❖ **Quarantine**:

Participants get a small project with a one-week deadline. Each member will be assigned to a mentor(usually one of the lab members). Each project has a dedicated tech lead who determines project requirements. The tech lead will give feedback to participants, at the end of this phase.

❖ **Requirements**:

- ➢ Adding your project to GitHub/GitLab:
  - Create a remote, empty folder/repository on Github ("Lab projects").
  - Make sure that git tracks your project locally
  - Connect your local project folder to your empty folder/repository on Github.
  - Send the link to the technical lead via email.
- ➢ Coding Style:
  - https://google.github.io/styleguide/
- ➢ Do not share your ideas with other participants.
- ➢ Feel free to contact the technical manager if you have any questions.

❖ What the program is intended to do.

- ➢ Goals of the project.
- ➢ How to use the program?
- ➢ How well it works, which features do not work properly or need to be improved.
- ➢ An appendix containing a listing of the program, correctly formatted and with comments that explain what it is doing.
- ➢ All the files necessary to build the program and try it out.

❖ Assessment

- ➢ Here are the evaluation criteria that will be applied. The projects' objectives will be very different, hence the criteria will be given varying weights in various situations. The assessed work is code and its associated documentation. The participant should write the project so as to show off his/her technical skills to best advantage. The aim of the ***quarantine*** module is that the participant should have written and documented the working program with a degree of professionalism. Professionalism in this context refers to the

ability of the user interface, documentation, source code structure, resilience, and readability to support colleague usage or extension of the software in addition to correct operation.

- ➢ Features such as the following are expected :
  - definition of appropriate classes (C++, Python, Java)
  - use of the standard library features
  - use of data structures, algorithms, built-in objects and methods, etc.
  - use of OOAD

- ❖ Timetable
  - ➢ Seven days after the project's start date (the start is counted from the day after receiving the project)
- ❖ Coding standards
  - ➢ https://google.github.io/styleguide/

❖ Project description

Keyword analyzer

- **Type** : Console application, desktop application
- **Technical lead : Martun Davitavyan**

  (martun.davitavyan@picsart.academy)

- **Technologies** : C/C++, Python, Java, C#, JavaScript
- **Participant(s) : Grigori Verdyan (verdyan01@gmail.com)**

Write a program that analyzes the C++ file and prints the number of keywords (use the following list C++ keywords - cppreference.com), functions, variables and

classes in it.

For functions the program should also print how many arguments each function takes. For example for the following code:

```
int main() {
   int x = 12;
```

```
    double y;
    cout << x;
}
```

the program should print the following:

functions : 1
Variables : 2
functions:
main : 0 arguments

The analysis should be kept in a text file.