

P2- practical test CPL2, 19 Mai 2022

1. Define a *Vehicle* class with protected attributes, *name* and *producer* character strings or strings, *wheels* of *int* type. The class has adequate constructors/destructor and *set/get* methods for each attribute. The class will also contain a virtual *display* method that will display on the screen the attributes using a *user manipulator* each attribute *left* aligned in a 30-width zone. The class will allow to write a *Vehicle* object in an *output file* by *overloading* the << operator. The *Car* class inherits in *public* mode the class *Vehicle* and adds the *private* attributes *power* and *capacity* of *int* type. The class will consider adequate constructors/ destructor *set/get* methods and will *overload* the << operator including for the new attributes of the class to write a *Car* object in an *output file*. The *Car* class will override the *virtual display* method adding the new attributes of the class.

In *main* define 2 objects for the first *Vehicle* class, first, with predefined attributes, the other, no parameters. In the same mode for the derived class *Car*.

For each class *assign* the first object defined with predefined attributes to the second object no parameters of the class and *clone* the first object with predefined attributes to a *new* specific class object.

Write the initial first objects and the cloned objects from each class in a *file* with the name *Vehicle.txt* (2+2) and *display* all objects (3+3) with *display* methods.

Consider a *base class pointer*. Associate the pointer to the initial base class object, modify the attributes using class setters, *display* the new obtained object and *write* the object in the file. With the *same base class pointer*, associated to the initial derived class object modify with setters all allowed attributes. Also modify all other attributes, *display* the new obtained object, and *write* the object in the file.