

1. Which of the following are legal entry point methods that can be run from the command line? (Choose all that apply.)
- A. `private static void main(String[] args)`
 - B. `public static final main(String[] args)`
 - C. `public void main(String[] args)`
 - D. `public static final void main(String[] args)`
 - E. `public static void main(String[] args)`
 - F. `public static main(String[] args)`

Legal entry point method: se ejecuta desde la línea de comandos necesita un punto de entrada específico para comenzar. Este punto de entrada estándar es el método `main`, que debe cumplir con ciertos requisitos en su firma para ser "legal" o "válido". Si el método `main` no cumple con esta firma específica, la JVM no lo reconocerá como el punto de inicio, y no podrás ejecutar el programa directamente desde la línea de comandos.

Requisitos:

- public:**
 - El método debe ser público para que la JVM pueda acceder a él desde fuera de la clase. Si el método no es `public`, la JVM no podrá ejecutarlo, y lanzará un error.
- static:**
 - Debe ser `static` porque la JVM necesita llamarlo sin crear una instancia de la clase. Al ser `static`, se puede ejecutar sin tener que crear un objeto de la clase en la que se encuentra.
- void:**
 - Debe tener un tipo de retorno `void` porque no se espera que devuelva un valor. El método simplemente inicia la ejecución del programa y no debería retornar ningún resultado.
- String[] args:**
 - Debe aceptar un parámetro de tipo `String[]` llamado `args`. Este parámetro permite pasar argumentos al programa desde la línea de comandos.

```
public static void main(String[] args){  
    // Código del programa }
```

También sería posible:

```
public static void main(String args []) {  
    System.out.println("Hola Mundo 17");  
}  
  
public static void main(String patitos[] ) {  
    System.out.println("Hola Mundo 21");  
}
```

Arg: es un nombre se puede poner cualquier cosa.

```
public static void main(String... patitos ) {  
    System.out.println("Hola Mundo 21");  
}
```

... : Baracks?

```
final static public void main(String... patitos) throws Exception {  
    System.out.println("Hola Mundo 23");  
}
```

Agregando la excepción

2. Which answer options represent the order in which the following statements can be assembled into a program that will compile successfully? (Choose all that apply.)

X: `class Rabbit {}`
Y: `import java.util.*;`
Z: `package animals;`

- A. X, Y, Z
- B. Y, Z, X
- C. Z, Y, X
- D. Y, X
- E. Z, X
- F. X, Z
- G. None of the above

Primero deben llamarse a los paquetes porque si ponemos al principio las importaciones el código no compila.
El paquete no es obligatorio ni las importaciones a menos que tengamos otras clases que necesitemos usar.

3. Which of the following are true? (Choose all that apply.)

```
public class Bunny {  
    public static void main(String[] x) {  
        Bunny bun = new Bunny();  
    } }
```

- A. **Bunny** is a class.
- B. **bun** is a class.
- C. **main** is a class.
- D. **Bunny** is a reference to an object.
- E. **bun** is a reference to an object.
- F. **main** is a reference to an object.
- G. The **main()** method doesn't run because the parameter name is incorrect.

La clase tiene el nombre de **Bunny**
Bun, no es una clase sino una variable de referencia
No importa que nombre le pongas al parametro del metodo main, va a funcionar correctamente
Main no es una clase es un método

4. Which of the following are valid Java identifiers? (Choose all that apply.)

- A. **_**
- B. **_helloWorld\$**
- C. **true**
- D. **java.lang**
- E. **Public**
- F. **1980_s**
- G. **_Q2_**

No se pueden usar valores primitivos como identificadores

Reglas para identificadores en Java

- Los identificadores pueden contener letras, dígitos, guiones bajos (**_**), y signos de dólar (**\$**)
- No pueden comenzar con un dígito:
- No pueden usar palabras reservadas de Java:
- Sensible a mayúsculas y minúsculas:
- Longitud ilimitada:
- No debe contener espacios:

Ejemplos de identificadores válidos e inválidos

Identificador	Válido	Razón
nombre	✓	Sigue todas las reglas
_edad	✓	Comienza con _
\$total	✓	Comienza con \$
2nombre	✗	Comienza con un número
public	✗	Es una palabra reservada
nombre completo	✗	Contiene un espacio
MAX_VALOR	✓	Sigue la convención para constantes

5. Which statements about the following program are correct? (Choose all that apply.)

```
1: public class Bear {  
2:     private Bear pandaBear;  
3:     private void roar(Bear b) {  
4:         System.out.println("Roar!");  
5:         pandaBear = b;  
6:     }  
7: }  
8: public static void main(String[] args) {  
9:     Bear brownBear = new Bear();  
10:    Bear polarBear = new Bear();  
11:    brownBear.roar(polarBear);  
12:    polarBear = null;  
13:    brownBear = null;  
14:    System.gc(); } }
```

A. The object created on line 9 is eligible for garbage collection after line 13.
B. The object created on line 9 is eligible for garbage collection after line 14.
C. The object created on line 10 is eligible for garbage collection after line 12.
D. The object created on line 10 is eligible for garbage collection after line 13.
E. Garbage collection is guaranteed to run.
F. Garbage collection might or might not run.
G. The code does not compile.

