1. In the previous course you were asked to implement a basic geometric library. In this exercise you are asked to add constructors to classes written before.

2. Add to the class Point a new field:
static unsigned int num;

Then modify this class in such a way that each new object increase value of the filed num and each removed object decrease this value. Moreover, provide a getter to obtain the current value of the field num. Do the same with class GeoObject. Perform some tests with subclasses of it. What did you notice?

3. Add to the class Point:
static double distance (const Point & p1, const Point & p2);

4. Getters should be of type const. Fix this.

5. Provide two different solutions to the below problem:

```
class Person
{
  std::string name;
  void changeName (std::string name)
  {
     name = name; //solve the conflict here.
  }
}
```

6. Reimplemented your class Polygon in such a way that it allocates dynamically an array or vertices/points. Then, provide an implementation of its destructor where you should release memory allocated, before. For the moment, you can use C style functions like malloc/free to allocate/release memory. We will learn about C++ way later on.

7. Try to improve your code from the previous TD using references and other new elements of C++ and object-oriented programming discovered today. For instance, improve access to the field Address of the class Person.

8. Implement function print(const GeoObject & obj) and try to call it with several objects of different classes which inherit from GeoObject. What did you notice? Can we access fields/methods which are not defined in GeoObject?