GUÍA DE PREGUNTAS DE APX 1

Exercise 1.-

Consider the following Java code snippet:

```
public int divide (int a, int b){
    int c= -1;

    try{
        c = a/b;
    }
    catch(Exception e){
        System.err.print("Exception ");
    }
    finally{
        System.err.println("Finally ");
    }
    return c;
}
```

What will our code print when we call divide (4,0)?

Pick one of the choices:

- Exception Finally
- Finally Exception
- Exception

Explicación:

Cuando se llama al método divide(4, 0), ocurre una excepción ArithmeticException por intentar dividir entre cero. El bloque catch captura la excepción e imprime "Exception ". Luego, el bloque finally siempre se ejecuta e imprime "Finally ".

Exercise 2.-

The feature which allows different methods to have the same name and arguments type, but the different implementation is called?

Pick one of the choices:

- Overloading(SobreCarga)
- Overriding (SobreEscritura @Override)
- Java does not permit methods with same and type signature
- None of the above

Explicación:

El overriding permite que diferentes métodos tengan el mismo nombre y tipo de argumentos pero con diferentes implementaciones. Esto ocurre cuando un

método en una subclase sobreescribe un método con la misma firma en la superclase.

Exercise 3.-

What does the following for loop output?

```
for (int i=10, j=1, i>j; --i, ++j)
System.out.print(j %i);
```

Pick one of the choices:

- 12321
- 12345
- 11111
- 00000

Explicación:

El bucle comienza con i en 10 y j en 1, y sigue ejecutándose mientras i sea mayor que j. En cada paso, i se decrementa en 1 y j se incrementa en 1, y después imprime el residuo de j dividido por i, de tal manera que esto es lo que sucede:

Paso 1: i = 10	j = 1	→	1/10	Residuo = 1
Paso 1: i = 9	j = 2	→	2/9	Residuo = 2
Paso 1: i = 8	j = 3	→	3/8	Residuo = 3
Paso 1: i = 7	j = 4	→	4/7	Residuo = 4
Paso 1: i = 6	j = 5	→	5/6	Residuo = 5

El bucle termina cuando i ya no es mayor que j, es decir, cuando i=5 y j=6.

Exercise 4.-

We perform the following sequence of actions:

1. Insert the following elements into a set: 1,2,9,1,2,3,1,4,1,5,7.
2. Convert the set into a list and sort it in ascending order.

Which option denotes the sorted list?

Pick one of the choices:

- {1, 2, 3, 4, 5, 7, 9}
- {9, 7, 5, 4, 3, 2, 1}

- {1, 1, 1, 1, 2, 2, 3, 4, 5, 7, 9}
- None of the above

Explicación:

Primero, se insertan los elementos en un Set, que elimina los duplicados. Luego, al convertir el Set en una lista y ordenarla en orden ascendente, se obtiene la lista {1, 2, 3, 4, 5, 7, 9}.

Exercise 5.-

What is the output for the below Java code?

```
public class Test{
    public static void main (String[] args)
     {
        int i = 010;
        int j = 07;
        System.out.println(i);
        System.out.println(j);
    }
}
```

Pick one of the choices:

- 8 7
- 10 7
- Compilation fails with an error at line 3
- Compilation fails with an error at line 5

Explicación:

Tanto los valores de i y j están en notación octal, por lo que en notación decimal 010 equivale a 8 y 07 equivale a 7.

Exercise 6.-

A public data member with the same name is provided in both base as well as derived clases. Which of the following is true?

Pick one of the choices:

- It is a compiler error to provide a field with the same name in both base and derived class.
- The program will compile and this feature is called overloading
- The program will compile and this feature is called overriding
- The program will compile and this feature is called as hiding or shadowing

Explicación:

Cuando se tiene un miembro público con el mismo nombre en una clase base y en una clase derivada, el programa aún así se compilará. El miembro/atributo en la clase derivada simplemente oculta al miembro de la clase base. Esto se conoce como ocultamiento o sombreado.

Exercise 7.-

Given three clases A, B and C.

B is a subclass of A

C is a subclass of B

Which one of these boolean expressions is true only when an object denoted by reference o has actually been instantiated from class B, as opposed to from A or C?

