

02 – Operators and Statements - Exercises

1. Understanding Java Operators

1.1.Cuál de los siguientes operadores tiene mayor precedencia para cada uno de los casos (a o b):

- (a) * (b) &&
- (a) ! (b) =
- (a) n++ (b) %
- (a) > (b) +=

2. Working with Binary Arithmetic Operators

2.1. Cuál es el resultado de las siguientes operaciones:

- `int x = 6 * 8 + 4 / 3 - 20;`
- `int x = 6 * ((8 + 4) / 2 - 20 % 3);`

2.2. Cuál es el tipo de dato y el resultado para cada uno de los casos siguientes:

- La operación es `x + y`:

```
int x = 20
long y = 5
```

- La operación es `x * y / z`:

```
short x = 30;
float y = 10;
double z = 3
```

3. Working with Unary Operators

3.1. Cuál es la salida de las siguientes líneas (¿compila o no?):

- `boolean x = false;`
`System.out.println(!(!x));`
- `int x = 5;`
`System.out.println(!x);`
- `boolean y = -true;`
- `int a = 2;`
`int b = ++a * a-- / a-- + --a;`
`System.out.println("a es " + a + ", b es " + b);`

4. Using Additional Binary Operators

4.1. Cuál de los siguientes ejemplos compila:

- `int z = 6f;`
- `int x = 3.0;`
- `long t = 192301398193810323;`
- `int x = (int)1.0`
- `short x = 10;`
`short y = 3;`
`short z = x * y;`
- `long x = 5;`
`long y = (x=3);`

4.2. Cuál de los siguientes ejemplos nos lanzará una excepción de `NullPointerException`:

```
- Integer x = null;
  if(x != null && x.getValue() < 9) {
    // Do something
  }
- Integer x = null;
  if(x != null & x.getValue() < 9) {
    // Do something
  }
```

4.3. Cuál es la salida del siguiente código:

```
int a = 10;
boolean b = (a >= 10) || (++a <= 11);
System.out.println(a);
```

4.4. Cuál de los siguientes ejemplos compila:

```
- boolean x = true == 2;
- boolean z = 2 == "Lunes";
- boolean y = 4 == 5;
```

4.5. Cuál es la salida de los siguientes ejemplos:

```
- boolean b = false;
  boolean a = (b = true);
  System.out.println(x);
- Perro x = new Perro("Fido");
  Perro y = new Perro("Boby");
  Perro z = x;
  System.out.println(x == y);
  System.out.println(x == z);
```

5. Understanding Java Statements

5.1. Cuál es el resultado del siguiente código:

```
int hourOfDay = 12;
int morningGreetingCount = 1;
if(hourOfDay < 11)
  System.out.println("Good Morning");
  morningGreetingCount++;
System.out.println("Morning: " + morningGreetingCount);
```

5.2. Escriba un código que valida la variable hora24 (p. e. 15 que sería 3pm en formato 24 horas) y que muestre en consola "Buenos días", "Buenas tardes" o "Buenas noches" siendo de día de 6am a 12m, de tarde de 1pm a 6pm y de noche de 7pm a 5am.

5.3. Transforme el siguiente código usando un operador ternario:

```
int b = 5;
int a;
if(b > 5) {
  a = 4 * b;
} else {
  a = 5 * b;
}
```

5.4. Cuáles son los tipos admitidos para las sentencias switch. Recuerde que son 7.

5.5. Cuál es la salida del siguiente código:

```
int diaDeLaSemana = 5;
switch(diaDeLaSemana) {
    case 2:
        System.out.println("Martes");
    default:
        System.out.println("Otro dia");
    case 5:
        System.out.println("Viernes");
        break;
}
```

5.6. Cuál es la salida del siguiente código:

```
int contadorX = 0;
int contadorY = 3;
while (++contadorX > 0 && contadorY-- > 0) {
    contadorX--;
    contadorY--;
}
System.out.println("contadorX:" + contadorX + ",contadorY:" + contadorY);
```

5.7. Cuál es el resultado del siguiente código:

```
boolean seguir = true;
int result = 15;
do {
    if(result==13) seguir = false;
    result -= 2;
} while(seguir);
System.out.println(result);
```

5.8. Cuál es la salida del siguiente código:

```
String[] names = new String[3];
names[0] = "Lisa";
names[1] = "Kevin";
names[2] = "Roger";
for(int x = 2; x < 3; x++) {
    System.out.print(name + " ");
}
System.out.print(x);
```

6. Understanding Advanced Flow Control

6.1. Cuál es el resultado del siguiente código:

```
int a = 10;
while(a>0) {
    do {
        a -= 3
    } while (a>5);
    a--;
    System.out.print(a+" ");
}
```

6.2. Indique la salida del siguiente código:

```
int[][] myComplexArray = {{0, 1, 2, 3}, {4, 5, 6, 7}, {8, 9, 10, 11}};
OUTER_LOOP:
for (int[] mySimpleArray : myComplexArray) {
```

```
int i = 0;
INNER_LOOP:
while (i < mySimpleArray.length) {
    i++;
    System.out.print(mySimpleArray[i] + " ");
    if (i == 2) {
        continue OUTER_LOOP;
    } else {
        break INNER_LOOP;
    }
}
```