

Słowa Kluczowe w C++

- ☐ **asm** - wstawia instrukcje asemblera
- ☐ **auto** - deklaruje zmienną automatyczną
- ☐ **bool** - deklaruje zmienną typu logiczną prawda/fałsz
- ☐ **break** - przerywa działanie pętli
- ☐ **case** - część instrukcji **switch**
- ☐ **catch** - łapie wyjątki
- ☐ **char** - deklaruje zmienną typu znak
- ☐ **class** - deklaruje klasę
- ☐ **const** - deklaruje zmienną typu stałego
- ☐ **const_cast** - rzutowanie uzmienniające stałą
- ☐ **continue** - wymusza ponowne wykonanie pętli
- ☐ **default** - domyślna część instrukcji **switch**
- ☐ **delete** - zwalnia pamięć
- ☐ **do** - część pętli **do/while**
- ☐ **double** - deklaruje zmienną typu rzeczywistego o podwójnej precyzji
- ☐ **dynamic_cast** - rzutowanie w czasie trwania programu
- ☐ **else** - część wyrażenia warunkowego **if**
- ☐ **enum** - definiuje typ wyliczeniowy
- ☐ **explicit** - w konstruktorze zabrania domyślnych konwersji

- ☐ **·extern** - definiuje zmienną zewnętrzną
- ☐ **·false** - wartość typu **bool** oznaczająca fałsz
- ☐ **·float** - deklaruje zmienną typu rzeczywistego
- ☐ **·for** - pętla
- ☐ **·friend** - deklaruje przyjaźń
- ☐ **·goto** - wykonuje skok
- ☐ **·if** - instrukcja warunkowa
- ☐ **·inline** - wstawia funkcję w linii
- ☐ **·int** - deklaruje zmienną typu całkowitego
- ☐ **·long** - deklaruje zmienną typu całkowitego o większym lub równym zakresie niż **int**
- ☐ **·mutable** - uzmiennia stałą
- ☐ **namespace** - przestrzeń nazw
- ☐ **·new** - alokuje pamięć
- ☐ **·operator** - tworzy przetadowaną funkcję operatorową
- ☐ **·private** - deklaruje prywatne składniki i metody klasy
- ☐ **·protected** - deklaruje zabezpieczone składniki i metody klasy
- ☐ **·public** - deklaruje publiczne składniki i metody klasy
- ☐ **·register** - optymalizuje dostęp do zmiennej ze względu na szybkość
- ☐ **·reinterpret_cast** - zmienia typ zmiennej

- ☐ **·return** - zwraca wartość z funkcji
- ☐ **·short** - deklaruje zmienną typu całkowitego o mniejszym lub równym zakresie niż **int**
- ☐ **·signed** - określa, że zmienna jest ze znakiem
- ☐ **·sizeof** - zwraca rozmiar typu lub zmiennej
- ☐ **·static** - tworzy zmienną która istnieje przez całe wykonywanie się programu
- ☐ **·static_cast** - operator rzutowania
- ☐ **·struct** - tworzy strukturę
- ☐ **·switch** - wykonuje kod zależny od danej zmiennej
- ☐ **·template** - tworzy szablon
- ☐ **·this** - wskaźnik do obecnie używanego obiektu
- ☐ **·throw** - wyrzuca wyjątek
- ☐ **·true** - wartość typu **bool** oznaczająca prawdę
- ☐ **·try** - wykonuje kod który może wyrzucić wyjątek
- ☐ **·typedef** - tworzy synonim do istniejącego typu
- ☐ **·typeid** - opisuje typ obiektu
- ☐ **·typename** - w szablonach oznacza że następujący po nim symbol reprezentuje typ (synonim **class**)
- ☐ **·union** - tworzy unię
- ☐ **·unsigned** - deklaruje zmienną bez znaku
- ☐ **·using** - używa przestrzeni nazw
- ☐ **·virtual** - tworzy funkcję wirtualną
- ☐ **·void** - deklaruje zmienną z nieprzypisanym typem
- ☐ **·volatile** - ostrzega kompilator że zmienna może zostać zmodyfikowana nieoczekiwanie

- ❑ `wchar_t` - deklaruje "szeroką" zmienną znakową
- ❑ `while` - pętla z wyrażeniem warunkowym