

C++ - operatory działań

Operator	Działanie	Przykład
+	dodawanie	4 + 7 → 11
-	odejmowanie	21 - 4 → 17
*	mnożenie	4 * 3 → 12
/	dzielenie	5 / 2 → 2
%	reszta z dzielenia całkowitego	5 % 2 → 1

C++ - operatory porównania

Operator	Działanie	Przykłady	
==	równe	(4 == 4) → 1	(4 == 5) → 0
!=	różne	(4 != 7) → 1	(4 != 4) → 0
<	mniejsze	(3 < 9) → 1	(9 < 3) → 0
<=	mniejsze równe	(3 <= 3) → 1	(5 <= 4) → 0
>	większe	(4 > 1) → 1	(1 > 4) → 0
>=	większe równe	(4 >= 4) → 1	(3 >= 4) → 0

C++ - polecenia

Instrukcja <code>cout</code> - wypisywanie			
<code>cout << "las";</code>	<code>cout << 324;</code>	<code>cout << 3 << ' ' << 24;</code>	<code>cout << 3 << endl;</code>
↓	↓	↓	↓
las	324	3 24	3 24
Funkcje			
<code><typ wyniku> <nazwa funkcji>(<typ parametru><parametr>) { <instrukcje> }</code>			
<pre>int godziny(int t) { int g = t / 60; return g; }</pre>		<pre>int droga(int v, int t) { return v * t; }</pre>	
<pre>int main() { cout << godziny(325); } ↓ 5</pre>		<pre>int main() { cout << droga(53, 2); } ↓ 106</pre>	

Instrukcja przypisania

```
<typ wartości> <nazwa> = <wartość>;
```

```
int x = 45;
```

Instrukcja warunkowa - prosta

```
if (<wyrażenie>) {
    <wykonaj instrukcje>
}
```

```
if (t == 4)
    cout << "dobry";
```

Instrukcja warunkowa - złożona

```
if (<wyrażenie>) {
    <wykonaj instrukcje1>
}
else {
    <wykonaj instrukcje2>
}
```

```
if (x >= 0)
    cout << "nieujemny";
else
    cout << "ujemny";
```

Pętla for

```
for (instrukcja_na_start; warunek; instrukcja_w_pętli) {
    <instrukcje>
}
```

```
for (int i = 0; i <= 5; i++)
    cout << i + 1 << endl;
```

Pętla while

```
while (<wyrażenie>) {
    <instrukcje>
}
```

```
while (x > 100) {
    cout << x << endl;
    x -= 20;
}
```

Tablice jednowymiarowe

```
int tablica[4] = {2, 4, 5, 6};
```

```
cout << tablica[0]; → 2
cout << tablica[3]; → 6
```

```
for (int i = 0; i < 4; i++)
    cout << tablica[i] << " ";
```