F - al poner .gc // al mandarlo a llamar no es seguro que se ejecute

Se va desde la linea 9 porque la referencia ya no apunta a nada, es null

El objeto creado en la linea 10 pasa a garbage collection despues de la 13 porque pierde su variable de referencia, ya no apunta a nada.

//

Bloques de inicializacion, necesitan una instancia para poderse ejecutar

1. Primero se ejecutan los bloques anonimos Static
2. Bloques de inicializacion de instancia
3. Constructores

//

6. Assuming the following class compiles, how many variables defined in the class or method are in scope on the line marked on line 14?

```
1: public class Camel {  
2:     { int hairs = 3_000_0; }  
3:     long water, air=2;  
4:     boolean twoHumps = true;  
5:     public void spit(float distance) {  
6:         var path = "";  
  
7:         { double teeth = 32 + distance++; }  
8:         while(water > 0) {  
9:             int age = twoHumps ? 1 : 2;  
10:            short i=-1;  
11:            for(i=0; i<10; i++) {  
12:                var Private = 2;  
13:            }  
14:        // SCOPE  
15:        }  
16:    }  
17: }
```

- A. 2
B. 3
C. 4
D. 5
E. 6
F. 7
G. None of the above

7 variables que se pueden ver

2: solo se ve desde bloque anonimo

7: porque un bloque esta dentro de otro bloque

12: var. tampoco se puede porque esta dentro del for

//

7. Which are true about this code? (Choose all that apply.)

```
public class KitchenSink {  
    private int numForks;  
  
    public static void main(String[] args) {  
        int numKnives;  
        System.out.print("");  
        "# forks = " + numForks +  
        "# knives = " + numKnives +  
        "# cups = 0");  
    }  
}
```

- A. The output includes: # forks = 0.
- B. The output includes: # knives = 0.
- C. The output includes: # cups = 0.
- D. The output includes a blank line.
- E. The output includes one or more lines that begin with whitespace.
- F. The code does not compile.

Bloques de String / Cadenas multilinea;

Todo lo que venga en tres comillas aparece en texto —“”—

Se puede incluir comillas dobles dentro del texto, no afecta.

Elimina espacios a la izquierda dependiendo la identacion que le demos al texto.

C - Imprime # cups = 0

//

8. Which of the following code snippets about var compile without issue when used in a method? (Choose all that apply.)

```
A. var spring = null;  
B. var fall = "leaves";  
C. var evening = 2; evening = null;  
D. var night = Integer.valueOf(3);  
E. var day = 1/0;  
F. var winter = 12, cold;  
G. var fall = 2, autumn = 2;  
H. var morning = ""; morning = null;
```

Var;

- la JVM NO puede inferir en el tipo de dato.
- No se pueden definir varios tipos de variables/ valores.
- Solo para declarar variables locales dentro de metodos, constructores y bloques de inicializacion.
- Inicializacion obligada en la misma linea.

```
var day = 1/0;
```

En compilacion no hay error. Solo en ejecucion.

//

9. Which of the following are correct? (Choose all that apply.)

- A. An instance variable of type float defaults to 0.
- B. An instance variable of type char defaults to null.
- C. A local variable of type double defaults to 0.0.
- D. A local variable of type int defaults to null.
- E. A class variable of type String defaults to null.
- F. A class variable of type String defaults to the empty string "".
- G. None of the above.

Default de Float = 0.0;

"" chart = valor representacionde AACSI;

Ningun primitivo es Null;

Double es variable local, no tuene variable local.

E-Todos los objetos tienen por default un null;

//

10. Which of the following expressions, when inserted independently into the blank line, allow the code to compile? (Choose all that apply.)

```
public void printMagicData() {  
    var magic = _____;  
    System.out.println(magic);  
}
```

- A. 3_1
- B. 1_329_.0
- C. 3_13.0_
- D. 5_291..2
- E. 2_234.0_0
- F. 9___6
- G. _1_3_5_0

Uso de guiones en numeros;

- Guion bajo no puede ir antes ni despues del punto decimal
- Guion no puede ir al final
- Guion no puede ir al principio/

