Natalia Denise Fuentes Rubio
Semana 4 – Evidencias
Academia Java

1. Which declaration initializes a boolean variable?
   a) boolean m = null
   b) Boolean j = (1<5)
   c) boolean k = 0
   d) boolean h = 1

   El argumento 1<5 dará como resultado un boolean con valor de true, y será guardado dentro de Boolean que es un wraper

2. What is the DTO pattern used for?

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

   El Data Transfer Object es un patrón utilizado para tranferir daos entre procesos, generalmente, a través de límites de red, de forma encapsulada.

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

```
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);
    }
}
```

   a) -1
   b) 1
   c) 5
   d) 8
   e) 11

Al iniciar en kk = 11, se realizarán 5 iteraciones para cumplir con la condición >6,

4. ¿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;
   }
}
```

a) Compilation fails
b) An exception is thrown at runtime
c) There is no result because this is not correct way to determine the hash code
d) Hash is: 111111, 44444444, 999999999.

sc es una instancia de SampleClass, scA es una instancia de SampleClassA y scB es una instancia de SampleClassB. Tanto scA y scB están haciendo un Override al método getHash();. El método main imprimirá los valores de las instancias, dando como resultado sc 111111, scA 44444444 y scB 999999999

5. ¿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);
```

```
        }
}
```

a) 0
b) 0 1 2
c) 0 1 2 0 1 2 0 1 2
d) Compilation fails

Iniciará en 0, se incrementa en 1 por lo que ya es 1 que sigue siendo menos que la longitud, se incrementa en 1 la i y ahora 2, por lo que sigue siendo menor a la longitud, por lo que imprime 012

6. ¿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;
            while (ii < table.length) {
                System.out.println(ss + ", " + ii);
                ii++;
            }
        }
    }
}
```

```
public class DoCompare1 {
    public static void main(String[] args) {
        String[] table = {"aa", "bb", "cc"};
        for (String ss : table) {
            int ii = 0;
            while (ii < table.length) {
                System.out.println(ss + ", " + ii);
                ii++;
            }
        }
    }
}
```

a) Zero.
b) Once.
c) Twice
d) Thrice

e) Compilation fails

Las impresiones serán aa, 0, aa, 1, aa, 2, bb, 0, bb, 1, bb, 2, cc, 0, cc, 1, cc, 2. Por lo que el 2, se imprime tres veces

7. What code should be inserted?

```
4.    public class Bark {
5.        // Insert code here - Line 5
6.              public abstract void bark();
7.        }
8.
9.        // Insert code here - Line 9
10.             public void bark() {
11.                   System.out.println("woof");
12.             }
13.       }
14.   }
```

a) 5. class Dog { 9. public class Poodle extends Dog {
b) 5. abstract Dog { 9. public class Poodle extends Dog {
c) 5. abstract class Dog { 9. public class Poodle extends Dog {
d) 5. abstract Dog { 9. public class Poodle implements Dog {
e) 5. abstract Dog { 9. public class Poodle implements Dog {
f) 5. abstract class Dog { 9. public class Poodle implements Dog {

Para el que código funcione, se debe definir una class abtract Dog que tenga el método public abstract void bark(); y en la línea 9, se va a heredar la clase Dog para que pueda realizar el bark mediante un Override

8. Wich statement initializes a stringBuilder to a capacity of 128?

a) StringBuilder sb = new String("128");
b) StringBuilder sb = StringBuilder.setCapacity(128); C.
c) StringBuilder sb = StringBuilder.getInstance(128); D.
d) StringBuilder sb = new StringBuilder (128);

La opción d inicializa un StringBuilder con una capacidad inicial de 128 caracteres.

9. 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();
        }
}
```

a) 20
b) 100
c) 1000
d) 2

Se inicial la variable num en 100, en el método main se crea una nueva Calculator con un obj, se llama el método calc para darle un argumento 2, por lo que al ser multiplicado por 10, dará como resultado 20.

10. What three modifications, made independently, made to class Greet, enable the code to compile and run?

```
package handy.dandy;
public class KeyStroke {
        public void typeExclamation() {
                System.out.println("!");
        }
}
```

And:

```
01. package handy;
02.
03.
04. public class Greet {
05.         public static void main(String[] args) {
06.                 String greeting = "Hello";
07.                 System.out.print(greeting);
08.                 KeyStroke stroke = new KeyStroke();
09.                 stroke.typeExclamation();
10.         }
11. }
```

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.

