

C++ Compact Course

Lab Sheet 4

Aufgabe 1:

Schreibe eine Funktion `get_maximum_number(container<int>)`, die sowohl `std::vector<int>` als auch `std::list<int>` als Eingabe akzeptiert und den größten Wert zurück gibt.

Aufgabe 2:

Schreibe eine Funktion `multiply_by_two(container)`, welche `std::vector` und `std::list` mit einem beliebigen Datentypen (`float`, `int`, ...) als Eingabe akzeptiert, dessen Werte mit 2 multipliziert und in einem neuen `std::vector/std::list` speichert und zurück gibt.

Aufgabe 3:

Schreibe eine Funktion `get_maximum_number_anytype(container)`, welche `std::vector` und `std::list` mit einem beliebigen Datentypen (`float`, `int`, ...) als Eingabe akzeptiert und den größten Wert zurück gibt. Der Rückgabewerte muss dem Datentypen den `std::vector/std::list` speichert entsprechen.

Aufgabe 4:

Schreibe eine Funktion `convert_container_type(container)`, welche `std::vector<int>` und `std::list<int>` als Eingabe akzeptiert.

Bei einem Aufruf mit `convert_container_type<std::list<int>>` eine `std::list<int>` und bei einem Aufruf mit `convert_container_type<std::vector<int>>` einen `std::vector<int>` mit den Werten der Eingabe zurück gibt.

Aufgabe 5:

Löse Aufgabe 4 ohne Spezialisierung zu verwenden.

Aufgabe 6:

Schreibe eine Funktion `multiply_by_value(container, value)`, welche `std::vector` und `std::list` mit einem beliebigen Datentypen (`float`, `int`, ...) und einen zweiten Parameter vom selben Datentype akzeptiert. Die Funktion sollte die Werte aus `std::vector`/`std::list` mit dem zweiten Parameter multiplizieren, in einem neuen `std::vector`/`std::list` speichert und diese(n) zurück geben.

Aufgabe 7:

Schreibe eine Funktion `add_two_points(p1, p2)`, welche `PointClassA` und `PointClassB` in einer beliebigen Kombination als Eingabe akzeptiert und die Membervariablen `x` und `y` von `p1` und `p2` addiert in einem neuen `PointClassA` speichert und diesen zurück gibt.
