a list as a parameter. Which of the following properly documents this parameter?

* Incorrect -

@parameter index the index in the list

* Incorrect -

@param int index the index in the list

* Your choice: correct -

@param index the index in the list

* Incorrect -

@parameter int index the index in the list

* Incorrect -

I don't know yet.

11.

You have a String of 10 characters. You write it into a binary file, character by character. What is the length of the resulting file?

* Incorrect -

The length is 11 bytes, since each character is encoded on 8 bits, and you need to add the final \0 char.

* Correct -

It is not possible to tell, since in Unicode, a single character can be encoded on more than one char.

* Your choice: incorrect -

The length is 20 bytes, since each character is encoded on 2 bytes.

* Incorrect -

The length is 10 bytes, since each character is encoded on 8 bits.

* Incorrect -

I don't know yet.

12.

You have a stream of integers with duplicates in it. From that stream, how can you create another stream of the same integers, sorted?

* Incorrect -

By calling the compare() method and pass it the right comparator

* Incorrect -

By calling the treeSet() method

* Your choice: correct -

By calling the sorted() method

* Incorrect -

By calling the method sort()

* Incorrect -

I don't know yet.

13.

How can you learn the type of lambda expression you need to pass to the List.forEach() method?

* Correct -

By checking the type of the parameter in the signature of the List.forEach() method

* Your choice: incorrect -

By checking the return type and the parameters of the function that needs to be passed as the lambda expression

* Incorrect -

By checking the return type in the signature of the List.forEach() method

* Incorrect -

By checking the documentation of the functional interfaces and seeing which one matches the signature of the List.forEach() method

* Incorrect -

I don't know yet.

14.

You have the following interface:

public interface Holder<E> {

   E get();

}

You need to add a factory method to this interface, which returns instances of the Holder interface. Which of the following snippets of code are you going to choose?

* Your choice: incorrect -

public static <E> Holder<E> holder(Object e) {

   return () -> e;

}

* Incorrect -

public static Holder holder(E e) {

   return () -> e;

}

* Correct -

public static <E> Holder<E> holder(E e) {

   return () -> e;

}

* Incorrect -

public static Holder<E> holder(E e) {

   return () -> e;

}

* Incorrect -

I don't know yet.

15.

How does a TreeMap store its key-value pairs?

* Your choice: correct -

It uses a red black tree.

* Incorrect -

It uses a skip list of key value pairs.

* Incorrect -

It uses a set of Map.Entry objects.

* Incorrect -

It puts the keys in an array with pointers to the values.

* Incorrect -

I don't know yet.

16.

In the following example, which ProductProcess can you make private?

* Incorrect -

public class Product {

    public void process(Product p) {

        class ProductProcess {}

    }

}

* Incorrect -

public class Product {}

class ProductProcess {

    // declared in Product.java

}

* Incorrect -

class ProductProcess {

    public class Product {}

}

* Your choice: correct -

public class Product {

    class ProductProcess {}

}

* Incorrect -

I don't know yet.

17.

You have the following interface Reader:

public interface Reader {

   void read(File file) throws IOException;

}

Among the following methods, which one is NOT an implementation of the read() method from this interface?

* Incorrect -

public void read(File file) throws IOException {}

* Correct -

public void read(File file) throws Exception {}

* Your choice: incorrect -

public void read(File file) {}

* Incorrect -

public void read(File file) throws FileNotFoundException {}

* Incorrect -

I don't know yet.

18.

You have the following functional interface:

public interface Converter {

   int convert(int i);

}

What code would allow for the composition of two converters?

* Your choice: incorrect -

default Converter compose(Converter other) {

   return  i -> other.convert(i);

}

* Incorrect -

default Converter compose(Converter other) {

   return  i -> convert(other, i);

}

* Correct -

default Converter compose(Converter other) {

   return  i -> convert(other.convert(i));

}

* Incorrect -

default Converter compose(Converter other) {

   return  i -> convert(other, convert(i));

}

* Incorrect -

I don't know yet.

