Select the correct order of restrictiveness for access modifiers... (First one should be least restrictive)

public < protected < package (i.e. no modifier) < private

Explanation : That's right, protected is less restrictive than package.

public < package (i.e. no modifier) < protected < private

public < protected < private < package (i.e. no modifier)

protected < package (i.e. no modifier) < private < public

depends on the implementation of the class or method.

2 Can we have two public classes in one java file?

True

False

Explanation : No, a java file can contain only one public class.

3 What will the following code print when compiled and run? (Assume that MySpecialException is an unchecked exception.)

```
1. public class ExceptionTest {
2.    public static void main(String[] args) {
3.       try {
4.          doSomething();
5.       } catch (MySpecialException e) {
6.          System.out.println(e);
7.       }
8.    }
9.
```

```
10.  static void doSomething() {
11.      int[] array = new int[4];
12.      array[4] = 4;
13.      doSomethingElse();
14.  }
15.
16.  static void doSomethingElse() {
17.      throw new MySpecialException("Sorry, can't do something else");
18.  } }
```

It will not compile.

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 4
    at ExceptionTest.doSomething(ExceptionTest.java:12)
    at ExceptionTest.main(ExceptionTest.java:4)
```

```
Exception in thread "main" MySpecialException: 4
    at ExceptionTest.doSomethingElse(ExceptionTest.java:17)
    at ExceptionTest.doSomething(ExceptionTest.java:12)
    at ExceptionTest.main(ExceptionTest.java:4)
```

```
Exception in thread "main" MySpecialException: Sorry, can't do something else
    at ExceptionTest.doSomethingElse(ExceptionTest.java:17)
    at ExceptionTest.doSomething(ExceptionTest.java:12)
    at ExceptionTest.main(ExceptionTest.java:4)
```

```
Exception in thread "main" MySpecialException: Sorry, can't do something else
    at ExceptionTest.doSomethingElse(ExceptionTest.java:17)
    at ExceptionTest.doSomething(ExceptionTest.java:13)
    at ExceptionTest.main(ExceptionTest.java:4)
```

4 How do you get the size of an array?

myArray.size

myArray.length

myArray.size()

myArray.length()

5 Which of the following are benefits of ArrayList over an array?

You do not have to worry about the size of the ArrayList while appending elements.

You do not have to worry about the size of the ArrayList while appending elements.

Explanation : An ArrayList resized dynamically at run time as per the situation. An array cannot be resized once created. This reduces the amount of boiler plate code that is required to do the same task using an array.

It consumes less memory space.

You do not have to worry about thread safety.

It allows you to write type safe code.

6 Which of the following is not a Wrapper Class

Boolean

Long

Double

Short

None of the above

7 Which of the following Map methods is used to add a key/value pair? (Choose the best option)

put

get

keyset

add

8 Methods defined in an interface are implicitly

public

static

public abstract

static void

9 Abstract classes:

cannot be extended.

contain only abstract methods.

can extend other abstract classes multiple times.

are made to be extended not instantiated.

10 Which of the following classes have a default constructor?

