**1.- Which declaration initializes a boolean variable?**

* a) boolean m = null
* b) Boolean j = (1<5)
* c) boolean k = 0
* d) boolean h = 1

**2.- What is the DTO pattern used for?**

* a) To Exchange data between processes RESPUESTA CORRECTA
* b) To implement the data Access layer
* c) To implement the presentation layer

**3.- Int a = 10; int b = 37; int z = 0; int w = 0;**

If (a==b) {z=3; } else if (a>b) {z=6;}

w = 10 \* z;

What is the result?

* a) 30
* b) 0
* c) 60

class Class1{String v1;}

class Class2{

Class1 c1;

String v2;

}

class Class3 {Class2 c1; String v3;}

**4.- Which three options correctly describe the relationship between the classes?**

* Class2 has-a v3.
* Class1 has-a v2.
* Class2 has-a v2.
* Class3 has a v1.
* Class2 has-a Class3.
* Class2 has-a Class1.

**Pregunta 5**

try {

    // assume "conn" is a valid Connection object

    // assume a valid Statement object is created

    // assume rollback invocations will be valid

    // use SQL to add 10 to a checking account

    Savepoint s1 = conn.setSavePoint();

    // use SQL to add 100 to the same checking account

    Savepoint s2 = conn.setSavePoint();

    // use SQL to add 1000 to the same checking account

    // insert valid rollback method invocation here

} catch (Exception e) {}

**What is the result?**

1. If conn.rollback(s1) is inserted, account will be incremented by 10.
2. If conn.rollback(s1) is inserted, account will be incremented by 1010.
3. If conn.rollback(s2) is inserted, account will be incremented by 100.
4. If conn.rollback(s2) is inserted, account will be incremented by 110.
5. If conn.rollback(s2) is inserted, account will be incremented by 1110.

**Pregunta 6**

**Which two statements are true an Abstract ?**

A) An abstract class can implement an interface.

B) An abstract class can be extended by an interface.

C) An interface CANNOT be extended by another interface.

D) An interface can be extended by an abstract class.

E) An abstract class can be extended by a concrete class.

F) An abstract class CANNOT be extended by an abstract class.

**Pregunta 7**

public class Main {

    public static void main(String[] args) throws Exception {

        doSomething();

    }

    private static void doSomething() throws Exception {

        System.out.println("Before if clause");

        if (Math.random() > 0.5) {

            throw new Exception();

        }

        System.out.println("After if clause");

    }

}

**Which two are possible outputs?**

1. Before if clause Exception in thread “main” java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15).
2. Before if clause Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15) After if clause
3. Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15)
4. Before if clause After if clause

respuesta: A y D

**Pregunta 8**

public class MyFive {

    public static void main(String[] args) {

        //short kk = ?;

        short ii;

        short jj = 0;

        for (ii = kk; ii > 6; ii-=1) {

            jj++;

        }

        System.out.println("jj = " + jj);

    }

}

**What value should replace kk in line 18 to cause jj = 5 to be output?**

1. -1
2. 1
3. 5
4. 8
5. 11

**Pregunta 9**

What will make this code compile and run?

1. Change line 3 to the following: public int price;
2. Change line 7 to the following: int price = new Simple();
3. Change line 7 to the following: float price = new Simple();
4. Change line 7 to the following: price = 4f;
5. Change line 7 to the following: price.price = 4;

Explicación:

Si quisiéramos cambiar la referencia de el price de la clase, tendríamos que ingresar al objeto para hacerlo, de otra forma estamos intentando cambiar la referencia de Simple price, y 4 (int) no es de la clase Simple.

**Pregunta 10**

In the Java collections framework a Set Is:

1. A collection that cannot contain duplicate elements.
2. An ordered collection that can contain duplicate elements.
3. An object that maps value key sets and cannot contain values Duplicates.

Explicación:

Un Set pertenece a collection y no permite el tener elementos duplicados en su interior.

Los otros dos puntos hacen referencia a List y Map, respectivamente.