1. What does it mean for a field to be final?

Your choice: incorrect -The field cannot be overridden in any extension this class may have.

Incorrect -The field is a constant, accessible from all the class instances.

Correct -Its value has to be set during the construction of the object, and cannot be modified.

Incorrect -Once set, its value can only be modified within the instance initializer block.

 2. Among the following, which one is NOT a property of Throwable?

Incorrect -The stack trace

Your choice: incorrect -The message

Incorrect -The cause

Correct -The severity

Incorrect -I don't know yet.

 3. You have a Java application packaged in a JAR file Application.jar. What is the command to run it?

Incorrect -

$ jar -java Application.jar

Incorrect -

$ java -run Application.jar

Incorrect -

$ java -start Application.jar

Your choice: correct -

$ java -jar Application.jar

Incorrect -I don't know yet.

 4. What code do you need to add to a class Collection so that all the extensions of this class provide their own implementation of a method size()?

Incorrect -

@Override int size();

Your choice: incorrect -

super int size();

Correct -

abstract int size();

Incorrect -

default int size();

Incorrect -I don't know yet.

 5. You need to create a method in a class to be used from within this class and from all the extending classes. What visibility modifier are you going to use?

Your choice: correct -

protected

Incorrect -

private

Incorrect -

default

Incorrect -No visibility modifier

Incorrect -I don't know yet.

6. You have two classes:

public class Holder {

    public String toString() {

        return "I am a holder";

    }

}

public class ExtendedHolder extends Holder {

    public String toString() {

           return "I am an extended holder";

    }

}

What does the following code print out when executed?

public class Main {

    public static void display(Holder holder) {

        Object o = holder;

        System.out.println(o.toString());

    }

    public static void main(String[] args) {

        Holder holder = new ExtendedHolder();

        display(holder);

    }

}

Incorrect -java.lang.Object@4e04a765 (the number may vary)

Correct -

I am an extended holder

Your choice: incorrect -I am an extended holderI am aa holderjava.lang.Object@4e04a765 (the number may vary)

Incorrect -

I am a holder

Incorrect -I don't know yet.

7. You have the following class:

public abstract AbstractCollection {

   public abstract int size();

   public boolean isEmpty() {

      return size() == 0;

   }

}

You are about to write a Collection class that will extend this class. Among the following statements, which one is true for your Collection class?

Your choice: incorrect -You need to override the isEmpty() method.

Incorrect -You need to create an empty field of type boolean.

Correct -You need to override the size() method.

Incorrect -You need to create a size field of type int.

Incorrect -I don't know yet.

8. How would you determine whether an instance of String is empty and it contains only blank characters (such as new lines or spaces)?

Incorrect -Call the isEmpty() method.

Correct -Call the trim() method and test the result with the isEmpty() method.

Incorrect -Compose the isEmpty() with the trim() call.

Your choice: incorrect -Call the isEmpty() method and test the result with the trim() method.

Incorrect -I don't know yet.

9. Which of the following is a valid declaration for an array that will hold 5 integers?

Correct -

int[] array = new int[5];

Incorrect -

int[5] array = new int[];

Your choice: incorrect -

int[] array = new array[5];

Incorrect -

int[] array = new Array(int)[5];

Incorrect -I don't know yet.

10. What Java API is not supported in Android?

Incorrect -Java I/O

Incorrect -Java String

Your choice: correct -Java Swing

Incorrect -Java Thread

Incorrect -I don't know yet.

11. You need to create a class which inherits from the type of an interface. What keyword do you use?

Incorrect -

inherits

Your choice: correct -

implements

Incorrect -

overrides

Incorrect -

extends

Incorrect -I don't know yet.

12. You created a Set of strings. You add the string "abc" three times to this set, then remove this string. What occurs when you call size() on this set?

Incorrect -It returns 2.

Your choice: incorrect -It throws an IllegalStateException.