```
class A{ }
class B {
 B(){ }
}
class C{
 C(String s){ }
}
```

A

Explanation : There is only one rule regarding the "default" constructor: The Java compiler automatically adds a constructor that takes no argument and has the same access as the class, if and only if the programmer does not define ANY constructor in the class. In this case, the programmer has not defined any constructor for class A, hence it will have the default constructor. For class B, the programmer has defined a constructor that is exactly same as the default constructor that would have been provided automatically. It is a matter of interpretation whether it can be called a default constructor or not. Based on Java Language Specification section 8.8.9, quoted below, our interpretation is that class B will not get a default constructor: (http://docs.oracle.com/javase/specs/jls/se7/html/jls-8.html) 8.8.9 Default Constructor If a class contains no constructor declarations, then a default constructor with no formal parameters and no throws clause is implicitly declared. If the class being

declared is the primordial class Object, then the default constructor has an empty body. Otherwise, the default constructor simply invokes the superclass constructor with no arguments. It is a compile-time error if a default constructor is implicitly declared but the superclass does not have an accessible constructor (6.6) that takes no arguments and has no throws clause. It follows that if the nullary constructor of the superclass has a throws clause, then a compile-time error will occur.

A and B

B

C

B and C

11 Object class:

Does not have any methods.

It is an abstract class.

It has one method, finalize().

It is directly or indirectly the superclass of all Java classes.

12 What is the standard signature for the main method?

public void main(String args[])

public void static main(String args[])

public static void main()

public static void main(String args[])

13 The keyword "static" means:

something belongs to a specific instance.

something is a constant.

something is locally available to a single instance of the class.

something belongs to the class and is globally available to all instances.

14 Which method must be implemented by all threads?

wait()

start()

stop()

run()

Explanation : All threads must implement the run() method.

15 RuntimeException and its subclasses are checked exceptions.

False

True

16 Which of the following is not a primitive data value in Java? (Choose all that apply.)

"x"

Explanation : Java has only the following primitive data types: boolean, byte, short, char, int, long, float and double.

'x'

10.2F

Object

Explanation : Java has only the following primitive data types: boolean, byte, short, char, int, long, float and double.

false

17 The == operator

Checks for equality of two primitive variables

Checks for equality of two objects based on the attributes

Checks for data type equality only

None of the above.

18 Generics provide compile time safety.

True

False

19 Doubles and floats are different in that

Any number with a decimal point is considered to be a float-literal, with double literals needing to be appended with a 'd'.

They are the same.

Doubles have 64 bits of precision, while floats have 32 bits of precision.

Doubles have 32 bits of precision, while floats have 64 bits of precision.

20 Garbage collection can be forced by

Using the gc() method

Calling the finalize() method

It can never be forced.

None of the options

21 A _____ is a collection that is based on the last-in-first-out (LIFO) policy.

Stack

Queue

Heap

List

22 How does a Collection differ from a Map?

A Collection has an iterator whereas a Map does not.

Collections have a type whereas Maps have key-value pairs.

Collection inherits from iterable, whereas Map does not.

All of the Above.

23 What collection will NOT allow duplicate elements?

Map

List

Tree

Set

24 What is the string pool?

A location in memory where all strings are stored.

A location in memory where strings referenced as literals in an application are stored.

Java does not feature a string pool.

A location for strings in the stack.

25 True or False: Inherited behaviors can be overriden in a subclass

TRUE

FALSE

26 True or False: If the class Car extends class Automobile, then an instance of Automobile can access Car methods

TRUE

FALSE

27 True or False: A class in Java can extend multiple parent classes

TRUE

Explanation : No, Java does not support multiple inheritance

FALSE

Explanation : Correct

28 Assuming a method contains code which may raise an Exception (but not a RuntimeException), what is the correct way for the method to indicate that it expects the caller to handle that exception?

throw Exception

new Exception

throws Exception

Don't need to specify anything

29 True or False: Data given no access modifier can only be accessed within the same class in which it is declared.

TRUE

FALSE

30 Which is not an access modifier in Java?

public

protected

default

private

31 What is the use of the '%' operator?

It returns the remainder of a division operation between integers

It converts the argument into a percentage

It divides two integers, but not decimals

32 A boolean variable can only hold the values true or false.

TRUE

FALSE

33 What is the root interface of the Collections API?

Collections

List

Collection

Throwable

34 Each thread gets its own stack and heap.

False

True

Explanation : They only get their own Stack.

35 The continue statement pulls the flow of control out of a looping block.

False

True

36 Which of the following are Short-Circuit?

|

||

&

&&

37 The keyword "final" means:

the method can be overloaded.

the class cannot be extended.

the class is abstract and cannot be instantiated.

the class cannot be implemented.

38 Which one of the following is a valid class declaration?

class example implements ArrayList

static abstract class example implements Serializable extends ArrayList

class example implements Serializable, Runnable

class Example extends ArrayList, HashMap

39 Given the command line argument in quotes "1 2 3" what will be the output of the following code?

