Examen Semana 1 - Fabiola Gómez Montiel

Capítulo 3

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

- [x] ~~==~~
- [ ] +
- [ ] -
- [x] ~~!~~
- [ ] %
- [ ] <=
- [x] ~~Cast with (boolean)~~

> 👾 Las opciones no seleccionadas no pueden usarse con datos de tipo booleano.

---

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;
```

- [x] ~~int~~
- [x] ~~long~~
- [ ] boolean

- [x] ~~double~~
- [ ] short
- [ ] byte

La suma de apples y oranges se convierte en un int y por eso los únicos tipos de datos que pueden usarse son int, long y double. Para los demás datos habría que hacer un casteo.

---

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

```
3: long ear = 10;
4: int hearing = 2 * ear;
```

- [ ] No change; it compiles as is.
- [x] ~~Cast ear on line 4 to int.~~
- [x] ~~Change the data type of ear on line 3 to short.~~
- [ ] Cast 2 * ear on line 4 to int.
- [ ] Change the data type of hearing on line 4 to short.
- [x] ~~Change the data type of hearing on line 4 to long.~~

Las primeras opciones nos ayudan a convertir el tipo de dato de ear en un int.
La última opción cambia el tipo de dato de hearing para que pueda contener el long.

---

6. 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)); } }
```

- [ ] 4-6-6.0
- [ ] 3-5-6
- [ ] 3-6-6
- [ ] 4-5-6
- [ ] The code does not compile because of line 9.
- [x] ~~None of the above~~

El casteo en la línea 3 solamente afecta a fruit, así que el resultado final de la operación es un float que no puede devolverse ya que el método addCandy debe retornar un valor de tipo long.

---

9. 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);
```

- [x] ~~1~~
- [ ] 2

- [ ] 3
- [x] ~~4~~
- [x] ~~5~~
- [ ] 6
- [ ] The code does not compile.

---

17. 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;
```

- [ ] ticketsSold is 8
- [ ] ticketsTaken is 2
- [x] ~~ticketsSold is 6~~
- [ ] ticketsTaken is 6
- [ ] ticketsSold is 7
- [x] ~~ticketsTaken is 4~~
- [ ] The code does not compile.

La lógica que se sigue es la siguiente:

1. Al valor de ticketsSold se le suma 1 y además se le agrega el valor de ticketsTaken, dando como resultado 5.
2. Al valor de ticketsTaken se le aplicó un incremento en la línea 3, por lo que su valor nuevo es de 2 y debe multiplicarse por 2, dando como resultado final 4.
3. Al valor (5) de ticketsSold nuevamente se le suma 1, así que su valor final es 6.

Capítulo 4

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");
```

- [ ] Too Low
- [x] ~~Just Right~~
- [ ] Too High
- [ ] A NullPointerException is thrown at runtime.
- [ ] The code will not compile because of line 7.
- [ ] The code will not compile because of line 8.

👾 Se ejecuta el if de la línea 5 ya que temperature = 4, pero debido a que humidity = 8 se aplica el resultado del else de la línea 7 que lanza en consola “Just Right”.

---

6. 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);
```

- [ ] break BUNNY
- [x] ~~break RABBIT~~
- [x] ~~continue BUNNY~~
- [ ] continue RABBIT
- [x] ~~break~~
- [ ] continue
- [ ] None of the above, as the code contains a compiler error

---

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);
```

- [ ] 7
- [ ] 9
- [ ] 10
- [ ] 11
- [ ] 15
- [ ] The code will not compile because of line 6.
- [x] ~~The code does not compile for a different reason.~~

👾 Le hacen falta paréntesis al código en la línea 8, por lo que no compila.

---

20. 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);
```

- [ ] ABC
- [ ] ABCABC
- [ ] ABCABCABC
- [ ] Line 15 contains a compilation error.
- [ ] Line 18 contains a compilation error.
- [x] ~~The code compiles but never terminates at runtime.~~

- [ ] The code compiles but throws a NullPointerException at runtime.

👾 Los incrementos de los valores w y r se encuentran fuera del do - while, por lo que se crea un loop infinito.

Capítulo 5

1. 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: } }
```

- [ ] 4 1
- [ ] 5
- [ ] 5 tuna
- [ ] 5tuna
- [ ] 51tuna
- [x] ~~The code does not compile.~~

👾 No compila porque en la línea 5 no se puede contener el valor de un int dentro del String.

---

4. 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);
```

- [ ] abbaaccc
- [x] ~~abbaccca~~
- [ ] bbaaaccc
- [ ] bbaaccca
- [ ] An empty line
- [ ] The code does not compile.

El método que se sigue es el siguiente:

1. append(”aaa”) = **aaa**
2. insert(1,”bb”) = a**bb**aa (se insertó “bb” en el índice 1)
3. insert(4, “ccc”) = abba**ccc**a (se insertó “ccc” en el índice 4)

---

5. 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);
```

- [ ] 0
- [ ] 1

- [ ] 2
- [ ] 3
- [ ] 4
- [ ] An exception is thrown.
- [x] ~~The code does not compile.~~

> 👾 No se puede comparar el String con el StringBuilder usando == como se tiene en la línea 18, por lo que el código no compila.

---

6. 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);
} }
```

- [ ] roar roar
- [x] ~~roar roar!!!~~
- [ ] roar!!! roar
- [ ] roar!!! roar!!!
- [ ] An exception is thrown.
- [ ] The code does not compile.

El método concat devuelve un String nuevo, no cambia el original ya que String es inmutable.
El método append sí puede añadir caracteres a un StringBuilder ya que StringBuilder sí es mutable.

---

13. 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);
```

- [x] ~~puzzle.reverse();~~
- [ ] puzzle.append("vaJ$").substring(0, 4);
- [ ] puzzle.append("vaJ$").delete(0, 3).deleteCharAt(puzzle.length() - 1);
- [ ] puzzle.append("vaJ$").delete(0, 3).deleteCharAt(puzzle.length());
- [ ] None of the above

El método reverse puede usarse para revertir los caracteres del StringBuilder.