

The answers to the chapter review questions can be found in the Appendix.

1. Which of the following data types can be used in a switch expression? (Choose all that apply.)

- A. enum
- B. int
- C. Byte
- D. long
- E. String
- F. char
- G. var
- H. double

Long y Double no se utilizan porque esta diseñado para trabajar con int, los double/float pueden tener problema de precision debido a su representacion binaria.

//

2. What is the output of the following code snippet? (Choose all that apply.)

```
3: int temperature = 4;
4: long humidity = -temperature + temperature * 3;
5: if (temperature>=4)
6: if (humidity < 6) System.out.println("Too Low");
7: else System.out.println("Just Right");
8: else System.out.println("Too High");
```

- A. Too Low
- B. Just Right
- C. Too High
- D. A NullPointerException is thrown at runtime.
- E. The code will not compile because of line 7.
- F. The code will not compile because of line 8.

La humedad es 8, por lo que se imprime "Just Right" ya que humedad es mayor a 6;

//

3. Which of the following data types are permitted on the right side of a for-each expression? (Choose all that apply.)

- A. Double[][]
- B. Object
- C. Map
- D. List
- E. String
- F. char[]
- G. Exception
- H. Set

Para el For-each: Pueden ser arrays, cualquier collection que implemente itarable.

//

4. What is the output of calling printReptile(6)?

```
void printReptile(int category) {  
    var type = switch(category) {  
        case 1,2 -> "Snake";  
        case 3,4 -> "Lizard";  
        case 5,6 -> "Turtle";  
        case 7,8 -> "Alligator";  
    };  
    System.out.print(type);  
}
```

- A. Snake
- B. Lizard
- C. Turtle
- D. Alligator
- E. TurtleAlligator
- F. None of the above

Como no se considera todos los valores, es error de compilacion para el nuevo switch expression.

//

5. What is the output of the following code snippet?

```
List<Integer> myFavoriteNumbers = new ArrayList<>();  
myFavoriteNumbers.add(10);  
myFavoriteNumbers.add(14);  
for (var a : myFavoriteNumbers) {  
    System.out.print(a + ", ");  
    break;  
}  
  
for (int b : myFavoriteNumbers) {  
    continue;  
    System.out.print(b + ", ");  
}  
  
for (Object c : myFavoriteNumbers)  
    System.out.print(c + ", ");
```

- A. It compiles and runs without issue but does not produce any output.
- B. 10, 14,
- C. 10, 10, 14,
- D. 10, 10, 14, 10, 14,
- E. Exactly one line of code does not compile.
- F. Exactly two lines of code do not compile.
- G. Three or more lines of code do not compile.
- H. The code contains an infinite loop and does not terminate.

El segundo for each contiene un continue, por esta razon no compilaria esa linea correctamente.

//

6. Which statements about decision structures are true? (Choose all that apply.)

- A. A for-each loop can be executed on any Collections Framework object.
- B. The body of a while loop is guaranteed to be executed at least once.
- C. The conditional expression of a for loop is evaluated before the first execution of the loop body.
- D. A switch expression that takes a String and assigns the result to a variable requires a default branch.
- E. The body of a do/while loop is guaranteed to be executed at least once.
- F. An if statement can have multiple corresponding else statements.

For each // solo los que utilicen iterable se puede implementar en un for each.

El do while al menos se ejecuta una vez
For; antes de entrar a un for se evalua para poder entrar.
Un switch expression requerimos el valor por default;

//

7. Assuming `weather` is a well-formed nonempty array, which code snippet, when inserted independently into the blank in the following code, prints all of the elements of `weather`? (Choose all that apply.)

```
private void print(int[] weather) {  
    for(_____) {  
        System.out.println(weather[i]);  
    }  
}
```

- A. `int i=weather.length; i>0; i--`
- B. `int i=0; i<=weather.length-1; ++i`
- C. `var w : weather`
- D. `int i=weather.length-1; i>=0; i--`
- E. `int i=0, int j=3; i<weather.length; ++i`
- F. `int i=0; ++i<10 && i<weather.length;`
- G. None of the above

A: es incorrecto porque no tiene acceso a `weather`, causa una excepcion.

B: es correcto compila en orden.

C: `weather` no esta definido, asi que es error de compilacion.

D: Es correcto hace un reverse en el array.

