Ejercicio Orientacion a Objetos:

- Implementar dentro del namespace cpp_math una clase llamada Point que representa un punto en el espacio tridimensional. Deben tener Point.h y Point.cpp. Colocar todas las definiciones en el cpp.

- 3 variables privadas miembros de tipo float* llamadas mX, mY y mZ respectivamente

- Implementar constructor por defecto que debe poner las 3 variables en 0.0f

- Implementar constructor con parametros que tome un x, y y z para inicializarlas

- Implementar constructor por copia

- Implementar operator= (assignment operator). Pueden consultar en Google su implementacion que es muy conocida.

- Implementar funciones getX() getY() getZ() que devuelven x, y y z respectivamente

- Implementar setX() setY() setZ() setXY() setYZ() setXYZ() que setean las components del punto. No retornan nada y tomar floats segun corresponda

- Implementar Point getAddition(const Point& p) que retorna un nuevo objeto donde cada variable miembro es la suma de la variable miembro del objeto Point que invoca el metodo y la de p.

- Implementar const Point& add(const Point& p) que suma p al objeto Point que invoca el metodo y devuelve el mismo objeto que invoca el metodo.

- Implementar getSubtraction() y subtract() que son analogos al anterior.

- Implementar float distance(const Point& p) que retorna la distancia entre el objeto Point que invoca el metodo y p.

- Implementar el destructor

Notas:

- Deben testear las funcionalidades (menos la del destructor) utilizando asserts asi pueden chequear que cada operacion funciona bien.
- Tengan en cuenta que no pueden comparer floats usando ==. Investiguen y hagan su propia funcion para comparar floats. Coloquenla en el namespace cpp_math_utils en el archive MathUtils.h y MathUtils.cpp
- Todos los metodos anteriores son publicos