

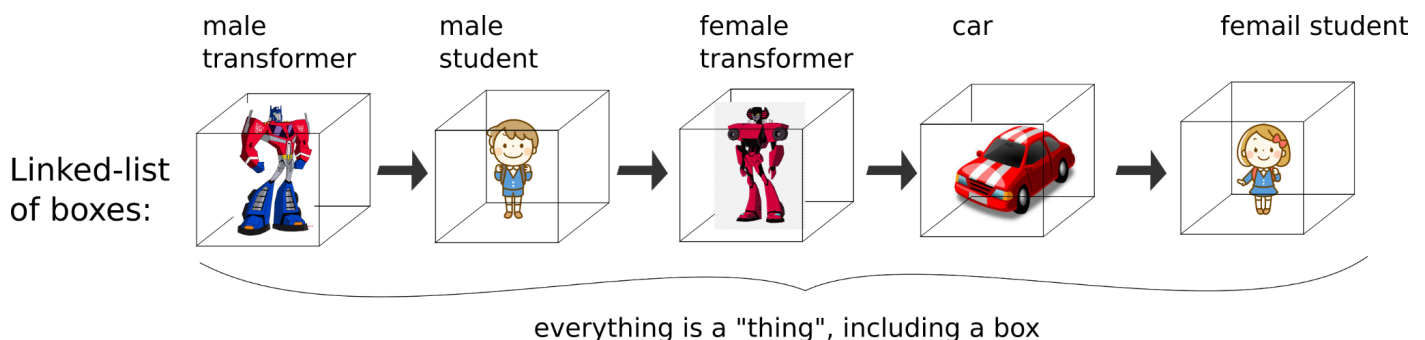
# Introduction to Programming II 2021 - Assignment

**Deadline: 23:59 on 21/04/2020**

In this assignment, you will use and fix the code you implemented for the midterm as a basis, with the following differences:

1. Instead of implementing your own linked list, you must use C++ containers and adaptor (vector, list and stack)
2. The type parameter for the containers is the class Thing that you created previously
3. There is no need for the class Box, which should be removed from your code

**From the previous task:** For this project, you should implement a linked list of boxes "things", which will be sorted, scrambled and then sorted again. Each box "thing" contains a "thing" (see the figure below) and a pointer to the next box.



Many things can be a "thing" - a car, a student, a transformer. Each thing has a unique ID. You should write a C++ program that does the following:

1. Generate a vector with at least 5 items of each type shown in the figure above. Remember that:
  - 1.1. A student is a person - a very annoying person sometimes ♥
  - 1.2. A transformer is also a person
  - 1.3. A transformer is also a car
  - 1.4. Transformers and students can be either male or female
2. Using the items within the vector of item 1, create other two collections:
  - 2.1. A list containing copies of the items
  - 2.2. A stack adaptor of a vector containing copies of all the items
3. Create a function that shuffles the containers described in item 2
4. Create a function that sorts the containers described in item 2
  - 4.1. Sort by ID
  - 4.2. Any comparison-based sorting algorithm is acceptable. Implement it yourself. Do not copy-paste it from somewhere - someone might do the same and you will be caught for plagiarism.
5. Create a function that shows all the "things" in the containers described in item 2
6. Show the highest (or lowest, it doesn't matter) "thing" in each container described in item 2
7. Show only the male "things" in the original vector using somehow a boolean predicate function. In other words, a function that returns true or false given a "thing"

## 1. Additional Requirements

- a. Your code should contain only UTF-8 characters
- b. Your project should be compatible with C++ 20

## 2. Assignment Requirements

- a. Your entire code (to be submitted) should consist of **a single CPP file**
  - i. While we recognize this is not a good practice, it will facilitate a more accurate and easier grading of tens of students
  - ii. The name of the CPP file should have the form: **yourname\_yoursurname.cpp**
- b. There should be a comment before every single method explaining what it does. Use the standard: <https://developer.lsst.io/cpp/api-docs.html>
- c. Forgetting to identify the items using the standard **// ITEM 3.<x>. <Justification>** in the line before the implementation of the feature incurs in 0 points and no appeal

## 3. Grading schema (total of 20 points)

You have to demonstrate knowledge on each concept given to you through the course till this moment. In other words, your code should contain at least one justifiable use of each of the following concepts and requirements:

- a. **The 7-step procedure described in the introduction of this document [6 points]**
- b. At least one non-type parameter (lecture 8) [1 point]
- c. At least one function that is called for different containers (lecture 8 and 11) [2 points]
- d. At least one explicit specialization (lecture 9) [2 points]
- e. **At least one use of `decltype` (lecture 9) [1 point]**
- f. Either the function `shuffle` or the function `sort` (or both) should use iterators (lecture 10) [2 points]
- g. **Filter the vector using a predicate function (lecture 11) [1 point]**

Five points are also given for the quality and readability of your code as perceived by the instructors. For example, naming conventions, indentation, consistence, comments before functions stating arguments, return and what the function does, quality of the visualization.



**IMPORTANT:** Add a one-line comment right before the part of the code in which you implemented each of the requested items. This will allow us to quickly search for your answers. Format of the comment:

```
// ITEM X.y: My justification of why I implemented the item here  
// some code...
```

*Example:*

```
// ITEM 3.a: Bla bla bla..
```

*Forgetting this will incur zero as the grade for the item, without appeal!*

## 4. Notes

- a. Plagiarism will incur zero as the grade of your midterm, independently of if you are the person who implemented the original code or made the copy.
- b. This assignment is individual.