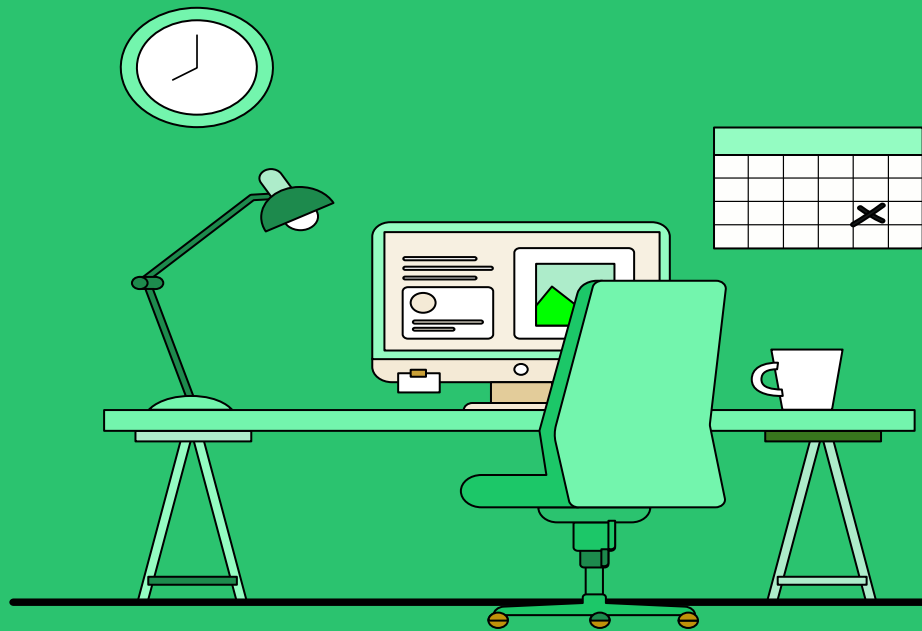


ПЕРЕВАНТАЖЕННЯ ФУНКЦІЙ

C++



Що робить +?



```
int a=1, b=1;  
double f1=1.0, f2=1.0;  
string s1="1", s2="1";
```

```
cout<<a+b;
```

```
cout<<f1+f2;
```

```
cout<<s1+s2;
```

Перевантаження функцій

Перевантаження функцій - це особливість в C ++, яка дозволяє визначати декілька функцій з одним і тим же ім'ям, але з різними параметрами

```
Add (int a,int b)
```

```
{
```

```
.....
```

```
}
```

```
Add(float a,float b)
```

```
{
```

```
.....
```

```
}
```

```
double perimeter(int r) {  
    return 2*(3,14)*r;  
}
```

```
double perimeter(int l , int w) {  
    return 2*(l+w);  
}
```

```
double perimeter(int a, int b, int c) {  
    return a+b+c;  
}
```

```
int main() {  
    cout << "Perimeter of Circle: " << perimeter(5) << endl;  
    cout << "Perimeter of Rectangle: " << perimeter(10,20) <<  
endl;  
    cout << "Perimeter of Triangle: " << perimeter(5,10,15) <<  
endl;  
    return 0;  
}
```



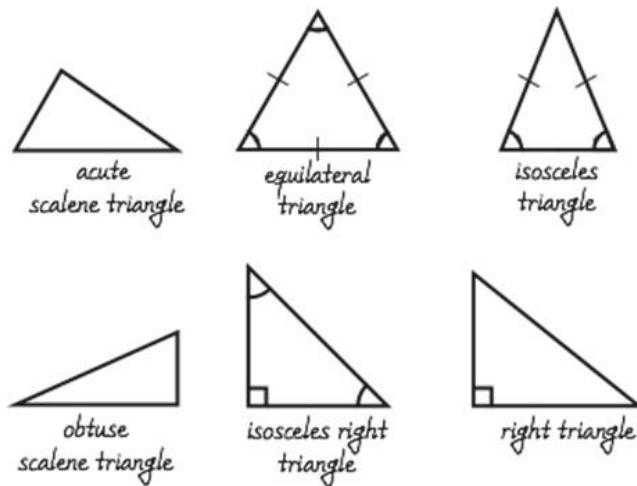
Знайдіть відмінності

```
int area(int a, int b)
{
    return 0.5*a*b;
}

float area(float a, float b)
{
    return 0.5*a*b;
}

int area(int a, int b, int c)
{
    int p = (a + b + c) / 2;
    return sqrt(p*(p-a)*(p - b)*(p - c));
}
```

Розберемо на прикладі.
Площа трикутника.
Скільки знаєте формул?