Pregunta 11

public class SuperTest {

public static void main(String[] args) {

// statement1

// statement2

// statement3

}

}

class Shape {

public Shape() {

System.out.println("Shape: constructor");

}

public void foo() {

System.out.println("Shape: foo");

}

}

class Square extends Shape {

public Square() {

super();

}

public Square(String label) {

System.out.println("Square: constructor");

}

public void foo() {

super.foo();

}

public void foo(String label) {

System.out.println("Square: foo");

}

}

**11.- What should statement1, statement2, and statement3, be respectively, in order to produce the result?**

Shape: constructor

Shape: foo

Square: foo

1. Square square = new Square("bar"); square.foo("bar"); square.foo();
2. Square square = new Square("bar"); square.foo("bar"); square.foo("bar");
3. Square square = new Square(); square.foo(); square.foo(bar);
4. Square square = new Square(); square.foo(); square.foo("bar");
5. Square square = new Square(); square.foo(); square.foo();

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**12.- What is the result?** A. sc: class.Object asc: class.AnotherSampleClass

 B.  sc: class.SampleClass asc: class.AnotherSampleClass

 C. sc: class.AnotherSampleClass asc: class.SampleClass

 D. sc: class.AnotherSampleClass asc: class.AnotherSampleClass

Explicación:

getClass evalúa el la clase de el objeto, y no de la variable de referencia, por lo que al pasarle a sc el objeto AnotherSampleClass() , nos da este resultado.

**13. What is true about the class Wow?**Imagen que contiene Texto

Descripción generada automáticamente

A.      It compiles without error.

B.      It does not compile because an abstract class cannot have private methods.

C.      It does not compile because an abstract class cannot have instance variables.

D.      It does not compile because an abstract class must have at least one abstract method.

E.       It does not compile because an abstract class must have a constructor with no arguments.

**Explicación:**

Respuesta:  *A. It compiles without error.*

Una clase abtracta puede tener constructores, variables de instancia, métodos privados y no abstractos.

**14. The SINGLETON pattern allows:**

A.      Have a single instance of a class and this instance cannot be used by other classes.

B.      Having a single instance of a class, while allowing all classes have access to that instance.

C.      Having a single instance of a class that can only be accessed by the first methods that calls it.

**Explicación:**

Respuesta: *B. Having a single instance of a class, while allowing all classes have access to that instance.*

**15. How many times is 2 printed?Texto

Descripción generada automáticamente**

A.      Zero.

B.      Once.

C.      Twice.

D.      Thrice.

E.       It is not printed because compilation fails.

**Explicación:**

Respuesta:  *B. Once.*

Dentro del foreach tenemos un ciclo que implementa un *break* sin condición por lo que el ciclo while no continuará después de imprimir e incrementar el valor de “ii”, el valor de “ii” se incrementará hasta 2, debido a la expresión booleana dada en el ciclo. Así que sólo una vez tendrá el valor de 2 y solo se podrá imprimir una vez.

**16.What is the result?Texto

Descripción generada automáticamente**

A.      3false1

B.      2true3

C.      2false3

D.      3true1

E.       3false3

F.       2true1

G.      2false1

**Explicación:**

Respuesta: *D. 3true1*

En la primera impresión se obtiene el tamaño del primer arreglo en array2D que es de **3**, la segunda impresión obtiene la clase a la que pertenece el segundo arreglo almacenado en array2D, y verifica si es de tipo Array, por lo que se obtiene **true,** y la última impresión imprime el entero almacenado en el primer arreglo que se encuentra en el índice 1, y corresponde a **1.**

**17.- In Java the difference between throws and throw is:**

a. Throws throws an exception and throw indicates the type of exception that the method.

b. Throws is used in methods and throw in constructores.

c. Throws indicates the type of exception that the method does not handle and throw an exception.

**18.- What is the result?**

Texto

Descripción generada automáticamente

1. 4321.
2. 0000.
3. An exception is thrown at runtime.
4. Compilation fails because of an error in line 18.