Natalia Denise Fuentes Rubio
Semana 4 – Evidencias
Academia Java

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)?

a) Exception Finally
b) Finally Exception
c) Exception

c se inicializa como -1, en el try, se pretende realizar c=4/0, pero al no ser matemáticamente posible, lanzará una ArithmeticException.
En el catch, se cachara la excepción, imprimiendo Exception.
El finally se ejecuta se haya efectuado una excepción o no, por lo que se imprimirá el Finally

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

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

3. What does the following for loop output?

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

a) 12321
b) 12345
c) 11111
d) 00000

El loop itera mientras i > j, j % i calcula el resto de la división. Entonces irá sacando los restos de la división los cuáles irán dando resultados de 1 para 1%10, 2 para 1%9 y así sucesivamente hasta llegar a 5 para 5%6.

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?

a) {1, 2, 3, 4, 5, 7, 9}
b) {9, 7, 5, 4, 3, 2, 1}
c) {1, 1, 1, 1, 2, 2, 3, 4, 5, 7, 9}
d) None of the above

Al convertirlo en un Set, almacenará solo los elementos no duplicados, aquellos duplicados serán removidos automáticamente, posteriormente, se van a ordenar de menor a mayor.

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);
    }
}
```

a) 8 7
b) 10 7
c) Compilation fails with an error at line 3
d) Compilation fails with an error at line 5

Son variables declaradas con números octales ya que empiezan con 0, el 07 será un 7, el 010 será un 8 ya que está es su equivalencia en octales.

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?

a) It is a compiler error to provide a field with the same name in both base and derived class

b) The program will compile and this feature is called overloading
c) The program will compile and this feature is called overriding
d) The program will compile and this feature is called as hiding or shadowing

7. Which statement is true?

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

Las clases estáticas pueden tener tanto métodos estáticos como no estáticos.

8. A constructor is called whenever

a) An object is declared
b) An object is used
c) A class is declared
d) A class is used

Un constructor es un método especial en una clase que se llama en automático cuando un objeto de la clase es declarado.

9. Which of the following data types in Java are primitive?

a) String
b) Struct
c) Boolean
d) Char

Char es el único tipo de dato primitivo, los demás son

10. Which of the following are true for Java Classes?

a) The Void class extends the Class class
b) The Float class extends the Double class
c) The System class extends the Runtime class
d) The Integer class extends the Number class

Integer, por si solo hereda de la clase Number. La clase Number es una clase abstracta que hereda a los Integer, Double, Float, etc…

11. 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;
        }
    }
}
```

a) Factory Design Pattern
b) Strategy Pattern
c) Singleton
d) Facade Design Pattern

Posee las características del patrón; un constructor protegido, una instancia estática y un método estático.

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

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

Posee las características para declarar un arreglo, los [], los elementos separados por comas (,).

13. Which is true of the following program?

```
1  package exam.java;
2
3  public class TestFirstApp {
4      static void doIt(int x, int y, int m) {
5          if(x==5) m=y;
6              else m=x;
7      }
8
9      public static void main(String[] args) {
10         int i=6, j=4, k=9;
11         TestFirstApp.doIt(i, j, k);
12             System.out.println(k);
13     }
14 }
```

a) Doesn't matter what the values of *i* and *j* are, the output will always be *5*.
b) Doesn't matter what the values of *k* and *j* are, the output will always be *5*.
c) Doesn't matter what the values of *i* and *j* are, the output will always be *9*.
d) Doesn't matter what the values of *k* and *j* are, the output will always be *9*.

La variable k no se verá afetada por el método doIt, ya que este modifica a la variable m que es una copia local.

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

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

El package, siempre debe ser la primera declaración en el archivo, incluso antes de los imports. Los comentarios, al ser comentarios no influyen dentro de esto.

15. 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);
        }

    }


}
```

a) -1
b) 1
c) 2
d) Compilation fails

MySort implementa Comparator, pero no específica el tipo del parámetro. Además, el metódo inarySearch requiere un array ordenado de acuerdo al comparador.

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

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

Este método remueve todos los key – values del HashMap.

17. Which pattern do you see in the code below:

java.util.Calendar.getInstance();

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

Este patrón se utiliza cuando se crean instancias de la clase sin exponer la instanciación lógica. Tal es el caso de java.util.Calendar.getInstance();

18. 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();
                }
        }
```

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

BaseI es una interface con un method(), BaseC es una clase que implementa method(), ImplC hereda de BaseC e implementa BaseC. Cuando se crea la instancia de ImplC y se llamal a method(), se ejecutará el definido en BaseC.

19. 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:

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

b instance of B es verdadero. A su vez, b no es una instancia de C, pero C si es una subclase de B, por lo que (!(b instanceof C)) también en verdadero.

20. What is the output of 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);
        }
}
```

a) In Constructor
b) Null
c) Compilation fails
d) None of the above

Se está utilizando la palabra reservada de Constructor para definir un método. Al ser un método, no se llamará al crear su instancia, por lo que nunca se inicializará el "Hello World", al no declararse, se quedará con el valor de null.