Java Academy Xideral 2022

Student: Moisés Vidal Hernández

Teacher: Miguel Angel Rugerio

Java Bootcamp Training

Test: Week 1

November 25 of 2022

Chapter 3

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

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

R= You can only use the == and ! operators because Booleans cannot apply decrement or increment or store numeric data.

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

1. byte apples = 5;
2. short oranges = 10;
3. _____ bananas = apples + oranges;

1. int
2. long
3. boolean
4. double
5. short
6. byte

R= You can use int, long and double types, you can also use short and byte types but you must add a cast before the operation like shown on the next picture.

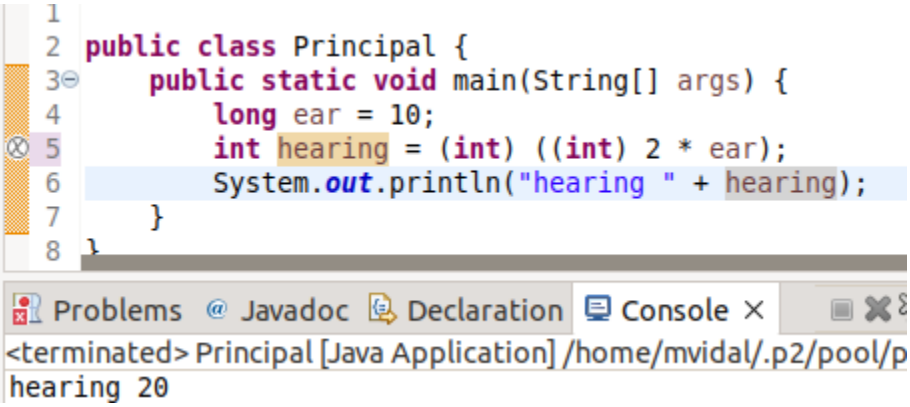
```
public static void main(String[] args) {
    byte apples = 5;
    short oranges = 10;
    byte bananas = (byte) (apples * oranges);
    System.out.println("Bananas: "+ bananas);
}
```

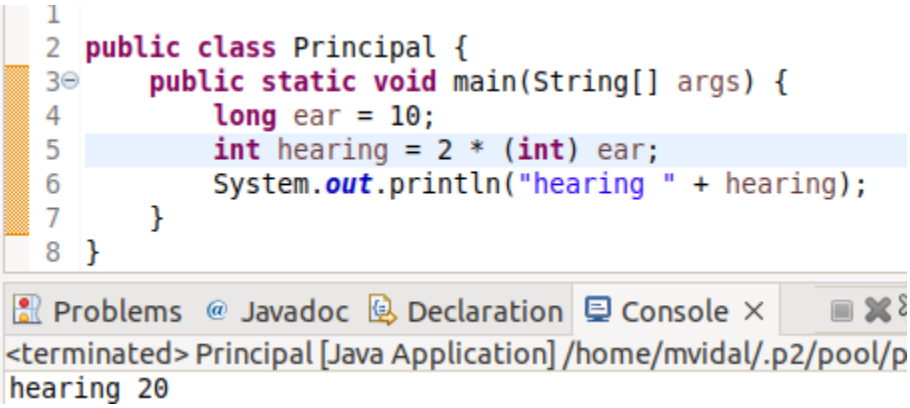
3.- What change, when applied independently, would allow the following code snippet to compile? (Choose all that apply.)

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

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= The correct answer is option 2 & 4, where we apply int line 4 just like shown in the next pictures.





This has a simple justification, because the type of data of result must be int.

We can also use option 3, because the number of bits of the short data type is less than that of the int data type.

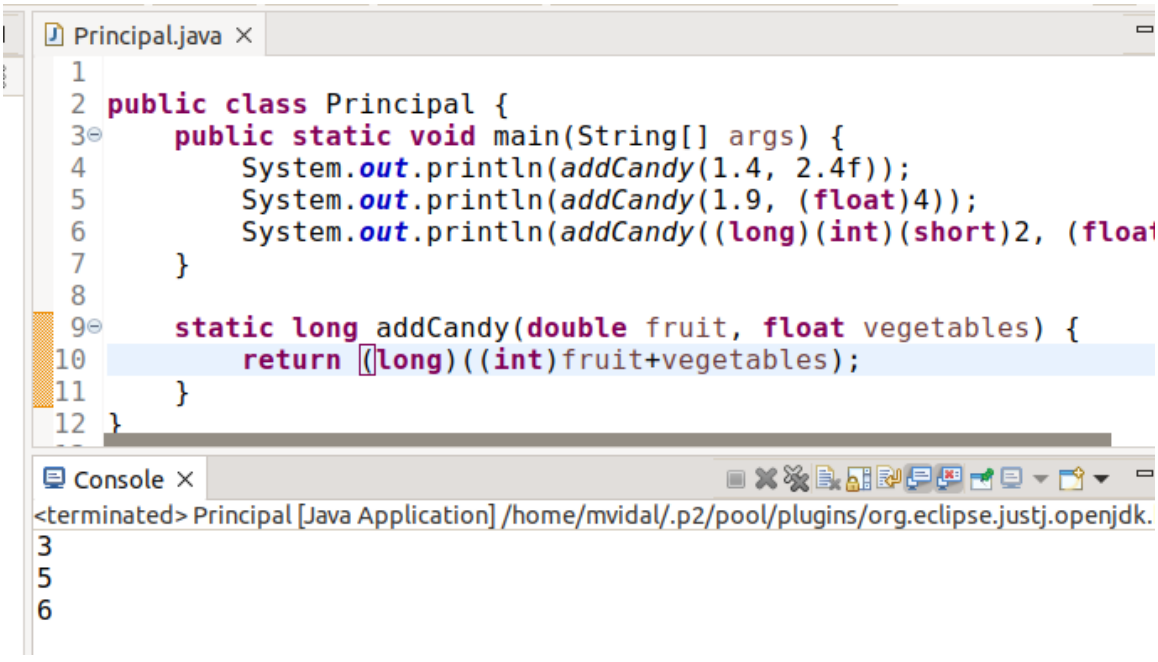
Finally we can use option 6, because the type data result of its operation is long.