Correct -It returns 0.

Incorrect -It returns 1.

Incorrect -I don't know yet.

13. Which statement is true?

Incorrect -An abstract class can declare a static abstract method.

Incorrect -An abstract class can declare an abstract constructor.

Your choice: correct -An abstract class can have abstract methods and non-abstract methods.

Incorrect -An abstract class can have abstract private methods.

Incorrect -I don't know yet.

14. Which is a legal class declaration given that ClassA and ClassB are classes and InterfaceA and InterfaceB are interfaces?

Incorrect -

class A implements InterfaceA, InterfaceB extends ClassA

Incorrect -

class A implements InterfaceA extends ClassA

Incorrect -

class A extends ClassA, ClassB implements InterfaceA, InterfaceB

Your choice: correct -

class A extends ClassB implements InterfaceA, InterfaceB

Incorrect -I don't know yet.

15. What is a wrapper type in Java?

Correct -A class that wraps a primitive value in an object

Incorrect -An interface that models a primitive value

Your choice: incorrect -An interface that models the wrapping of any object

Incorrect -A class that wraps any object

Incorrect -I don't know yet.

16. What is the prefered way of implementing different behaviors using abstraction?

Incorrect -Create a specialized object for each behavior, that will be created by querying a registry with a String that models each behavior.

Incorrect -Create a base class and a hierarchy of extensions, each implementing one behavior.

Your choice: correct -Create interfaces to model the behavior, then a set of classes to implement them and compose them in the business classes you need.

Incorrect -Create a single class with all the behaviors and use it everywhere.

Incorrect -I don't know yet.

17. You need to create an interface that adds methods to another interface. What do you do?

Your choice: correct -You extend the interface.

Incorrect -You override the interface.

Incorrect -You compose the interface.

Incorrect -You implement the interface.

Incorrect -I don't know yet.

18. You want to concatenate several instances of String together. What is the most efficient way to do it?

Incorrect -Use the concat() method of the String class.

Your choice: correct -Create a StringBuilder, add the strings in it, and convert it into the result String.

Incorrect -Use the + arithmetic symbol; that does the concatenation.

Incorrect -Put the instances of String in a list, then join that list using a StringJoiner.

Incorrect -I don't know yet.

1. What command must you use to execute a Main class that contains a main() method?

//Correct -$ java Main

Incorrect -$ java Main.class

Incorrect -$ Main

Incorrect -$ java Main.main()

Incorrect -I don't know yet.

2. What is a wrapper type in Java?

Incorrect -A class that wraps any object

Incorrect -An interface that models the wrapping of any object

Incorrect -An interface that models a primitive value

//Correct -A class that wraps a primitive value in an object

Incorrect -I don't know yet.

3. You have a class Holder<E>, where E is a type parameter. Where can E be used in this class?

Incorrect -As a returned type for a static factory method

//Correct -As a type for the parameters of the constructors

Incorrect -As a type of an exception thrown by an instance method

Incorrect -As a type for a static final field

Incorrect -I don't know yet.

4. You have the following class:

public abstract AbstractCollection {

   public abstract int size();

   public boolean isEmpty() {

      return size() == 0;

   }

}

You are about to write a Collection class that will extend this class. Among the following statements, which one is true for your Collection class?

Incorrect -You need to override the isEmpty() method.

Incorrect -You need to create an empty field of type boolean.

Incorrect -You need to create a size field of type int.

//Correct -You need to override the size() method.

Incorrect -I don't know yet.

5. You have the following interface:

public interface Comparator<E> {

   int compare(E e1, E e2);

   static <E, U> Comparator<E> comparing(

      Function<E, U> mapper){

      return ...//assume this is implemented;

   }

}

You need to create a Comparator<Person> using a Function<Person, String> called byName. How can you call the comparing() method?