//

8. What is the output of calling `printType(11)`?

```
31: void printType(Object o) {  
32:     if(o instanceof Integer bat) {  
33:         System.out.print("int");  
34:     } else if(o instanceof Integer bat && bat < 10) {  
35:         System.out.print("small int");  
36:     } else if(o instanceof Long bat || bat <= 20) {  
37:         System.out.print("long");  
38:     } default {  
39:         System.out.print("unknown");  
40:     }  
41: }
```

No podemos utilizar un default, tendriamos que usar un else para que compile;

//

9. Which statements, when inserted independently into the following blank, will cause the code to print 2 at runtime? (Choose all that apply.)

```
int count = 0;
BUNNY: for(int row = 1; row <=3; row++)
    RABBIT: for(int col = 0; col <3 ; col++) {
        if((col + row) % 2 == 0)
            _____;
        count++;
    }
System.out.println(count);
```

- A. break BUNNY
- B. **break RABBIT**
- C. **continue BUNNY**
- D. continue RABBIT
- E. break
- F. **continue**
- G. None of the above, as the code contains a compiler error.

Break RABBIT, rompe el segundo ciclo for y se pasa al primero, imprime dos;

Break, crea lo mismo que breal RABBIT;

Continue no afecta el codigo

//

10. Given the following method, how many lines contain compilation errors? (Choose all that apply.)

```
10: private DayOfWeek getWeekDay(int day, final int thursday) {
11:     int otherDay = day;
12:     int Sunday = 0;
13:     switch(otherDay) {
14:         default:
15:             case 1: continue;
16:             case thursday: return DayOfWeek.THURSDAY;
17:             case 2,10: break;
18:             case Sunday: return DayOfWeek.SUNDAY;
19:             case DayOfWeek.MONDAY: return DayOfWeek.MONDAY;
20:     }
21:     return DayOfWeek.FRIDAY;
22: }
```

- A. None, the code compiles without issue.
- B. 1
- C. 2
- D. 3
- E. **4**
- F. 5
- G. 6
- H. The code compiles but may produce an error at runtime.

Switch:

15; No se puede agregar continue, se puede cambiar por un break o regresar un valor

16; error de compilacion porque no se sabe como se comportara al tiempo de compilar, es una constante pero no esta asegurado en este proceso.

18: falta agregar final, no es constante en tiempo de compilacion.

19; error de compilacion, es un enum no un int.

//

11. What is the output of calling `printLocation(Animal.MAMMAL)`?

```
10: class Zoo {  
11:     enum Animal {BIRD, FISH, MAMMAL}  
12:     void printLocation(Animal a) {  
13:         long type = switch(a) {  
14:             case BIRD -> 1;  
15:             case FISH -> 2;  
16:             case MAMMAL -> 3;  
17:             default -> 4;  
18:         };  
19:         System.out.print(type);  
20:     } }
```

- A. 3
- B. 4
- C. 34
- D. The code does not compile because of line 13.
- E. The code does not compile because of line 17.
- F. None of the above

Imprimiría 3 porque el switch case apunta que es 3 cuando se manda a llamar a MAMMAL,
El default no es necesario pero es buena práctica;

//

12. What is the result of the following code snippet?

```
3: int sing = 8, squawk = 2, notes = 0;  
4: while(sing > squawk) {  
5:     sing--;  
6:     squawk += 2;  
  
7:     notes += sing + squawk;  
8: }  
9: System.out.println(notes);
```

- A. 11
- B. 13
- C. 23
- D. 33
- E. 50
- F. The code will not compile because of line 7.

Siguiendo el orden de precedencia y el ciclo while para validacion el resultado de note = 23.

//

13. What is the output of the following code snippet?

```
2: boolean keepGoing = true;  
3: int result = 15, meters = 10;  
4: do {  
5:     meters--;  
6:     if(meters==8) keepGoing = false;  
7:     result -= 2;  
8: } while keepGoing;  
9: System.out.println(result);
```

- A. 7
- B. 9
- C. 10
- D. 11
- E. 15
- F. The code will not compile because of line 6.
- G. The code does not compile for a different reason.

8: la condicion en while keepGoing, no cuenta con parentesis

//

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

```
for(var penguin : new int[2])
    System.out.println(penguin);
var ostrich = new Character[3];
for(var emu : ostrich)
    System.out.println(emu);
List<Integer> parrots = new ArrayList<Integer>();
for(var macaw : parrots)
    System.out.println(macaw);
```

- A. The data type of penguin is Integer.
- B. The data type of penguin is int.
- C. The data type of emu is undefined.
- D. The data type of emu is Character.
- E. The data type of macaw is List.
- F. The data type of macaw is Integer.
- G. None of the above, as the code does not compile.