Pick one of the choices:

- (o instanceof B) && (! (o instanceof A))
- (o instanceof B) && (! (o instanceof C))
- (o instanceof B)
- None of the above

Explicación:

No existe una forma directa de verificar si un objeto es una instancia exacta de una clase y que no sea de ninguna de sus subclases. Esto se debe a la naturaleza de la herencia y a cómo funciona el mecanismo instanceof.

Exercise 8.-

Which statement is true?

Pick one of the choices:

- Non-static member classes must have either default or public accessibility
- All nested classes can declare static member classes
- Methods in all nested classes can be declared static
- Static member classes can contain non-static methods

Explicación:

Las clases anidadas estáticas pueden contener tanto métodos estáticos como no estáticos.

Exercise 9.-

A constructor is called whenever

Pick one of the choices:

- An object is declared
- An object is used
- A class is declared
- A class is used

Explicación:

El propósito de un constructor en Java es inicializar un nuevo objeto de una clase. Por lo tanto, el constructor se invoca cuando se crea o se instancia un nuevo objeto de esa clase, no cuando se declara o usa un objeto o clase.

Exercise 11.-

Which of the following data types in Java are primitive?

Pick one of the choices:

- String
- Struct
- Boolean
- char

Explicación:

Los tipos de datos primitivos incluyen int, char, boolean, byte, short, long, float, y double.

Exercise 12.-

Which of the following are true for Java Classes?

Pick one of the choices:

- The Void class extends the Class class
- The Float class extends the Double class
- The System class extends the Runtime class
- The Integer class extends the Number class

Explicación:

Las clases que extienden la clase Number son aquellas que representan tipos numéricos (Wrappers) y proporcionan métodos para convertir esos valores a diferentes tipos numéricos. Entre ellas se incluyen Integer, Float, Double, Long, Short, y Byte.

Exercise 13.-

The following code snippet is a demostration of a particular design pattern. Which desing pattern is it?

```
public class Mystery{
    private static Mystery instance = null;
    protected Mystery(){
        public static Mystery getInstance(){
            if(instance == null){
                instance = new Mystery();
            }
            return instance;
        }
    }
}
```

Pick one of the choices:

- Factory Design Pattern
- Strategy Pattern
- Singleton
- Facade Design Pattern

Explicación:

Aunque el código tiene errores, la opción es Singleton, ya que garantiza que se cree una única instancia de la clase en toda la aplicación y proporciona un punto de acceso global a esa instancia.

Exercise 14.-

Which of the following Java declaration of the String array is correct?

Pick one of the choices:

- String temp [] = new String {“j”, “a”, “z”};
- String temp [] = {“j” “b” “c”};
- String temp = {“a”, “b”, “c”};
- String temp [] = {“a”, “b”, “c”};

Explicación:

Esta declaración inicializa correctamente un arreglo de String con los valores especificados entre llaves. Las otras opciones contienen errores de sintaxis o tipo de datos.

Exercise 15.-

Which is true of the following program?

```java
package exam.java;

public class TestFirstApp {
    static void doIt(int x, int y, int m) {
        if(x==5) m=y;
            else m=x;
    }

    public static void main(String[] args) {
        int i=6, j=4, k=9;
        TestFirstApp.doIt(i, j, k);
            System.out.println(k);
    }
}
```

Pick one of the choices

- Doesn't matter what the values of *i* and *j* are, the output will always be *5*.
- Doesn't matter what the values of *k* and *j* are, the output will always be *5*.
- Doesn't matter what the values of *i* and *j* are, the output will always be *9*.
- Doesn't matter what the values of *k* and *j* are, the output will always be *9*.

Explicación:

En Java, cuando se pasan parámetros a un método, se hace por valor. Es decir, el método recibe una copia del valor original, no una referencia directa a la variable original. Por eso, independientemente de lo que ocurra dentro del método doIt, el valor de k en main seguirá siendo 9, y eso es lo que se imprimirá.

Exercise 16.-

Which of the following statements are correct. Select the correct answer.