//

11. Given the following two class files, what is the maximum number of imports that can be removed and have the code still compile?

```
// Water.java
package aquarium;
public class Water { }

// Tank.java
package aquarium;
import java.lang.*;
import java.lang.System;
import aquarium.Water;
import aquarium.*;
public class Tank {
    public void print(Water water) {
        System.out.println(water); } }
```

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. Does not compile

Java.lang - esta por default / system / String

Los que incluyen java.lang se eliminan porque estan por default;
Clase aquarium, estan en el mismo paquete

//

12. Which statements about the following class are correct? (Choose all that apply.)

```
1: public class Clownfish {
2:     int gills = 0, double weight=2;
3:     { int fins = gills; }
4:     void print(int length = 3) {
5:         System.out.println(gills);
6:         System.out.println(weight);
7:         System.out.println(fins);
8:         System.out.println(length);
9:     } }
```

- A. Line 2 generates a compiler error.
- B. Line 3 generates a compiler error.
- C. Line 4 generates a compiler error.
- D. Line 7 generates a compiler error.
- E. The code prints 0.
- F. The code prints 2 .0.
- G. The code prints 2.
- H. The code prints 3.

2; Error de compilacion, se pueden definir varias variables en una misma linea pero debe de ser del mismo tipo.

4; no se puede inicializar una variable local

7; Vive solo dentro del bloque de inicializacion

//

13. Given the following classes, which of the following snippets can independently be inserted in place of INSERT IMPORTS HERE and have the code compile? (Choose all that apply.)

```
package aquarium;
public class Water {
    boolean salty = false;
}

package aquarium.jellies;
public class Water {
    boolean salty = true;
}

package employee;
INSERT IMPORTS HERE
public class WaterFiller {
    Water water;
}

A. import aquarium.*;
B. import aquarium.Water;
import aquarium.jellies.*;
C. import aquarium.*;
import aquarium.jellies.Water;
D. import aquarium.*;
import aquarium.jellies.*;
E. import aquarium.Water;
import aquarium.jellies.Water;
F. None of these imports can make the code compile.
```

Se tiene que definir de forma explícita cual clase importar para que no se confunda.

//

14. Which of the following statements about the code snippet are true? (Choose all that apply.)

```
3: short numPets = 5L;  
4: int numGrains = 2.0;  
5: String name = "Scruffy";  
6: int d = numPets.length();  
7: int e = numGrains.length;  
8: int f = name.length();
```

- A. Line 3 generates a compiler error.
- B. Line 4 generates a compiler error.
- C. Line 5 generates a compiler error.
- D. Line 6 generates a compiler error.
- E. Line 7 generates a compiler error.
- F. Line 8 generates a compiler error.

3: error, long no cabe en un short

4: error, no cabe en un double;

6:

7: Los primitivos no tienen ni atributos, ni propiedades, ni metodos.

//

15. Which of the following statements about garbage collection are correct? (Choose all that apply.)

- A. Calling `System.gc()` is guaranteed to free up memory by destroying objects eligible for garbage collection.
- B. Garbage collection runs on a set schedule.
- C. Garbage collection allows the JVM to reclaim memory for other objects.
- D. Garbage collection runs when your program has used up half the available memory.
- E. An object may be eligible for garbage collection but never removed from the heap.
- F. An object is eligible for garbage collection once no references to it are accessible in the program.
- G. Marking a variable `final` means its associated object will never be garbage collected.

-Llamar al Garbage collection no es una garantía de que libere memoria

-Limpia la memoria de variables a las que ya no tengan referencia.

//

Saltos de linea

Formatting	Meaning in regular String	Meaning in text block
\"	"	"
\\"\\\"\\\"	n/a - invalid	"""
\\"\\\"	""	""
Space (at end of line)	Space	Ignored
\s	Two spaces (\s is a space and preserves leading space on the line)	Two spaces
\ (at end of line)	n/a - Invalid	Omits new line on that line

16. Which are true about this code? (Choose all that apply.)

```
var blocky = """  
    squirrel \s  
    pigeon \\  
    termite""";  
System.out.print(blocky);
```

- A. It outputs two lines.
- B. It outputs three lines.
- C. It outputs four lines.
- D. There is one line with trailing whitespace.
- E. There are two lines with trailing whitespace.
- F. If we indented each line five characters, it would change the output.