**19.- Which is true?**

Texto

Descripción generada automáticamente

1. If line 10 is removed, the compilation succeeds.
2. If line 11 is removed, the compilation succeeds.
3. If line 12 is removed, the compilation succeeds.
4. If line 13 is removed, the compilation succeeds.
5. More than one line must be removed for compilation to succeed.

20.- What is the result?

class Atom {

Atom() {System.out.print("atom ");}

}

class Rock extends Atom {

Rock(String type) {System.out.print(type);}

}

public class Mountain extends Rock {

Mountain(){

super("granite ");

new Rock("granite ");

}

public static void main(String[] a) {new Mountain();}

}

1. Compilation fails.
2. Atom granite.
3. Granite granite.
4. Atom granite granite.
5. An exception is thrown at runtime.
6. Atom granite atom granite.

correcta: **F.** \*atom granite atom granite.

respuesta: F atom granite atom granite

**Pregunta 21**

Which statement is true?

A. 420 is the output.

B. An exception is thrown at runtime.

C. All constructors must be declared public.

D. Constructors CANNOT use the private modifier.

E. Constructors CANNOT use the protected modifier.

class ClassA {

public int numberOfInstances;

protected ClassA(int numberOfInstances) {

     this.numberOfInstances = numberOfInstances;

}

}

public class ExtendedA extends ClassA {

private ExtendedA(int numberOfInstances) {

     super(numberOfInstances);

}

public static void main(String[] args) {

     ExtendedA ext = new ExtendedA(420);

     System.out.print(ext.numberOfInstances);

}

}

**Pregunta 22**

**What is the result?**

A. 4 Null.

B. Null 4.

C. An IllegalArgumentException is thrown at run time.

D. 4 An ArrayIndexOutOfBoundsException is thrown at run time.

public class Test {

public static void main(String[] args) {

     int[][] array = { {0}, {0,1}, {0,2,4}, {0,3,6,9}, {0,4,8,12,16} };

     System.out.println(array[4][1]);

     System.out.println(array[1][4]);

}

}

**Pregunta 23**

What is the result?

A. There is no output.

B. d is output.

C. A StringIndexOutOfBoundsException is thrown at runtime.

D. An ArrayIndexOutOfBoundsException is thrown at runtime.

E. A NullPointException is thrown at runtime.

F. A StringArrayIndexOutOfBoundsException is thrown at runtime.

public class X {

public static void main(String[] args) {

     String theString = "Hello World";

     System.out.println(theString.charAt(11));

}

}

**Pregunta 24**

What is the result?

A. The program prints 1 then 2 after 5 seconds.

B. The program prints: 1 thrown to main.

C. The program prints: 1 2 thrown to main.

D. The program prints: 1 then t1 waits for its notification.

public class Bees {

public static void main(String[] args) {

     try {

         new Bees().go();

     }catch (Exception e) {

         System.out.println("thrown to main");

     }

}

synchronized void go() throws InterruptedException {

     Thread t1 = new Thread();

     t1.start();

     System.out.print("1 ");

     t1.wait(5000);

     System.out.print("2 ");

}

}

**Pregunta 25**

**¿Cuál será el resultado?**

public class SampleClass {

public static void main(String[] args) {

SampleClass sc, scA, scB;

sc = new SampleClass();

scA = new SampleClassA();

scB = new SampleClassB();

System.out.println(“Hash is: “ + sc.getHash() +

“, “ + scA.getHash() + “, “ + scB-getHash());

}

public int getHash() {

return 111111;

}

}

class SampleClassA extends SampleClass {

public int getHash() {

return 44444444;

}

}

class SampleClassB extends SampleClass {

public int getHash() {

return 999999999;

}

}

1. Compilation fails
2. An exception is thrown at runtime
3. There is no result because this is not correct way to determine the hash code
4. Hash is: 111111, 44444444, 999999999.

