1.- Which of the following Java operators can be used with boolean variables? (Choose all that apply.)

Opciones: 1. ==
2. +
3. –
4. !
5. %
6. <=
7. Cast with (boolean)

R= == es el operador de comparacion el cual se usa con primitivas ,valores booleanos y referencias a objetos
! es el operador logico y unicamentes se para comparar que son diferentes se caeste el valor de los booleanos para saber si regresa un true o un false

2.- What data type (or types) will allow the following code snippet to compile? (Choose all that apply.)

```
byte apples = 5;
short oranges = 10;
_______ bananas = apples + oranges;
```

Opciones: 1. int
2. long
3. boolean
4. double
5. short
6. Byte.

R= se promociona automáticamente a int, por lo que funcionarán int y los tipos de datos que se pueden promocionar automáticamente desde int a los demas tipos.

3.- What change, when applied independently, would allow the following code snippet to compile? (Choose all that apply.)

```
long ear = 10;
int hearing = 2 * ear;
```

Opciones: 1. No change: it compiles as is.

2. Cast ear on line 4 to int.
3. Change the data type of ear on line 3 to short.
4. Cast 2 * ear on line 4 to int.
5. Change the data type of hearing on line 4 to short.
6. Change the data type of hearing on line 4 to long.

R= las 2 alternativas son correctas por que puedes castear la variable a int o realizar la multiplicacion y castear el resultado a int para no tener incompatibilidad de tipos.

4.- What is the output of the following program?

```
1: public class CandyCounter {
2:       static long addCandy(double fruit, float vegetables) {
3:       return (int)fruit+vegetables;
4: }
5:
6: public static void main(String[] args) {
7:       System.out.print(addCandy(1.4, 2.4f) + "- ");
8:       System.out.print(addCandy(1.9, (float)4) + "-");
9:       System.out.print(addCandy((long)(int) (short)2, (float)4)); } }
```

Opciones:

1. 4-6-6.0
2. 3-5-6
3. 3-6-6

4. 4-5-6
5. The code does not compile because of line 9.
6. None of the above

R= no compila por que al final se realiza un casteo en long pero el metodo regresa un entero por lo tanto, no puedes meter algo grando en un recipiente pequeño.

5. What are the unique outputs of the following code snippet? (Choose all that apply.)

```
int a = 2, b = 4, c = 2;
System.out.println(a > 2 ? --c : b++);
System.out.println(b = (a!=c ? a : b++));
System.out.println(a > b ? b < c ? b : 2 : 1);
```

Opciones: 1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. The code does not compile

R= en el primer sistem la variable entra como un 4 ,en el segundo la b llega con el valor aumentado 5 despues de usarlo y al final son if que comparan como da un false devuelve 1 .

6. Given the following code snippet, what is the value of the variables after it is executed? (Choose all that apply.)

```
int ticketsTaken = 1;
int ticketsSold = 3;
ticketsSold += 1 + ticketsTaken++;
ticketsTaken *= 2;
ticketsSold += (long)1;
```

Opciones: 1. ticketsSold is 8
2. ticketsTaken is 2
3. ticketsSold is 6
4. ticketsTaken is 6
5. ticketsSold is 7
6. ticketsTaken is 4
7. The code does not compile.

R= por gerarquia se deben primero resolver los de derecha y luego los de izquierda, para simplificarlo a operaciones aritmeticas

7. 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");
```

Opciones: 1. Too Low
2. Just Right
3. Too High

R= es un if anidado que checa la temperatura y humedad el cual compara la humedad con ses y como es falso se va al else e imprime un just right

4. A NullPointerException is thrown at runtime.
5. The code will not compile because of line 7.
6. The code will not compile because of line 8.

8. 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);
```

Opciones: 1. break BUNNY
2. break RABBIT
3. continue BUNNY
4. continue RABBIT
5. break
6. continue
7. None of the above, as the code contains a compiler error

R= esa instrucion nos permite decirle al programa que continue dirigiendola a la etiqueta bunny.