- Each Java file must have exactly one package statement to specify where the class is stored.
- If a java file has both import and package statement, the import statement must come before package statement.
- A java file has at least one class defined
- If a java file has a package statement, it must be the first statement (except comments).

Explicación:

Si se incluye una declaración de paquete, debe ser la primera declaración en el archivo, precedida solo por comentarios. Las declaraciones import deben ir después de la declaración del paquete si están presentes.

Exercise 17.-

What is the output for the following program:

```
class Constructor
        {
                static String str;
                public void Constructor(){
                        System.out.println("In constructor");
                        str = "Hello World";
                }
                public static void main (String [] args){
                        Constructor C = new Constructor ();
                        System.out.println(str);
                }
        }
```

Pick one of the choices

- In Constructor
- Null
- Compilation Fails
- None of the above

Explicación:

La variable str se declara como un atributo estático, y dado que no se inicializa explícitamente, por defecto se le asigna null. La única inicialización de la variable ocurre dentro de un método, no en el constructor. Debido a que este método no se llama durante la ejecución del programa, el valor de str permanece null. Es importante destacar que el método de inicialización no es el constructor, sino un método normal, lo que podría llevar a la confusión si se asume que el constructor se encarga de la inicialización.

Exercise 18.-

Given the following code, what is the most likely result.

```
import java.util.*;

public class Compares {

    public static void main(String[] args) {

    String [] cities= {"Bangalore","Pune","San Francisco","New York City"};
    MySort ms= new MySort();
    Arrays.sort(cities,ms);
    System.out.println(Arrays.binarySearch(cities,"New York City" ));
}
    static class MySort implements Comparator{
        public int compare(String a, String b){

            return b.compareTo(a);
        }

    }


}
```

Pick one of the choices

- -1
- 1
- 2
- Compilation fails

Explicación:

El código no compilará debido a un problema con la interfaz Comparator. La interfaz Comparator debe ser parametrizada con el tipo de dato que va a comparar, en este caso String.

Exercise 19.-

To delete all pairs of keys and values in a given HashMap, which of the following methods should be used?
Pick one of the choices

a) clearAll()
b) empty()
c) remove()
d) clear()

Explicación:

El método clear() elimina todas las entradas del HashMap, dejándolo vacío. Los otros métodos mencionados (clearAll(), empty(), y remove()) no son métodos de la clase HashMap.

Exercise 20.-

Which pattern do you see in the code below:
java.util.Calendar.getInstance();
Pick one of the choices

a) Singleton Pattern
b) Factory Pattern
c) Facade Pattern
d) Adaptor Pattern

Explicación:

Calendar.getInstance() usa este patrón para crear un objeto Calendar sin que necesites conocer la clase exacta (como GregorianCalendar). El método decide qué tipo de objeto devolver según la configuración y proporciona la instancia adecuada.

Exercise 21.-

What is the output of the following program:

```
interface BaseI { void method (); }
        class BaseC
        {
                public void method()
                {
                        System.out.println("Inside BaseC: :method");
                }
        }
        class ImplC extends BaseC implements BaseI
        {
                public static void main (String []s)
                {
                        (new ImplC()).method();
                }
        }
```

Pick one of the choices

a) Null
b) Complication fails
c) Inside BaseC::method
d) None of the above

Explicación:

La clase ImplC extiende BaseC e implementa la interfaz BaseI, pero no necesita implementar el método method() de la interfaz, ya que BaseC ya lo proporciona.

Por lo tanto, cuando se crea una instancia de ImplC y se llama al método method(), se utiliza el método heredado de BaseC, que imprime "Inside BaseC::method".

Exercise 22.-

Consider the following three classes:

```
class A {}
    class B extends A {}
    class C extends B {}
```

Consider n object of class B is instantiated, i.e.,

```
B b = new B();
```

Which of the following Boolean expressions evaluates to true:
Pick one of the choices

a) (b instanceof B)
b) (b instanceof B) && (!(b instanceof A))
c) (b instanceof B) && (!(b instanceof C))
d) None of the above

Explicación:

La expresión (b instanceof B) && (!(b instanceof C)) es la única opción que es verdadera porque un objeto de tipo B siempre será una instancia de B y de A, pero no será una instancia de C.