

# Програмування мовою C++

24

Функції:  
перевантаження

# Перевантаження функцій

- **Перевантаження функції - створення різних варіантів функції з однаковими назвами але різними прототипами.**
- **Різні варіанти перевантаженої функції відрізняються кількістю і типом аргументів, а також можуть відрізнитись типом результату.**
- **При виклику функції версія функції, яка викликається, визначається в контексті команди виклику функції (по кількості чи типу аргументів).**

# Перевантаження функцій - приклад

```
// Один int-аргумент:
```

```
void show(int a){  
    cout<<" [1]: "<<a<<endl;  
}
```

```
// Один char-аргумент:
```

```
void show(char a){  
    cout<<" [2]: "<<a<<endl;  
}
```

```
// Два int-аргументи:
```

```
void show(int a,int b){  
    cout<<" [3]: "<<a<<" & ";  
    cout<<b<<endl;  
}
```

```
// Два char-аргументи:
```

```
void show(char a,char b){  
    cout<<" [4]: "<<a<<" & ";  
    cout<<b<<endl;  
}
```

# Перевантаження функцій - приклад

Програма:

```
#include <iostream>
using namespace std;
void show(int a){
    cout<<"[1]: "<<a<<endl;
}
void show(char a){
    cout<<"[2]: "<<a<<endl;
}
void show(int a,int b){
    cout<<"[3]: "<<a<<" & ";
    cout<<b<<endl;
}
void show(char a,char b){
    cout<<"[4]: "<<a<<" & ";
    cout<<b<<endl;
}
int main(){
    show(100);        // 1
    show('A');        // 2
    show(200,300);   // 3
    show('B','C');    // 4
    show(400,'D');    // 3
    return 0;
}
```

Результат

виконання:

```
[1]: 100
[2]: A
[3]: 200 & 300
[4]: B & C
[3]: 400 & 68
```

# Перевантаження функцій

Дозволяється перевантажувати функції, аргументи яких мають значення за замовчуванням. При цьому перевантаження має виконуватись так, аби по команді виклику можна було однозначно зрозуміти, яка версія функції викликається. Також слід враховувати автоматичне приведення типів

**Так робити не можна!**

```
void show(int a, int b=0) {}
```

```
void show(int a) {}
```

```
show(100); // Яка версія?
```

**Тут можлива помилка!**

```
void show(int a, char b) {}
```

```
void show(char a, int b) {}
```

```
show('A', 'B'); // Яка версія?
```