**Pregunta 26**

**¿Cuál sería el resultado?**

public class Test {

public static void main(String[] args) {

int b = 4;

b–;

System.out.println(--b);

System.out.println(b);

}

}

1. 22
2. 12
3. 32
4. 33

**Pregunta 27. ¿Cuál sería el resultado?**

import java.util.\*;

public class App {

public static void main(String[] args) {

List p = new ArrayList();

p.add(7);

p.add(1);

p.add(5);

p.add(1);

p.remove(1);

System.out.println(p);

}

}

1. [7, 1, 5, 1]
2. [7, 5, 1]
3. [7, 5]
4. [7, 1]

**Pregunta 28. ¿Cuál sería el resultado?**

public classDoCompare4 {

public static void main(String[] args) {

String[] table = {“aa”, “bb”, “cc”};

int ii = 0;

do {

while (ii < table.length) {

System.out.println(ii++);

}

} while (ii < table.length);

}

}

1. 0
2. 0 1 2
3. 0 1 2 0 1 2 0 1 2
4. Compilation fails

**Pregunta 29. ¿Cuál sería el resultado?**  
public class DoCompare1 {

    public static void main(String[] args) {

        String[] table = {"aa", "bb", "cc"};

        for (String ss : table) {

            int ii = 0;9

            while (ii < table.length) {

                System.out.println(ss + ", " + ii);

                ii++;

            }

        }

    }

}

1. Zero.
2. Once.
3. Twice
4. Thrice
5. Compilation fails

El resultado final de las impresiones será:aa, 2 -bb, 2 - cc, 2 = 3 veces sale el 2.

aa, 0 aa, 1 aa, 2

bb, 0 bb, 1 bb, 2

cc, 0 cc, 1 cc, 2

**30. What is the result?**

Texto

Descripción generada automáticamente

A. The value “4” is printed at the command line.

B. Compilation fails because of an error in line 5.

C. Compilation fails because of an error in line 9.

D. A NullPointerException occurs at runtime.

E. A NumberFormatException occurs at runtime.

F. An IllegalStateException occurs at runtime.

**Solución**

Para evitar el NullPointerException, debemos asegurarnos de que i esté inicializada antes de usarla en la operación. Aquí hay una versión corregida del código:

Ahora, cuando ejecutemos el código, y estará inicializada a 0, por lo que la operación x = i + y será x = 0 + 4, y se imprimirá 4.

**31. What is the result?**

A. Class Base1 { abstract class Abs1 { } }

* Esto es legal. Es posible tener una clase abstracta dentro de otra clase.

B. Abstract class Abs2 { void doit() { } }

* Esto es legal. Una clase abstracta puede tener métodos concretos.

C. class Base2 { abstract class Abs3 extends Base2 { } }

* Esto es legal. Es posible tener una clase abstracta que extiende otra clase dentro de una clase.

**D. class Base3 { abstract int var1 = 89; }**

* Esto es ilegal. Las variables no pueden ser abstractas. La palabra clave abstract solo se aplica a métodos y clases.

Por lo tanto, el fragmento de código ilegal es el **D**.

**Pregunta 32. ¿Cuál sería el resultado?**

public class ScopeTest {

    int z;

    public static void main(String[] args) {

        ScopeTest myScope = new ScopeTest();

        int z = 6;

        System.out.print(z);

        myScope.doStuff();

        System.out.print(z);

        System.out.print(myScope.z);

    }

    void doStuff() {

        int z = 5;

        doStuff2();

        System.out.print(z);

    }

    void doStuff2() {

        z = 4;

    }

}

1. **6564**
2. 6554
3. 6566
4. 6565

Para confirmar nuestra comprensión:

1. **System.out.print(z);** en main imprime 6.
2. **myScope.doStuff();** llama a doStuff:
   * **doStuff** declara int z = 5;.
   * **doStuff2** establece myScope.z = 4;.
   * **System.out.print(z);** en doStuff imprime 5.
3. **System.out.print(z);** en main después de doStuff imprime 6.
4. **System.out.print(myScope.z);** imprime 4.