Se puede poner de que tipo de dato es por eso pinguin es int primitivo

Emu es Character porque es un arreglo de objetos

Macaw es tipo Integer

//

15. What is the result of the following code snippet?

```
final char a = 'A', e = 'E';
char grade = 'B';
switch (grade) {
    default:
    case a:
    case 'B': 'C': System.out.print("great ");
    case 'D': System.out.print("good "); break;
    case e:
    case 'F': System.out.print("not good ");
}
```

- A. great
- B. great good
- C. good
- D. not good
- E. The code does not compile because the data type of one or more case statements does not match the data type of the switch variable.
- F. None of the above

El codigo no compila porque

```
case 'B': 'C': System.out.print("great ");
```

Esta mal definido, se debe agregar un case o en su defecto una coma

//

16. Given the following array, which code snippets print the elements in reverse order from how they are declared? (Choose all that apply.)

```
char[] wolf = {'W', 'e', 'b', 'b', 'y'};
```

A.

```
int q = wolf.length;
for( ; ; ) {
    System.out.print(wolf[--q]);
    if(q==0) break;
}
```

B.

```
for(int m=wolf.length-1; m>=0; --m)
    System.out.print(wolf[m]);
```

C.
for(int z=0; z<wolf.length; z++)
 System.out.print(wolf[wolf.length-z]);

D.
int x = wolf.length-1;
for(int j=0; x>=0 && j==0; x--)
 System.out.print(wolf[x]);

E.
final int r = wolf.length;
for(int w = r-1; r>-1; w = r-1)
 System.out.print(wolf[w]);

F.
for(int i=wolf.length; i>0; --i)
 System.out.print(wolf[i]);

G. None of the above

A: Imprime a lo inverso y no tiene errores el código, funciona.

B: El ciclo impone de forma descendente, por lo que es correcto.

C: Empieza por imprimir el índice 5, nuestro índice es de 4, error marca una excepción.

D: El código imprime en forma descendente, aunque esta de más j.

E: Error, es un bucle infinito, siempre r será mayor a 5.

F: El índice es mayor, arroja una excepción.

//

17. What distinct numbers are printed when the following method is executed? (Choose all that apply.)

```
private void countAttendees() {  
    int participants = 4, animals = 2, performers = -1;  
    while((participants = participants+1) < 10) {}  
    do {} while (animals++ <= 1);  
    for( ; performers<2; performers+=2) {}  
  
    System.out.println(participants);  
    System.out.println(animals);  
    System.out.println(performers);  
}
```

- A. 6
B. 3
C. 4
D. 5
E. 10
F. 9
G. The code does not compile.
H. None of the above

El primer while llega a 10 porque se suma el +1 // participants= 10

Animals comienza con 2, al ser un do whiel al menos se evalua una vez // animals = 3

//

18. Which statements about pattern matching and flow scoping are correct? (Choose all that apply.)

- A. Pattern matching with an if statement is implemented using the instanceof operator.
B. Pattern matching with an if statement is implemented using the instanceon operator.
C. Pattern matching with an if statement is implemented using the instanceof operator.
D. The pattern variable cannot be accessed after the if statement in which it is declared.
E. Flow scoping means a pattern variable is only accessible if the compiler can discern its type.
F. Pattern matching can be used to declare a variable with an else statement.

Pattern matching; Es implementado con el operador instanceof;

El flow scoping asegura que solo usemos las variables del pattern cuando el compilador es seguro de su tipo, evitando errores de ejecución.

//

19. What is the output of the following code snippet?

```
2: double iguana = 0;
3: do {
4:     int snake = 1;
5:     System.out.print(snake++ + " ");
6:     iguana--;
7: } while (snake <= 5);
8: System.out.println(iguana);
```

- A. 1 2 3 4 -4.0
- B. 1 2 3 4 -5.0
- C. 1 2 3 4 5 -4.0
- D. 0 1 2 3 4 5 -5.0
- E. The code does not compile.
- F. The code compiles but produces an infinite loop at runtime.
- G. None of the above

La variable snake es local y se encuentra dentro del bloque do {}, se manda a llamar en While esto no es posible

//

20. Which statements, when inserted into the following blanks, allow the code to compile and run without entering an infinite loop? (Choose all that apply.)

```
4: int height = 1;
5: L1: while(height++ <10) {
6:     long humidity = 12;
7:     L2: do {
8:         if(humidity-- % 12 == 0) _____;
9:         int temperature = 30;
10:        L3: for( ; ; ) {
11:            temperature++;
12:            if(temperature>50) _____;
13:        }
14:    } while (humidity > 4);
15: }
```

- A. break L2 on line 8; continue L2 on line 12
- B. continue on line 8; continue on line 12
- C. break L3 on line 8; break L1 on line 12
- D. continue L2 on line 8; continue L3 on line 12
- E. continue L2 on line 8; continue L2 on line 12
- F. None of the above, as the code contains a compiler error

bucle más interno es un bucle infinito por lo tanto se debe evitar entrar en el bucle

A: La linea 8 tiene un break, hace que el segundo bucle interno termine cada vez que se ingresa, evitando por completo que se ejecute el bucle infinito.
E: Al agregar un continue hace que se salga del bucle infinito

//

- 21.** A minimum of how many lines need to be corrected before the following method will compile?

```
21: void findZookeeper(Long id) {  
22:     System.out.print(switch(id) {  
23:         case 10 -> {"Jane"}  
24:         case 20 -> {yield "Lisa";};  
25:         case 30 -> "Kelly";  
26:         case 30 -> "Sarah";  
27:         default -> "Unassigned";  
28:     });  
29: }
```

- A. Zero
- B. One
- C. Two
- D. Three
- E. Four
- F. Five

22: No compila porque long no es compatible con switch.