//Correct -Comparator.<Person, String>comparing(byName);

Incorrect -Comparator.<Person>comparing(byName);

Incorrect -Comparator.<>comparing(byName);

Incorrect -Comparator.<String>comparing(byName);

Incorrect -I don't know yet.

6. You need to mark that a method is obsolete and should not be used anymore. What Javadoc tag are you going to use?

//Correct -@deprecated

Incorrect -@toBeRemoved

Incorrect -@obsolete

Incorrect -@replacedBy

Incorrect -I don't know yet.

7. What is the prefered way of implementing different behaviors using abstraction?

//Correct -Create interfaces to model the behavior, then a set of classes to implement them and compose them in the business classes you need.

Incorrect -Create a single class with all the behaviors and use it everywhere.

Incorrect -Create a base class and a hierarchy of extensions, each implementing one behavior.

Incorrect -Create a specialized object for each behavior, that will be created by querying a registry with a String that models each behavior.

Incorrect -I don't know yet.

8. You need to create an interface that adds methods to another interface. What do you do?

Incorrect -You compose the interface.

//Correct -You extend the interface.

Incorrect -You implement the interface.

Incorrect -You override the interface.

Incorrect -I don't know yet.

1. All the methods from the String class that return a type String (toUpperCase(), toLowerCase(), etc...) share a common property. What is this property?

Incorrect -They all return the current instance of String, pointing to a new internal array.

Incorrect -They all return a new instance of String pointing to the same internal array as the current String.

//Correct -They all return a new instance of String, pointing to a new internal array.

Incorrect -They all return the current instance of String, pointing to the same internal array.

Incorrect -I don't know yet.

2. Where can you use a wildcard (?) to denote a parameter type in your code?

Incorrect -An instance creation

Incorrect -A class declaration

Incorrect -A method invocation

//Correct -A method parameter declaration

Incorrect -I don't know yet.

3. Which of the following snippets would compile?

//Correct -long l = 1L + 2;

Incorrect -int i = 1L + 2L;

Incorrect -float = 3.14;

Incorrect -int i = 1L + 2;

Incorrect -I don't know yet.

4. Which statement declares an integer field that is shared by all the instances of the Holder class?

//Correct -public class Holder {

   private static int index;

}

Incorrect -public class Holder {

   final int index;

}

Incorrect -public class Holder {

   private int index;

}

Incorrect -public class Holder {

   private final int index;

}

Incorrect -I don't know yet.

5. What Java API is not supported in Android?

Incorrect -Java I/O

Incorrect -Java String

//Correct -Java Swing

Incorrect -Java Thread

Incorrect -I don't know yet.

6. What does JAR stand for?

Incorrect -Java Application Register

//Correct -Java Archive

Incorrect -Java Application Repository

Incorrect -Java Repository

Incorrect -I don't know yet.

7. You need to store key-value pairs in an instance of Map. You need to be able to iterate over these key-value pairs in the order in which they have been added to the Map. Which implementation can you choose?

Incorrect -SkipListMap

Incorrect -TreeMap

//Correct -LinkedHashMap

Incorrect -ArrayHashMap

Incorrect -I don't know yet.

8. Which tool is the most appropriate to create a web application in Java?

Incorrect -Java Development Kit

//Correct -Java Enterprise Edition

Incorrect -Java Runtime Environment

Incorrect -Java Standard Edition

Incorrect -I don't know yet.

9. What is a lambda expression in Java?

Incorrect -An implementation of any interface

Incorrect -An implementation of an interface that must be defined in the java.util.function package

//Correct -An implementation of any functional interface

Incorrect -An implementation of any function

Incorrect -I don't know yet.

10. You need to use an exception specific to your application instead of the standard IOException. Every time an IOException is thrown by some code in your application, you need to catch this exception and rethrow it, wrapped in your custom application exception. How can you preserve the IOException as the root cause of your custom exception?