9. 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);
```

Opciones: 1. 7
2. 9
3. 10
4. 11
5. 15
6. The code will not compile because of line 6.
7. The code does not compile for a different reason.

R= el codigo no compila el loop while no esta bien definido.

10. 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);
```

Opciones: 1. ABC
2. ABCABC
3. ABCABCABC
4. Line 15 contains a compilation error.
5. Line 18 contains a compilation error.
6. The code compiles but never terminates at runtime.
7. The code compiles but throws a NullPointerException at runtime.

R= el codigo compila de manera exitosa pero en ejecucion el while nunca se sale

11. What is output by the following code? (Choose all that apply.)

```
1: public class Fish {
2:     public static void main(String[] args) {
3:         int numFish = 4;
4:         String fishType = "tuna";
5:         String anotherFish = numFish + 1;
6:         System.out.println(anotherFish + " " + fishType);
7:         System.out.println(numFish + " " + 1);
8: } }
```

Opciones: 1. 4 1
2. 5
3. 5 tuna
4. 5tuna
5. 51tuna
6. The code does not compile.

R= no puedes compilar por que no puedes realizar operaciones aritmeticas con strings

12. What is the result of the following code?

```
7: StringBuilder sb = new StringBuilder();
8: sb.append("aaa").insert(1, "bb").insert(4, "ccc");
9: System.out.println(sb);
```

Opciones: 1. abbaaccc
2. abbaccca
3. bbaaaccc
4. bbaaccca
5. An empty line
6. The code does not compile.

R= es el insert funcina como un posicionamiento para poder insertar el string que queremos agregar

13. What is the result of the following code?

```
12: int count = 0;
13: String s1 = "java";
14: String s2 = "java";
15: StringBuilder s3 = new StringBuilder("java");
16: if (s1 == s2) count++;
17: if (s1.equals(s2)) count++;
18: if (s1 == s3) count++;
19: if (s1.equals(s3)) count++;
20: System.out.println(count);
```

Opciones: 1. 0
2. 1
3. 2
4. 3
5. 4
6. An exception is thrown.
7. The code does not compile.

R= no compila por compara objetos y los objetos son diferentes, el string del pool de string es un objeto diferente al de streambuilder.

14. What is the result of the following code?

```
public class Lion {
	public void roar(String roar1, StringBuilder roar2) {
		roar1.concat("!!!");
		roar2.append("!!!");
	}
	public static void main(String[] args) {
		String roar1 = "roar";
		StringBuilder roar2 = new StringBuilder("roar");
		new Lion().roar(roar1, roar2);
		System.out.println(roar1 + " " + roar2);
} }
```

Opciones: 1. roar roar
2. roar roar!!!
3. roar!!! roar
4. roar!!! roar!!!
5. An exception is thrown.
6. The code does not compile.

R= roar uno es un string del pull de strings entonces cuando se le concatenan los signos de esclamacion se debe cambiar la referencia al nuevo objeto que se crea pero nunca se le referencia debio a que los strings del pull de strings son inmutables a diferencia de los que son creados por string builders que si son mutables entonces al accerle el append() el stringbuilder muta y agarra la forma de roar!!!

15. Which of the following can replace line 4 to print "avaJ"? (Choose all that apply.)

```
3: var puzzle = new StringBuilder("Java");
4: // INSERT CODE HERE
5: System.out.println(puzzle);
```

Opciones: 1. puzzle.reverse();
2. puzzle.append("vaJ$").substring(0, 4);
3. puzzle.append("vaJ$").delete(0, 3).deleteCharAt(puzzle.length() - 1);
4. puzzle.append("vaJ$").delete(0, 3).deleteCharAt(puzzle.length());
5. None of the above

R= tiene un metodo que te regresa el string builder de reversa en la instancia puzzle.