

# Quiz Preparation for Week 2

---

## Concepts

- More functions
  - function overload
  - void return
- Strings
  - concatenation
  - dot notation
- Algorithms and Strings
  - type casting
  - immutable types
  - string parsing
- null type

## Warm-ups problems

1. What does function overload mean?
2. For the following functions:

```
int foo(int a, int b) { /* something */ }
/*
...
*/
double foo(int c, int d) { /* something */ }
```

is the function overloading correct? If not, what's wrong?

3. How do you get the length of a String *theString*?
4. How do you concatenate String *foo* and String *bar* together, with a question mark added in between?
5. What does null mean in Java?
6. How do you check if an object is null?
7. What will the following codes do?

```
int i = 6;
i = 6.7;
System.out.println(i);
```

8. Give some examples of mutable types and immutable types.
9. How do you parse a String "I have\nan\napple" based on line break (character "\n")? What would you get?
10. Why do you want to check whether a variable/object is null at the start of a function?

*(Solutions in a separate file)*

## Combining it all together!

*(NOTICE: background story not necessary for the practice, you can jump right to To solve)*

You are a code bricklayer at lalve® doing some tedious data analysis job. Every day your boss GabeN electronically throws a pile of unprocessed csv data files at you, and if you don't finish compiling them by the end of the day, you know you are done. Well, job is easy, life is tough.

Today however, GabeN is particularly mean and he gives you a single txt file unprocessed to the point that it is like from the stone age. The file contains this week's game sales and is very important to Lord GabeN. You decide to load the file and it appears as a single String looks like this.

```
String sales = ""
Call Of Coding CXXV,6.67,3.2
Computer-Science: Good Overloading,3.44,-5.5
Float Guys,10.56,68.2
Data 2,0.0,0.0""";
```

You recognize that each line is a game and for each line, the three fields separated by commas are

*name, sales(million dollars), percentage change(%)*

## 1. Parsing the sales

- To solve: You need to parse the sales by line and store them in an array of String (don't worry about parsing each field within a line yet!). What would you do?

## 2. Check for errors

Dave somehow gives you a cleaner copy of this week's sales. He clearly wants you to check whether there exists errors on the reporting numbers. You dare not ask GabeN why he gives you the unprocessed file while there exists a cleaner one.

You decide to store the cleaner copy in the same format as yours with the follow code. For your convenience, Dave preserves the order of games so you won't mess up.

```
String[] refSales = getDavesCopy();
```

- To solve: Write a function *checkErrors* that takes two String arrays as parameters and compares both the sale number and the percentage change. If there is any error, report false, otherwise report true. (*hint: comparing two Strings str1 and str2, you should use str1.equals(str2)*)

### 3. Report to GabeN

- To solve: Write a function *report* to translate each sale to a sentence describing its status and print it out. The function should return void. The translated sentence should look like this:

[Game Name] total sale is \$[sale number] mil. Comparing to last week, it changes [percent change]%.