```
public static void main(String[] args)
{
      for(String temp: args){
            System.out.print(temp);
      }
}
```

Nothing prints

123

temp

1 2 3

40 What is an exception?

An unrecoverable condition.

An event that must always be thrown.

An event that freezes the normal program flow.

An event that disrupts the normal flow of a program's instructions.

41 An interface can _____ any number of _____.

implement, interfaces

inherit, classes

support, super interfaces

extend, interfaces

42 "Collections" is?

The interface that all java collections implement

A class filled with static methods used to manipulate collections

43 Which interface does not extend the Collection interface?

List

Set

Queue

Map

44 Can you catch an Error?

Yes, but you should not - an application can rarely handle them properly, or continue running after they occur

Yes, and you should - Errors can be dangerous if not handled

No

45 A class can be abstract, even if all of its methods are concrete.

TRUE

FALSE

46 The equals() method is equivalent to the == operator, unless overridden

TRUE

FALSE

47 True or False: An array is a type of Collection in Java

TRUE

FALSE

48 How many String objects have been created at the end of this code sequence?

```
String a = "Hello";
String b = "World";
a = a + b;
a = new String("Hello");
String c = "World";
```

One

Two

Three

Four

49 Java supports multiple inheritance of classes

TRUE

FALSE

50 What is the value stored at the index [1][2]?

```
int[][] twoDimensionalArray = { {0, 1, 2}, {2, 4, 6}, {2, 1, 0} };
```

6

2

4

1

51 What is the purpose of the Java compiler?

To transform Java code into instructions usable by the JVM

To transform Java code into instructions usable by the operating system

To transform Java code into instructions usable by a web browser

52 Which of the following is true of the StringBuilder class?

It is safe to use in multi-threaded environments.

Explanation : StringBuilder is not thread-safe, so it is not safe to use in multi-threaded environments.

It extends the String class.

Explanation : The String class is a final class, which means that you cannot extend it.

The class allows you to use the appendString() method to mutate a string.

Explanation : There is no such method on the StringBuilder class.

None of the above.

53 What should be the output of thsi code snippet?

```
1  package javaqas.java8;
2
3  public class Calculator {
4
5      interface IntegerMath {
6          int operation(int a, int b);
7      }
8
9      public int operateBinary(int a, int b, IntegerMath op) {
10         return op.operation(a, b);
11     }
12
13     public static void main(String... args) {
14
15         Calculator myApp = new Calculator();
16         IntegerMath addition = (a, b) -> a + b;
17         IntegerMath subtraction = (a, b) -> a - b;
18         System.out.println("40 + 2 = " + myApp.operateBinary(40, 2, addition));
19         System.out.println("20 - 10 = " + myApp.operateBinary(20, 10, subtraction));
20     }
21 }
22
```

compilation error

Runtime Exception

Compile and execute without any Exception or error

None

54 Which of the following are true regarding overloading of a method?

An overloading method must have a different parameter list and same return type as that of the overloaded method.

If there is another method with the same name but with a different number of arguments in a class then that method can be called as overloaded.

If there is another method with the same name and same number and type of arguments but with a different return type in a class then that method can be called as overloaded.

An overloaded method means a method with the same name and same number and type of arguments exists in the super class and sub class.

55 What is the difference between method overloading and overriding?

Method overloading is the same method name but different parameters. Method overriding is the same name but different implementation.

Method overloading is the same method name but return type. Method overriding is the same name but different implementation.

Method overloading is the same method name but different parameters. Method overriding is the same name but different parameters.

Method overloading is the same method name but different parameters. Method overriding is a different name but same parameters.