Entonces, el orden de las impresiones y sus valores será 6564.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

33. creo que G

**respuesta correcta: A. b 3 porque para variables de referencia tenemos que voltear a ver a la variable de referencia.**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

34 E o D.

la primera imprime la referencia en memoria y la segunda imprime walls

Logotipo

Descripción generada automáticamente con confianza baja

35. C

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

37. A, C, E, F, G

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

39. A

Interfaz de usuario gráfica

Descripción generada automáticamente

40. B y D

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

41. C

Texto

Descripción generada automáticamente

42. D creo CORRECTO

los métodos de setCapacity y getInstance no existen para la clase StringBuilder

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

43. creo que B

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

44. C

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

45. C

Texto

Descripción generada automáticamente con confianza media

46. A, E, FInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

47. C

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

48. C 7532

**49-What is the result?**

public class Calculator {

    int num = 100;

    public void calc(int num) {

        this.num = num \* 10;

    }

    public void printNum() {

        System.out.println(num);

    }

    public static void main(String[] args) {

        Calculator obj = new Calculator();

        obj.calc(2);

        obj.printNum();  // Esto imprimirá 20

    }

}

a-20

b-100

c-1000

d-2

**50-¿Qué tres modificaciones, hechas de manera independiente, permitirán que la clase Greet compile y se ejecute?**

package handy.dandy;

public class KeyStroke {

    public void typeExclamation() {

        System.out.println("!");

    }

}

package handy;

public class Greet {

    public static void main(String[] args) {

        String greeting = "Hello";

        System.out.print(greeting);

        KeyStroke stroke = new KeyStroke();

        stroke.typeExclamation();

    }

}

A. Line 8 replaced with handy.dandy.KeyStroke stroke = new KeyStroke();

B. Line 8 replaced with handy.\*.KeyStroke stroke = new KeyStroke();

C. Line 8 replaced with handy.dandy.KeyStroke stroke = new handy.dandy.KeyStroke();

D. import handy.\*; added before line 1.

E. import handy.dandy.\*; added after line 1.

F. import handy.dandy.KeyStroke; added after line 1.

G. import handy.dandy.KeyStroke.typeExclamation(); added after line 1.

49. A. 20

50. A, E y F

Interfaz de usuario gráfica

Descripción generada automáticamente

51. B

**52-What is the result?**

class Feline {

    public String type = "f ";

    public Feline() {

        System.out.print("feline ");

    }

}

public class Cougar extends Feline {

    public Cougar() {

        System.out.print("cougar ");

    }

    void go() {

        type = "c ";

        System.out.print(this.type + super.type);

    }

    public static void main(String[] args) {

        new Cougar().go();

    }

}

A. Cougar c f.

B. Feline cougar c c.

C. Feline cougar c f.

D. Compilation fails.

**53-What is the result?**

interface Rideable {

    String getGait();

}

public class Camel implements Rideable {

    int weight = 2;

    String getGait() {

        return mph + ", lope"; // Error de compilación: mph no está definido

    }

    void go(int speed) {

        ++speed;

        weight++;

        int walkrate = speed \* weight;

        System.out.print(walkrate + getGait());

    }

    public static void main(String[] args) {

        new Camel().go(8);

    }

}

A. 16 mph, lope.

B. 24 mph, lope.

C. 27 mph, lope.

D. Compilation fails.

**54-¿Cuáles de las siguientes opciones son instanciaciones e inicializaciones válidas de un arreglo multidimensional?**

a-int[][] array2D = {{0, 1, 2, 4}{5, 6}};

   int[][] array2D = new int[2][2];

b- array2D[0][0] = 1;

    array2D[0][1] = 2;

    array2D[1][0] = 3;

    array2D[1][1] = 4;

    int[] array3D = new int[2][2][2];

c-int[][] array3D = {{{0, 1}, {2, 3}, {4, 5}}};

   int[] array = {0, 1};

d-array3D[0][0] = array;

   array3D[0][1] = array;

   array3D[1][0] = array;

   array3D[1][1] = array;