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)); } }
```

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= The answer is option 6, because the program does not compile, since the addCandy method must return one value of type long and it is returning a value of type int.
The way to solve this is to add a cast of type long at the beginning of the return and have it take the entire operation and cast int. In this way.



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);
```

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

R= The outputs are,

- 4 because a is the same as 2, and the posincrement executes in the next call of his variable.
- 5 because a is the same as c, and the posincrement executes in the next line.
- 1 because the 1st. The condition is not true.

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;
```

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= The final values are,

- ticketsTaken is 4
- ticketsSold is 6

Chapter 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");
```

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

R= The output is "Just Right", because on line 4 the multiplication is 1st. that is solved, therefore its result is 12. After this we solve the sum with -4, its result is 8, in this way it will enter the first if and then it will enter the internal else of the 2nd. condition.

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);
```

1. break BUNNY
2. break RABBIT
3. continue BUNNY
4. continue RABBIT5. break
6. continue
7. None of the above, as the code contains a compiler error

R= The statements that we should insert are break RABBIT & continue BUNNY, because these statements will be executed when row and col has a value 1.

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);
```

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= We could consider that the output is 11, but this program may not compile because there is a syntax error on line 9, since keepGoing should be an in parenthesis.

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);
```

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= The answer is option 6, since the variable r does not increment in do while loop. The solution should be like this.

```
2  public class Principal {
3      public static void main(String[] args) {
4          int w = 0, r = 1;
5          String name = "";
6          while (w < 2) {
7              name += "A";
8              do {
9                  name += "B";
10                 if (name.length() > 0)
11                     name += "C";
12                 else
13                     break;
14                 r++;
15             } while (r <= 1);
16             w++;
17         }
18         System.out.println(name);
19     }
20 }
21
```

Console ×

<terminated> Principal [Java Application] /home/mvidal/.p2/pool/plugins/o

```
ABCABC
```

Chapter 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: } }
```

1. 4 1
2. 53. 5 tuna
4. 5tuna
5. 51tuna
6. The code does not compile.

R= The code does not compile, since in line 5 we want to add a string with an int, but this is not possible.
To resolve this problem we need to declare anotherFish as int, like this way.

```
2  public class Principal {
3      public static void main(String[] args) {
4          int numFish = 4;
5          String fishType = "tuna";
6          int anotherFish = numFish + 1;
7          System.out.println(anotherFish + " " + fishType);
8          System.out.println(numFish + " " + 1);
9      }
10 }
11
```

```
Console ×
<terminated> Principal [Java Application] /home/mvidal/.p2/pool/plugins/org.eclipse.justj.op
5 tuna
4 1
```

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);
```

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

R= The output is: abbaccca:
The solution is this.

- When executes sb.append(“aaa”) we obtain “aaa”
- Whe executes insert(1, “bb”) we obtain “abbaa”
- Finally when executes insert(4, “ccc”) we obtain “abbaccca”

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);
```

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

R= The code does not compile because we cannot compare one String with one StringBuilder, if we wanna compare we declare s3 as String.

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);
  } }
```

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

R= The result is the option 2, because the roar1 is not be assigned the new string with concat and the way of see is only direct inner of method roar()
If we wanna see, we need to do this.

```
2  public class Principal {
3      public static void main(String[] args) {
4          String roar1 = "roar";
5          StringBuilder roar2 = new StringBuilder("roar");
6          new Principal().roar(roar1, roar2);
7      }
8
9      public void roar(String roar1, StringBuilder roar2) {
10         roar1 = roar1.concat("!!!");
11         roar2.append("!!!");
12         System.out.println(roar1 + " " + roar2);
13     }
14 }
15
```

```
Console ×
<terminated> Principal [Java Application] /home/mvidal/.p2/pool/plugins/org.eclipse.justj.openjd
roar!!! roar!!!
```

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);
```

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= We can use option 1 and 3.

- With option 1, we just take the floor and reverse their order from end to start.
- With option 3,
    - When executes puzzle.append(“vaJ$”), we obtain “JavavaJ$”.
    - When executes delta(0,3), we obtain “avaJ$”.
    - Finally when executes deleteCharAt(Puzzle.lenght()-1), we obtain “avaJ”.