# Arithmetic game

Create a small console-based game using Python in an **object oriented** approach (see criteria below for further information). In the game, the player should be presented with different basic calculations that they must perform as quickly as possible. The game runs until a wrong answer was given or until the time limit of 10 seconds per question was reached.

The **basic functionality** of your program should be:

- When running the program, the player is asked to provide two inputs:
  - Their **name**
  - The name of the **statsfile** where the highscores will be recorded
- After this, the questions begin. Each question is of the format $a \otimes b = x$ where $a$ and $b$ are integers between 1 and 100, $\otimes$ is an arithmetic operation in $\{+, -, *, //\}$ ($//$ is the integer division) and $x$ is the result of the calculation that the player must find. The arithmetic operation as well as the integers $a$ and $b$ should be chosen at random.
- If the player gave a correct answer, the next question is presented to them and their score is increased by 1. If the answer was wrong, the game ends (see below).
- For each question, the time it took the player to answer is tracked and stored in a list (for later evaluation). If, for any question, the player took more than 10 seconds to answer, the game ends as if the given answer had been wrong.
- At the **end of the game**, the player is presented with statistics about the game they just played, containing
  - The score (number of correct answers) that they achieved
  - The average time they took to answer each question (average time per question)
- The highscores of each player are tracked in the stats file that they specified at the beginning of the game. Inside this file, the highest score that each player (identified by their name) **ever** achieved is stored. Therefore, if, for example, the player named *Alice* is already in the stats file with a highscore of 15, plays the game and achieves a score of 20, the old highscore is replaced by the new one. On the other hand, if Alice had achieved a score lower than 15, nothing would change in the file. **For this, you can choose any approach you like** as long as it can differentiate between players and behaves as described.

As stated above, you must use and **object oriented** approach, which means that at least one key feature of the program must be realized using objects and classes.

You are **permitted** to use any online/offline resources you like, **except**

- Generative AIs like ChatGPT
- Communication with other people

## Grading

Including the following in your solution will give you the specified number of points (partial points are of course possible):

- The basic functionality described above is implemented [50/100 Pts.]
- Your class(es) use(s) getters and setters in a meaningful way [15/100 Pts.]
- Proper error handling (user input is checked and the program doesn't crash or end on illegal input/keyboard interrupt) [15/100 Pts.]
- You implemented at least 3 meaningful tests that test different features of your program [10/100 Pts.]
- The code is documented and well organized [10/100 Pts.]
- *(Bonus)* Alternatively to giving the player name and the statsfile name via input functions, the user is able to pass this information to the program using a command line interface implemented using **docopt** [10 bonus Pts.]
- *(Bonus)* There is an `AUTHORS`, a `LICENSE` and a `README` file present, each with meaningful content (the `README` should contain instructions on how to use the program [5 bonus Pts.]

## Example runs

This is an example run of the finished program (how it COULD look like):

```
Your name: Max
Stats file name: stats.txt

Question 1: What is 23 + 29?
> 52

Question 2: What is 100 - 50?
> 50

Question 3: What is 10 * 15?
> 1

Wrong answer!

Your stats:
Score: 3
Average time per question: 4.23 seconds

Congratulations! New Highscore! Stored score in stats.txt
```

Another example run where the time runs out:

```
Your name: Max
Stats file name: stats.txt

Question 1: What is 23 + 29?
> 52

```

```
Question 2: What is 100 - 50?
> 50

You took 14.3 seconds to answer, which is above the time limit of 10 seconds.

Your stats:
Score: 1
Average time per question: 6.7 seconds
```

## Example statsfile

This is an example of how the statsfile could look like:

```
{"Alice": 15, "Bob": 4, "Charlie": 34}
```