23: falta declaracion "yield"

25, 26: tienen el mismo numero de case "30"

//

- 22.** What is the output of the following code snippet? (Choose all that apply.)

```
2: var tailFeathers = 3;  
3: final var one = 1;  
4: switch (tailFeathers) {  
5:     case one: System.out.print(3 + " ");  
6:     default: case 3: System.out.print(5 + " ");  
7: }  
8: while (tailFeathers > 1) {  
9:     System.out.print(--tailFeathers + " "); }
```

- A. 3
- B. 5 1
- C. 5 2
- D. 3 5 1
- E. 5 2 1
- F. The code will not compile because of lines 3–5.
- G. The code will not compile because of line 6.

//

- 23.** What is the output of the following code snippet?

```
15: int penguin = 50, turtle = 75;  
16: boolean older = penguin >= turtle;  
17: if (older = true) System.out.println("Success");  
18: else System.out.println("Failure");  
19: else if(penguin != 50) System.out.println("Other");
```

- A. Success
- B. Failure
- C. Other
- D. The code will not compile because of line 17.
- E. The code compiles but throws an exception at runtime.
- F. None of the above

19: empieza con else if, cuando deberia ser if else.

//

- 24.** Which of the following are possible data types for `friends` that would allow the code to compile? (Choose all that apply.)

```
for(var friend in friends) {  
    System.out.println(friend);  
}
```

- A. Set
- B. Map
- C. String
- D. int[]
- E. Collection
- F. StringBuilder
- G. None of the above

La palabra `in` no existe en java, no es un bucle valido.

//

- 25.** What is the output of the following code snippet?

```
6: String instrument = "violin";  
7: final String CELLO = "cello";  
8: String viola = "viola";  
9: int p = -1;  
10: switch(instrument) {  
11:     case "bass" : break;  
12:     case CELLO : p++;  
13:     default: p++;  
14:     case "VIOLIN": p++;  
15:     case "viola" : ++p; break;  
16: }  
17: System.out.print(p);
```

- A. -1
- B. 0
- C. 1
- D. 2
- E. 3
- F. The code does not compile.

//

26. What is the output of the following code snippet? (Choose all that apply.)

```
9: int w = 0, r = 1;
10: String name = "";
11: while(w < 2) {
12:     name += "A";
13:     do {
14:         name += "B";
15:         if(name.length()>0) name += "C";
16:         else break;
17:     } while (r <=1);
18:     r++; w++; }
19: System.out.println(name);
```

- A.** ABC
- B.** ABCABC
- C.** ABCABCABC
- D.** Line 15 contains a compilation error.
- E.** Line 18 contains a compilation error.
- F.** The code compiles but never terminates at runtime.
- G.** The code compiles but throws a `NullPointerException` at runtime.

//

27. What is printed by the following code snippet?

```
23: byte amphibian = 1;
24: String name = "Frog";
25: String color = switch(amphibian) {
26:     case 1 -> { yield "Red"; }
27:     case 2 -> { if(name.equals("Frog")) yield "Green"; }
28:     case 3 -> { yield "Purple"; }
29:     default -> throw new RuntimeException();
30: };
31: System.out.print(color);
```

- A.** Red
- B.** Green
- C.** Purple
- D.** RedPurple
- E.** An exception is thrown at runtime.
- F.** The code does not compile.

//

28. What is the output of calling `getFish("goldie")`?

```
40: void getFish(Object fish) {  
41:     if (!(fish instanceof String guppy))  
42:         System.out.print("Eat!");  
43:     else if (!(fish instanceof String guppy)) {  
44:         throw new RuntimeException();  
45:     }  
46:     System.out.print("Swim!");  
47: }
```

- A.** Eat!
- B.** Swim!
- C.** Eat! followed by an exception.
- D.** Eat!Swim!
- E.** An exception is printed.
- F.** None of the above

//

29. What is the result of the following code?

```
1: public class PrintIntegers {  
2:     public static void main(String[] args) {  
3:         int y = -2;  
4:         do System.out.print(++y + " ");  
5:         while(y <= 5);  
6:     } }
```

- A.** -2 -1 0 1 2 3 4 5
- B.** -2 -1 0 1 2 3 4
- C.** -1 0 1 2 3 4 5 6
- D.** -1 0 1 2 3 4 5
- E.** The code will not compile because of line 5.
- F.** The code contains an infinite loop and does not terminate.

//