Esercitazione sull'overload degli operatori

March 17, 2025

Questa esercitazione è obbligatoria e dev'essere consegnata al massimo alla data di consegna prevista per l'esercitazione sull'input output.

Un numero complesso è un numero della forma $z = a + ib$, dove $i = \sqrt{-1}$ è l'unità immaginaria. I due numeri reali a e b sono detti rispettivamente *parte reale* e *parte immaginaria*.

Si definisca una classe template `complex_number` che modella i numeri complessi. La classe template prende come parametro il tipo `T` con il quale si rappresentano a e b sulla macchina. La classe deve funzionare correttamente per `T = float` e `T = double`.

Tale classe template deve:

- Avere un costruttore di default
- Avere un costruttore user-defined per l'inizializzazione di parte reale ed immaginaria
- Avere metodi che restituiscano il coniugato, la parte reale e la parte immaginaria
- Fornire un overload dell'operatore `<<` per stampare il numero complesso. Se per esempio $a = 1$ e $b = 2$, dev'essere stampato `1+2i`, mentre se $b = -2$ dev'essere stampato `1-2i`.
- Fornire un overload degli operatori `+=` e `+`.
- Fornire un overload degli operatori `*=` e `*`.
- Opzionale: in modo simile a quanto fatto in classe nel caso di `rational`, utilizzare i *concept* per vincolare T ad essere un tipo floating poing. Verificare su `https://en.cppreference.com/w/cpp/header/concepts` qual'è il concept da utilizzare a tal scopo.

Similmente a quanto fatto su `rational`, gli operatori `+` e `*` implementati devono soddisfare correttamente i requisiti di commutatività quando l'operazione svolta è tra un `complex_numer<T>` ed un `T`.