Incorrect -Log the message and the stack trace using the standard JVM logging mechanism

//Correct -Pass the root IOException as a parameter to the constructor of your custom application exception

Incorrect -Preserve the message by passing it as a parameter to the constructor of your custom application exception, but not the stack trace

Incorrect -Set the message and the stack trace as a property of your custom application exception

Incorrect -I don't know yet.

11. You have the following method in an EmailService class:

public void sendEmail(User user) {

   sendEmail(user.getEmail());

}

What could happen here if your code is over-abstracted?

Incorrect -Subclassing EmailService could make the JVM unstable.

Incorrect -Providing two extensions of EmailService could create compile-time errors.

Incorrect -A race condition could occur on the reading of the email address.

//Correct -Anyone can subclass EmailService and change the logic of the sending of the email.

Incorrect -I don't know yet.

12. What structure of the Collection API can guarantee that the elements are iterated over in the same order in which they were added?

//Correct -The List

Incorrect -The OrderedSet

Your choice: incorrect -The SortedSet

Incorrect -The Set

Incorrect -I don't know yet.

13. Among the following, which one is NOT a property of Throwable?

Incorrect -The cause

Incorrect -The message

//Correct -The severity

Incorrect -The stack trace

Incorrect -I don't know yet.

14. What does this code segment print?

static void increment(int index) {

   index++;

}

public static void main(String[] args) {

   int i = 0;

   increment(i);

   i++;

   System.out.println(i);

}

Incorrect -2

Incorrect -0

[PluralSight -Java Testing.note](note://F48E8DB04B504158857B11C3F8ED48BD)

//Incorrect -3

Incorrect -I don't know yet.

15. How would you determine whether an instance of String is empty and it contains only blank characters (such as new lines or spaces)?

Incorrect -Call the isEmpty() method and test the result with the trim() method.

Incorrect -Compose the isEmpty() with the trim() call.

//Correct -Call the trim() method and test the result with the isEmpty() method.

Incorrect -Call the isEmpty() method.

Incorrect -I don't know yet.

16. Consider the following code:

try {

   String line = readFile("log.txt");

   int index = Integer.parseInt(line);

} catch (NumberFormatException e) {

   System.out.println("NumberFormatException");

} catch (IOException e) {

   System.out.println("IOException");

} finally {

   System.out.println("Finally");

}

The readFile() call may throw an IOException and the parseInt() call may throw a NumberFormatException, which is an unchecked exception. Suppose that the readFile() throws this IOException, and that the parseInt() also throws this NumberFormatException. What does this code print out?

//Correct -IOException

Finally

Incorrect -IOException

NumberFormatException

Finally

Incorrect -IOException

Incorrect -IOException

NumberFormatException

Incorrect -I don't know yet.

17. How can you impose that a class should not be instantiated?

Incorrect -You make the class an inner class.

Incorrect -You make the class static.

Incorrect -You make the class private.

//Correct -You make the class abstract.

Incorrect -I don't know yet.

18. What code do you need to add to a class Collection so that all the extensions of this class provide their own implementation of a method size()?

//Correct -abstract int size();

Incorrect -@Override int size();

Incorrect -super int size();

Incorrect -default int size();

Incorrect -I don't know yet.

1.Given the following enumeration:

public enum Civility {

   MRS, MR

}

What code can you write to get an instance of this enum?

* Incorrect -

Civility civility = new Civility("MRS");

* Incorrect -

Civility civility = Civility.of("MR")

* Your choice: correct -

Civility civility = Civility.MRS;

* Incorrect -

Civility civility = Civility.valueOf(1);

* Incorrect -

I don't know yet.

2.The List interface from the JDK has a default method called sort(Comparator), that sorts the content of this list. You have a custom implementation of the List interface in your application. What can you do with this sort method?

* Incorrect -

You cannot override it, since it is a default method.

* Incorrect -

You must override it and provide your own implementation.

* Incorrect -

You must override it, calling the default method from the interface.

* Your choice: correct -

You can override it if you think you can provide a better implementation than the one in the default method.

* Incorrect -

I don't know yet.

