

Chapter 1: Building Blocks

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)`

Para que el método main sea correcto tiene que ser static y de tipo void ya que no devuelve ningún valor. El método main tampoco puede ser privado. En la D aunque el final no es necesario ponerlo es una opción válida y la E es la opción por defecto.

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

Un package no es necesario, en los casos que si se tiene va al inicio. Los import tampoco son necesarios, pero siempre van antes de las clases.

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.

Bunny es una clase, bun es una variable de referencia y main es el método principal. El parámetro del main puede ser cualquiera.

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_

En versiones actuales de Java no es posible crear identificadores que sean solo un guion bajo. La palabra true no puede ser usada como identificador porque es una palabra reservada de Java usada en los booleanos. Los identificadores no pueden llevar punto, tampoco pueden iniciar con números.

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

```
2: public class Bear {  
3:     private Bear pandaBear;  
4:     private void roar(Bear b) {  
5:         System.out.println("Roar!");  
6:         pandaBear = b;  
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.

La clase tiene una variable de instancia de tipo Bear llamada pandaBear, dentro del método roar a esta variable se le asigna lo que se pase como parámetro en el método. En la línea 11 se llama al método roar por medio de la variable brownBear y se le pasa como parámetro polarBear por lo tanto, polarBear y la variable de instancia polarBear del objeto de brownBear apuntan al mismo objeto. En la línea 12 polarBear pierde la referencia al objeto al que apuntaba, pero la variable polarBear dentro del objeto BrownBear sigue apuntando a este. En la línea 13 brownBear pierde la referencia al objeto al que apuntaba, en este momento ese objeto se queda sin una referencia, por lo que es elegible para el garbage collection. Si el objeto que contiene a la variable polarBear es eliminado, el objeto al que apunta polarBear queda sin referencia por lo que también es elegible para el garbage collection en la línea 13. Aunque se use el System.gc() para el garbage collection, no es garantía de que se ejecutará.

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

Las variables que se crean en bloques de instancia son locales, por esto las variables de las líneas 2 y 7 no están en el scope de la línea 14. El for termina antes de la línea 14, por esto la variable definida dentro de este no está dentro del scope. El método spit tiene un parámetro, este también se cuenta como una variable ya que la línea 14 está dentro del método.

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;**

Con var no se puede definir una variable que sea null, tampoco se pueden definir más de una variable en una sola línea y siempre se deben inicializar. El valor de estas variables sí puede cambiar a null una vez creadas. La opción E no dará error de compilación, pero sí un error en tiempo de ejecución ya que no es posible dividir por 0.

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.

El valor por defecto de una variable de instancia `float` es `0.0`. Los tipos de datos primitivos no pueden tener el valor `null`. Las variables locales no se pueden inicializar por defecto. El valor por defecto de los objetos es `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`

El `_` no se puede poner antes o después de puntos, tampoco al inicio o final de los números. Se pueden poner cualquier cantidad de `_` entre dos números ya que solo son una referencia visual y no afectan al código.

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

Los import de `aquarium` son innecesarios porque ambas clases están en el mismo paquete. El paquete `java.lang` se importa por defecto en todos los programas de Java, por lo tanto, también son innecesarios esos imports.

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.

Dos variables de diferente tipo no pueden ser declaradas en una misma línea. Los parámetros de los métodos no pueden ser inicializados. No se puede acceder a variables de bloques de instancia desde afuera, ya que son locales.

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.

Al tener dos imports apuntando a dos paquetes con clases con el mismo nombre, esta se debe especificar de manera explícita en uno de ellos. No es posible que los dos imports apunten de manera explícita a una clase con el mismo nombre, tampoco se puede poner el * en los dos imports ya que no se sabría cuál de las dos clases tomar.

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.

No se puede almacenar un valor de tipo long en una variable de tipo short, lo mismo para un valor decimal en un tipo int. El método length() no puede ser usado en tipos primitivos.

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.

Aunque se use `System.gc()`, no se garantiza que el garbage collection se ejecutara, es por esto que un objeto puede no ser eliminado a pesar de ser elegible para el garbage collection. Un objeto es elegible cuando ya no está asignado a una variable de referencia.

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

Las tres variables al ser dos de instancia y una estática se inicializan dependiendo su tipo de dato, los objetos a null, los boolean a false y los float a 0.0, estos valores son los que se imprimen.

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.

No es posible usar `var` para definir parámetros, ya que se necesitan inicializar las variables para saber su tipo de dato, además `var` solo puede ser usada para definir variables locales y no se puede cambiar el tipo de dato.

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.

El método `parseLong()` transforma un valor a un primitivo `long` haciendo que la variable `num1` sea de este tipo. El método `valueOf()` pasa un valor primitivo o un `string` al tipo `Long`. El método `max()` devuelve el valor máximo, como ambos son iguales devuelve ese valor.

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.

Las variables de instancia no se crean con un valor, por lo que se inicializan a su valor por defecto. El método nunca se llama, por lo que el valor de las variables no cambia. A pesar de que en `main` se le da un valor a las variables del objeto de `p`, más adelante a esta variable se le cambia la referencia al objeto de `q` que tiene las variables con los valores por defecto.

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.

Primero se ejecuta la línea 10 que está dentro del main ya que aún no se crea un objeto para que se puedan ejecutar los bloques de instancia y el constructor de la clase. Una vez que se crea el método se ejecutan primero los bloques de instancia seguido del constructor, por último, se ejecuta la línea 12.

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;

La variable de la opción A lleva mayúscula al inicio, haciéndola diferente a la variable que se usa más adelante, esto causaría un error. No se puede asignar un valor de tipo long a un int, tampoco un valor decimal. El _ no puede ir antes de un punto.

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.

En la línea tres se crea una variable de tipo float pero a su valor le hace falta indicar que es float poniendo la letra f al final, esto generara un error. En la línea 10 se intenta imprimir una variable que no está dentro del scope, esto también generara un error.