

# 02393 C++ Programming Exercises

## Assignment 7

**Hand-in via** <https://dtu.codejudge.net/02393-e17/assignment/>

**Fun with vectors** The goal of the exercise is to implement a class for two dimensional vectors. Note that here “vector” refers to pairs of double values, not to the class `vector` of the standard C++ library. We use the mathematical notation  $\begin{pmatrix} x \\ y \end{pmatrix}$  for vectors. You are given a header-file `vector2d.h` (see CampusNet), that contains the declaration of the class `v2d` for vectors in two-dimensional space. Moreover we provide a program `main.cpp` file that uses the `v2d` class that you can use to do some tests. You should implement the class in a file `vector2d.cpp` as follows:

1. The constructor `v2d(double a, double b)` should build a vector  $\begin{pmatrix} x \\ y \end{pmatrix}$ .
2. The constructor `v2d(const v2d & v)` is meant to build a vector that is exactly like vector `v`.
3. The destructor `~v2d()` does not need to do anything special.
4. The assignment operator `v2d & operator=(const v2d &v)` updates a vector to make it exactly like vector `v`.
5. The vector addition method `v2d & operator+(const v2d &v)` updates a vector by adding another vector `v` to it. Remember that

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \end{pmatrix}$$

6. The scalar multiplication method `v2d & operator*(double k)` updates a vector by multiplying it by a scalar factor `k`. Remember that scalar multiplication is defined by:

$$k \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} k \cdot x \\ k \cdot y \end{pmatrix}$$

7. The scalar product method `double operator*(const v2d &v)` multiplies a vector by another vector `v` and returns the result. Remember that the scalar product of vectors is defined by

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} * \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = x_1 \cdot x_2 + y_1 \cdot y_2$$

8. Method `double length()` computes the length of the vector. Remember that the length of a vector  $w = \begin{pmatrix} u \\ v \end{pmatrix}$  is  $\sqrt{w * w}$ .

## Hints

- You need to upload your file `v2d.cpp` only, and not the rest of the files.
- Most of the *operator* methods you need to implement (`=`, `*`, etc.) do not need to generate a new vector, but change the vector for which the method was called. For example, if `u` and `v` are vectors then `u + v` will update `u` (with the addition of `u` and `v`).
- Most of the *operator* methods you need to implement (`=`, `*`, etc.) need to return the very same vector (by reference). One of the consequences is that `(u + v) + w` will have the effect of updating `u` (with the addition of `u`, `v` and `v`). Recall that the object for which the object was invoked is accessed with `*this`. Many of your methods, hence, will need to finish with `return *this;` which returns the current object by reference (not a pointer!).

**Challenge.** Can you use templates to make class parametric with respect to the datatype of the elements (e.g. `double`, `float`, etc.)? Can you use templates to generalize the class to arbitrary dimensions ( $n$ -dimensional vectors)?