3.You must describe a behavior that some classes of your application should have, but you do not want to create any implementation. Given that these classes could already be inheriting from other classes, what should you do?

* Incorrect -

Create an abstract class that they will extend.

* Incorrect -

Create a concrete class that they will compose.

* Incorrect -

Create a factory class that they will compose.

* Your choice: correct -

Create an interface that they will implement.

* Incorrect -

I don't know yet.

4.How can a final field be set to a value?

* Incorrect -

By using an initializer only

* ~~Incorrect -~~

By using an initializer only

* Incorrect -

By setting it in a constructor only

* ~~Your choice: incorrect -~~

By setting it in a constructor only

* Your choice: incorrect -

By using an initializer and by optionally modifying this value in a constructor

* ~~Correct -~~

By using an initializer and by optionally modifying this value in a constructor

* Correct -

By using an initializer or by setting it in a constructor

* ~~Incorrect -~~

By using an initializer or by setting it in a constructor

* Incorrect -

I don't know yet.

I don't know yet.

5，。What is the difference between a Predicate<Integer> and an IntPredicate?

* Incorrect -

An IntPredicate can be implemented using a lambda expression, whereas a Predicate<Integer> cannot

* Incorrect -

An IntPredicate works on 16 bits ints, whereas Predicate<Integer> works on 32 bits Integer.

* ~~Your choice: correct -~~

An IntPredicate works on 16 bits ints, whereas Predicate<Integer> works on 32 bits Integer.

* Your choice: correct -

The IntPredicate works with ints, which is more efficient since there is no need to unbox the Integer, and a Predicate<Integer> does not.

* ~~Incorrect -~~

The IntPredicate works with ints, which is more efficient since there is no need to unbox the Integer, and a Predicate<Integer> does not.

* Incorrect -

A Predicate<Integer> can be implemented using a lambda expression, whereas an IntPredicate cannot.

* ~~Incorrect -~~

A Predicate<Integer> can be implemented using a lambda expression, whereas an IntPredicate cannot.

* Incorrect -

I don't know yet.

I don't know yet.

6.You have a stream of strings of characters. You want to concatenate all those strings using a comma as a separator. How do you proceed?

* Incorrect -

Collect the elements in a StringJoiner and call join() on that object, passing the comma as a parameter.

* Your choice: incorrect -

Collect the elements in a list and call join() on that list, passing the comma as a parameter.

* Incorrect -

Reduce the stream in a StringBuilder, using the StringBuilder::add as the reduction function.

* Correct -

Collect the elements using the Collectors.joining() collector, passing the comma as a parameter.

* Incorrect -

I don't know yet.

7.What is the role of the function passed as a parameter to the Collectors.groupingBy() factory method?

* Your choice: correct -

It computes a property from the elements of the stream used as a key for this object.

* Incorrect -

It creates the entry objects that will be used to create the map.

* Incorrect -

It specifies which collection will hold this object.

* Incorrect -

It creates the type of the map that will hold the objects of the stream.

* Incorrect -

I don't know yet.

8.What is E in the following declaration?

public interface Wrapper<E> {}

* Incorrect -

The object wrapped by this wrapper

* Incorrect -

The type of the class that implements this wrapper

* Your choice: incorrect -

The type of the object or primitive type wrapped by this wrapper

* Correct -

The type parameter for the wrapper interface

* Incorrect -

I don't know yet.

How can you impose that a class should not be instantiated

9. On how many bytes is an int encoded in Java?

* Incorrect -

8

* Incorrect -

2

* Your choice: incorrect -

The number depends on the CPU and the operating system your application is running on.

* Correct -

4

* Incorrect -

I don't know yet.

10。You have a User class with a single field: name. In your application, two instances of User are the same if they have the same name. Keeping in mind the optimized program performance and efficient object comparison, how can you enforce such a rule?

