**Question 1- C**

A class may be declared with the modifier public, in which case that class is visible to all classes everywhere. If a class has no modifier (the default, also known as package-private), it is visible only within its own package. At the member level, you can also use the public modifier or no modifier (package-private) just as with top-level classes, and with the same meaning. For members, there are two additional access modifiers: private and protected. The private modifier specifies that the member can only be accessed in its own class. The protected modifier specifies that the member can only be accessed within its own package (as with package-private) and, in addition, by a subclass of its class in another package.

https://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html

**Question 2 – B**

Using this with a Constructor. From within a constructor, you can also use the this keyword to call another constructor in the same class. Doing so is called an explicit constructor invocation. Here's another Rectangle class, with a different implementation from the one in the Objects section.

https://docs.oracle.com/javase/tutorial/java/javaOO/thiskey.html

**Question 3 – D**

Code isn't compiling. Exception in thread "main" java.lang.Error: Unresolved compilation problem.

**Question 4 – D**

Cıde isn't compiling. Exception in thread "main" java.lang.Error: Unresolved compilation problem. The method print(boolean) in the type PrintStream is not applicable for the arguments (void)

**Question 5 - B**

The method parameter values are copied to another variable and then the copied object is passed, that's why it's called pass by value.

**Question 6 – C**

public void setRange(int range) is valid method signature.

**Question 7 – B**

Super() isn't both used in the same constructor with this.

**Question 8 – B**

Long is blanked the fill to make the class compiling.

**Question 9 – C**

Static is always available to all instances of the class.

**Question 10 – A**

Inserted at line //p1 this(4), System.out.print(new Jump().rope) give us output 5.

**Question 11 – ?**

**Question 12** – **A**

public String getStuff is to ensure the class data is properly encapsulated.

**Question 13 – ?**

**Question 14 – C**

public void sing1(String key, String... harmonies) is compiling with our code.

**Question 15 – C**

private static void slalom(Ski racer, int[] speed, String name) {

racer.age = 18

mySiker > Age:18

myName > "Rosei" int

mySpeed > 0 - null

**Question 16 -  B**

public class Calculations {

public Integer findAverage(int sum) { return sum; }

public Long findAverage(int sum, int divisor) is successfully added to an overloaded version of the this method.

**Question 17 – D**

Encapsulation isn't purpose to improve Java performance.

https://www.javatpoint.com/encapsulation

**Question 18 – A**

Int[] is can be modified after they are passed to a method as an argument.

**Question 19 – ?**

**Question 20 – A**

Byte is return statement from being used within the method.

**Question 21 – C**

public final class Games {

public final static int finish(final int score) {

final int win = 3;

final int result = score++ < 5 ? 2 : win;

return result+=win;

}

public static void main(final String[] v) {

System.out.print(finish(Integer.parseInt(v[0])));

Two final will be deleting to compile.

**Question 22 – D**

The super keyword in Java is a reference variable which is used to refer immediate parent class object. Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

https://www.javatpoint.com/super-keyword

**Question 23 – D**

Void run(String goverment) is can be calling all classes.

**Question 24 – A**

The access modifier of strength is protected, meaning subclasses and classes within the same package can modify it. Changing the value to private would improve encapsulation by making the Protect class the only one capable of directly modifying it. For these reasons, the first statement is correct.

**Question 25 – A**

Go_$Outside$2()      // valid

have-Fun()              // non-valid

new()                    // non-valid

9enjoyTheweather     // non-valid

private static void Go_$Outside$2()

**Question 26  - D**

getNumberOfChickens is non valid declaration.

**Question 27 – A**

A change made to primitive value passed to a method is reflected in the calling method is ture statement is about passing data to method.

**Question 28 – C**

Code isn't is compiling. Exception in thread "main" java.lang.Error: Unresolved compilation problems. The blank final field contents may not have been initialized

**Question 29 - ?**

**Question 30 – C**

import static clothes.Store.getClothes is line adding top on the code.

**Question 31 – D**

package-private (often just called package) means that other members of the same package have access to the item. package-private is the default access modifier and does not have a keyword, because package is used to specify the package for a class or interface. To declare package access for something, use no access modifier.

**Question 32 – B**

Code is just one contain compilation error.

**Question 33 – A**

An instance method is allowed to reference a static variable is true statement.

**Question 34 – B**

public short calculateDistance(double lat1, double lon1,

double lat2, double lon2) {

return new Byte((byte)6); // must be adding to compiling code.

**Question 35 – C**

Method Overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different. It is similar to constructor overloading in Java, that allows a class to have more than one constructor having different argument lists.

**Question 36 – B**

private static final String monday and final protected int Thursday are removing class declaration to succesfully compiling.

**Question 37 – D**

Code isn't compiling beucase q3 is un-defined (the constructor Puppy(int) is undefined).

**Question 38 – A**

Public access modifier is allow access to everthing.

**Question 39 – A**

System.out.print(phone.size) result output is 3.

**Question 40 – ?**

**Question 41 – C**

Call datas are firstly int, secondly string then last string.

**Question 42 – A**

The value of a static variable must be set when the variable is declared or in a static initialization block is true statement.

**Question 43 – B**

A class does not have to have a constructor explicitly defined is true statement.

**Question 44 – A**

Code isn't compiling whatever happens remove anything.

**Question 45 – D**

Code isn't compiling because Forest  class must be declarated.

**Question 46 – A**

System.out.print(new ChooseWisely().choose((byte)2+1)) result output is 5.

**Question 47 – C**

Code isn't compiling. The method getScore(Long) in the type Football is not applicable for the arguments (int).

**Question 48 – A**

$sprint() is valid method name in Java.

**Question 49 - ?**

**Question 50 – D**

private void deposit(int depositSlip) is must be adding in code for compiling.