CS 150 Lecture 26 Exercises

Complete each of the exercises below and upload to Canvas before the deadline.

Exercise 1 - Preserving Invariants

The working constructor must make sure that the **Date** it creates is valid. Since users are passing integers, it is **possible to get an invalid date**. What **should** happen if the user tries to create **Date d{35, Month:mar, 2020};**?

You should **throw an exception**. Here's how.

If **mktime()** returns **time\_t(-1)** or if the values stored in the **struct** **tm** are different **after** calling **mktime()**, then the **date is invalid**. You can throw an instance of the **BadDate** nested class to signal that. [Modify the working constructor](https://codecheck.it/files/1911120204a1mnvv57gycr7euz4w9ng423j).

| *Screenshot of the tested code and the passing score.* |
| --- |

Exercise 2 - Accessor Members, const & this

Complete the [**three accessor members**](https://codecheck.it/files/19111201072swpmwy98f0xlitkmoab77pqi), **day()**, **month()** and **year()**, so that the client program, which prints the date in the ISO 8601 international forma works correctly.

| *Screenshot of the tested code and the passing score.* |
| --- |

Exercise 3 - Mutator Members, Returning References

As long as our original **Date** object is valid, we can add or subtract days from our data member, and the date will remain valid. [Write the member function **addDays()**](https://codecheck.it/files/1911120313duy70oku2qs5138201c9q8yus) which modifies the **cur\_time** data member. Remember that there are **60** seconds to a minute and **60** minutes to an hour, and **24** hours to a day.

| *Screenshot of the tested code and the passing score.* |
| --- |

Exercise 4 - Different Constructors

A **Troll** has a **name** and **strength**. Look at [this example](https://codecheck.it/files/2011151659b5nal00mjnhnhwta1iae7p8cl) which constructs the **Trolls a-d** using different constructors. **Declare** and **implement** the constructors (inline), so that the testing program works correctly. (Note on the expected output, "ctor" stands for constructor.)

| *Screenshot of the tested code and the passing score.* |
| --- |

Exercise 5 - static Members

Complete the **Flea** class so that [the testing program works correctly](https://codecheck.it/files/2004131922dhyrrkyv69na1kh7xuejhrmaw). The **fleas()** **class member function** tells you how many **fleas** exist at any given time in your program. A class function is called using the class name (**Flea::fleas()**) instead of an instance. **Declare** and **implement** the **public** and **private** sections inline. Initialize **static** members where marked with comments.

| *Screenshot of the tested code and the passing score.* |
| --- |

# 