* Incorrect -

By only overriding an equals(Object) method in the User class

* Incorrect -

By creating an equals(User) and overriding hashCode() method in the User class

* Your choice: correct -

By overriding equals(Object) and hashCode() methods in the User class

* Incorrect -

By only creating an equals(User) method in the User class

* Incorrect -

I don't know yet.

11.You have a class with an empty constructor declared as private. How can you extend it?

* Incorrect -

Explicitly call it from the constructors of the extending classes.

* Incorrect -

Create a static method and call it explicitly from the constructors of the extending classes.

* Your choice: incorrect -

Extend the class normally because the constructors do not need to follow visibility modifiers.

* Correct -

Add a non-private constructor in it and explicitly call it from the constructors of the extending classes.

* Incorrect -

I don't know yet.

12.You need to concatenate a list of instances of String together. What precautions should you take?

* Your choice: incorrect -

Make sure all the elements of the list are properly consumed.

* Correct -

Avoid the unnecessary creation of instances of String.

* Incorrect -

Properly free the buffer of strings.

* Incorrect -

Properly flush the buffer of strings in the returned String.

* Incorrect -

I don't know yet.

13.What do you need to make an interface a functional interface?

* Your choice: incorrect -

Annotate it with @FunctionalInterface.

* Incorrect -

Remove its static methods.

* Correct -

Make sure it has only one abstract method that needs to be implemented.

* Incorrect -

Remove its default methods.

* Incorrect -

I don't know yet.

14.You have a list of doubles. Some of those doubles may be NaN or INFINITY and they should not be added. You must compute the sum of these doubles. What code will you write?

* Incorrect -

double sum = 0.0;

for (double d: list) {

if (!Double.isInfinite(d)) {

sum += d;

}

}

* Incorrect -

double sum = 0.0;

for (double d: list) {

sum += d;

}

* Correct -

double sum = 0.0;

for (double d: list) {

if (Double.isFinite(d)) {

sum += d;

}

}

* Your choice: incorrect -

double sum = 0.0;

for (double d: list) {

if (!Double.isNaN(d)) {

sum += d;

}

}

* Incorrect -

I don't know yet.

15.You are calling some code that is throwing an IOException. You need to catch this exception and rethrow it as an ApplicationException specific to your application. What is the correct code to do that?

* Your choice: incorrect -

catch (IOException e) {

throw new ApplicationException(e.getStackTrace());

}

* Incorrect -

catch (IOException e) {

throw new ApplicationException(e.getMessage());

}

* Incorrect -

catch (IOException e) {

throw new ApplicationException();

}

* Correct -

catch (IOException e) {

throw new ApplicationException(e);

}

* Incorrect -

I don't know yet.

16.In which case do you use the == operator to compare instances of String?

* Incorrect -

When using the equals() method

* Correct -

When both strings have been created as constants known at compile time

* Incorrect -

Any time because the String class has an internal cache and checks for duplicates

* Your choice: incorrect -

When both strings have been read from the same file

* Incorrect -

I don't know yet.

17.You need to distribute your application as a JAR file on the Internet. How can you make your customer sure that the JAR they will download has not been modified by anyone?

* Incorrect -

You can hash each class file in it.

* Correct -

You can sign it so that the JVM can check for its integrity.

* Incorrect -

You can distribute a password along with it.

* Your choice: incorrect -

You can distribute a hash code along with it.

* Incorrect -

I don't know yet.

18.You need to deploy a Java Web Application on a server. What elements do you need for this application to work?

* Your choice: correct -

A Java Runtime Environment and a Java EE application server

* Incorrect -

A Java Runtime Environment and the Java EE APIs

* Incorrect -

A Java Virtual Machine and a Java EE application server

* Incorrect -

A Java Development Kit and the Java EE APIs

* Incorrect -

19.I don't know yet.What is the root class of the exception class' hierarchy?

* Incorrect -

Exception

* Correct -

Throwable

* Incorrect -

Error

* Your choice: incorrect -

RuntimeException

* Incorrect -

I don't know yet.