//

17. What lines are printed by the following program? (Choose all that apply.)

```
1: public class WaterBottle {  
2:     private String brand;  
3:     private boolean empty;  
4:     public static float code;  
5:     public static void main(String[] args) {  
6:         WaterBottle wb = new WaterBottle();  
  
7:         System.out.println("Empty = " + wb.empty);  
8:         System.out.println("Brand = " + wb.brand);  
9:         System.out.println("Code = " + code);  
10:    } }
```

- A. Line 8 generates a compiler error.
- B. Line 9 generates a compiler error.
- C. Empty =
- D. Empty = false
- E. Brand =
- F. Brand = null
- G. Code = 0.0
- H. Code = 0f

Variables de instancia y static, tienen valores por default;

//

18. Which of the following statements about var are true? (Choose all that apply.)

- A. A var can be used as a constructor parameter.
- B. The type of a var is known at compile time.
- C. A var cannot be used as an instance variable.
- D. A var can be used in a multiple variable assignment statement.
- E. The value of a var cannot change at runtime.
- F. The type of a var cannot change at runtime.
- G. The word var is a reserved word in Java.

Var:

- Var es conocido en tiempo de compilacion.
- Solo se puede usar dentro de variables locales.
- Sin asignacion multiple.
- Se cambia el valor en tiempo de ejecucion.
- No se puede cambiar el type.

//

19. Which are true about the following code? (Choose all that apply.)

```
var num1 = Long.parseLong("100");  
var num2 = Long.valueOf("100");  
System.out.println(Long.max(num1, num2));
```

- A. The output is 100.
- B. The output is 200.
- C. The code does not compile.
- D. num1 is a primitive.
- E. num2 is a primitive.

**** Pendiente

//

20. Which statements about the following class are correct? (Choose all that apply.)

```
1: public class PoliceBox {  
2:     String color;  
3:     long age;  
4:     public void PoliceBox() {  
5:         color = "blue";  
6:         age = 1200;
```

```

7:     }
8:     public static void main(String []time) {
9:         var p = new PoliceBox();
10:        var q = new PoliceBox();
11:        p.color = "green";
12:        p.age = 1400;
13:        p = q;
14:        System.out.println("Q1="+q.color);
15:        System.out.println("Q2="+q.age);
16:        System.out.println("P1="+p.color);
17:        System.out.println("P2="+p.age);
18:    } }

```

- A. It prints Q1=blue.
- B. It prints Q2=1200.
- C. It prints P1=null.
- D. It prints P2=1400.
- E. Line 4 does not compile.
- F. Line 12 does not compile.
- G. Line 13 does not compile.
- H. None of the above.

Uso de BigDecimal por imprescion de los primitivos;

En la linea 13: se vuelve a inicializar y se ponen los valores por default
Todo lo que tenga age = null y lo que tenga color= 0;

//

- 21.** What is the output of executing the following class?

```

1:  public class Salmon {
2:      int count;
3:      { System.out.print(count+"-"); }
4:      { count++; }
5:      public Salmon() {
6:          count = 4;
7:          System.out.print(2+"-");
8:      }
9:      public static void main(String[] args) {
10:         System.out.print(7+"-");
11:         var s = new Salmon();
12:         System.out.print(s.count+"-"); } }

```

- A. 7-0-2-1-
- B. 7-0-1-
- C. 0-7-2-1-
- D. 7-0-2-4-
- E. 0-7-1-
- F. The class does not compile because of line 3.
- G. The class does not compile because of line 4.
- H. None of the above.

Se ejecutan:

1. variables static
2. Bloques de inicializacion
3. Constructores

//

- 22.** Given the following class, which of the following lines of code can independently replace
INSERT CODE HERE to make the code compile? (Choose all that apply.)

```

public class Price {
    public void admission() {
        INSERT CODE HERE
        System.out.print(amount);
    } }

```

- A. int Amount = 0b11;
- B. int amount = 9L;
- C. int amount = 0xE;
- D. int amount = 1_2_0;
- E. double amount = 1_0_.0;
- F. int amount = 0b101;
- G. double amount = 9_2.1_2;
- H. double amount = 1_2_.0_0;

C- numero hexadecimal cabe en int

F-numero binaria en int;

G-Uso correcto de los guiones

//

23. Which statements about the following class are true? (Choose all that apply.)

```
1:  public class River {  
2:      int Depth = 1;  
3:      float temp = 50.0;  
4:      public void flow() {  
5:          for (int i = 0; i < 1; i++) {  
6:              int depth = 2;  
7:              depth++;  
8:              temp--;  
9:          }  
10:         System.out.println(depth);  
11:         System.out.println(temp);  
12:     public static void main(String... s) {  
13:         new River().flow();  
14:     } }
```

- A. Line 3 generates a compiler error.
- B. Line 6 generates a compiler error.
- C. Line 7 generates a compiler error.
- D. Line 10 generates a compiler error.
- E. The program prints 3 on line 10.
- F. The program prints 4 on line 10.
- G. The program prints 50.0 on line 11.
- H. The program prints 49.0 on line 11.

3: Genera error de compilacion porque un double no cabe en un float, le falta la F al final.
10: El scope de la variable depth solo esta en el for, es local, no funciona cuando esta fuera

//