

Компактні умови та Switch

Тема 2.3: Тернарний оператор та оператор вибору

Тернарний оператор (?:)

"Однорядковий if" для простих дій.

(Умова) ? Дія_ТАК : Дія_НІ;

Python

```
res = "Ok" if x > 5 else "No"
```

C++

```
res = (x > 5) ? "Ok" : "No";
```


Приклад: Модуль числа

Логіка

Якщо число менше 0, змінюємо знак на протилежний. Інакше — залишаємо як є.

- $-5 \rightarrow 5$
- $10 \rightarrow 10$

```
int x = -5; // Якщо  $x < 0$ , то  $-x$ , інакше  $x$ 
int abs_x = (x < 0) ? -x : x; cout << abs_x; // 5
```

Оператор вибору (switch)

Зручно використовувати, коли потрібно перевірити **конкретні значення** (меню, кнопки).



Кавовий автомат: 1 – Еспресо, 2 – Лате...

```
switch (choice) { case 1: cout << "Espresso"; break;  
case 2: cout << "Latte"; break; default: cout <<  
"Error"; }
```




УВАГА: Пастка з BREAK

У C++ case не зупиняється автоматично!

Якщо забути break, програма "провалиться" далі (fall-through).

```
case 1: cout << "One"; // Забули break! case 2: cout << "Two"; // Це теж виконається!
```

Корисний трюк: Групування

Використовуємо "провал" на користь

Якщо для різних значень треба зробити одну дію, ми перелічуємо case без break.

📅 Будні: 1, 2, 3, 4, 5

☕ Вихідні: 6, 7

```
switch (day) { case 1: case 2: case 3: case 4: case  
5: cout << "Work :("; break; // Один на всіх case 6:  
case 7: cout << "Weekend!"; break; }
```


Практична робота 2.3

Час створити калькулятор та меню!

