

# Zaawansowane C++

## Lista 1: Szablony

### Zadanie 1

Stwórz przestrzeń nazw `cpplab`, której będziemy używać przez całe zajęcia i stwórz w niej następujące szablony:

- Szablon funkcji `cpplab::add` przyjmującej dwa argumenty i zwracającej wynik dodawania;
- Szablon funkcji `cpplab::alias` przyjmującej dwa argumenty oraz operację podaną jako trzeci argument. Szablon funkcji ma zwracać wynik operacji wykonanej na przekazanych argumentach;

W obu przypadkach jawnie zdefiniuj typ zwracanego argumentu używając specyfikatora `decltype`. Przetestuj działanie funkcji dla różnych typów argumentów np. numerycznych i `std::string`, jak i różnych operacji w przypadku szablonu `cpplab::alias`. Jedną z operacji zdefiniuj za pomocą wyrażenia lambda implementującego np. funkcję `max(a,b)`.

### Zadanie 2

a) Stwórz szablon klasy `cpplab::vector` z typem danych podanym jako argument szablonu. Klasa powinna wspierać dynamiczne dodawanie i usuwanie elementów oraz zmianę długości wektora. Domyślny konstruktor nie powinien alokować żadnej pamięci. Zaimplementuj podstawowe funkcjonalności tj. operatory do pobrania i modyfikacji poszczególnych składowych wektora. Dodatkowo klasa powinna mieć zdefiniowane `value_type`.

b) Dla klasy `cpplab::vector` dopisz operator mnożenia skalarnego tak, aby dało się policzyć iloczyny skalarne wektorów `cpplab::vector` oraz `std::vector` w dowolnej konfiguracji argumentów i typu składowych wektorów.

### Materiały pomocnicze:

- Szablony: <https://www.youtube.com/watch?v=XN319NYEScE&t=2082s>
- Type Punning: [https://www.youtube.com/watch?v=8egZ\\_5GA9Bc&t=4s](https://www.youtube.com/watch?v=8egZ_5GA9Bc&t=4s)
- Namespaces: <https://www.youtube.com/watch?v=ts1Eek5w7ZA&t=2s>