Набори параметрів перевантажених функцій можуть відрізнятися

- порядком проходження
- кількістю
- типом

```
int max(int a, int b) {  
    if (a >= b)  
        return a;  
    else  
        return b;  
}
```

```
float max(float a, float b) {  
    if (a >= b)  
        return a;  
    else  
        return b;  
}
```

Function Overloading in C++



```
Syntax:  
void add(int a, int b);  
void add(float a, float b);  
#include  
using namespace std;  
void print(int x) {  
    cout << " Here is the integer " << x << endl;  
}  
void print(double y) {  
    cout << " Here is the float " << y << endl;  
}
```

```
void myFunction()  
void myFunction(int a)  
void myFunction(float a)  
void myFunction(int a, float b)  
float myFunction (float a, int b)
```

```
#include <iostream>
using namespace std;
```

```
void display(int);
void display(float);
void display(int, float);
int main() {
```

```
    int a = 5;
    float b = 5.5;
    display(a);
    display(b);
    display(a, b);
    return 0;
}
```



```
void display(int var) {
    cout << "Integer number: " << var << endl;
}
void display(float var) {
    cout << "Float number: " << var << endl;
}
void display(int var1, float var2) {
    cout << "Integer number: " << var1;
    cout << " and float number:" << var2;
}
```



```
#include <iostream>
using namespace std;
```

```
int fun (int a, int b)
{ return (a*a+b*b);}
```

```
float fun (float x, float y)
{ return (x*x+y*y); }
```

```
int main( )
```

```
{ int x1 = 10, y1 = 3;
```

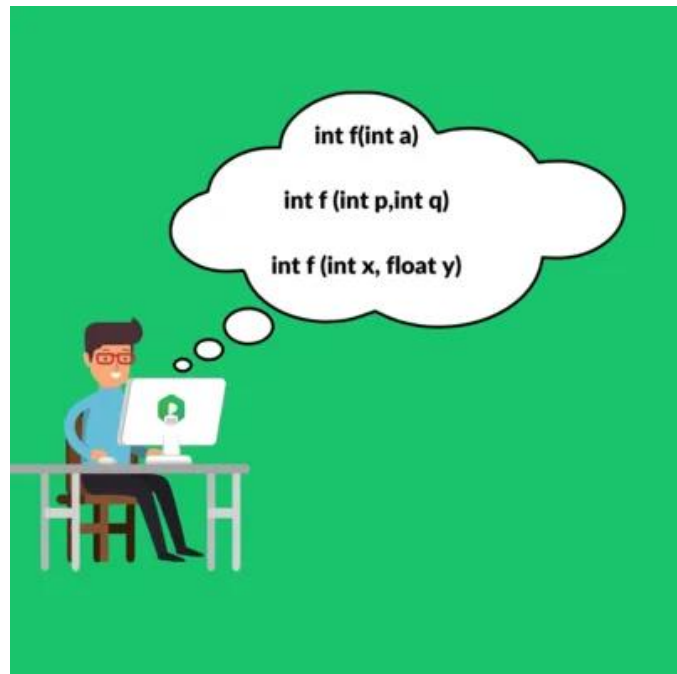
```
float x2 = 13.75, y2 = 11.25;
```

```
cout << <"Сума квадратів чисел " << x1<< " і " << y1 << "= "<< fun(x1, y1) << endl;
```

```
cout<<"Сума квадратів чисел " << x2<< " і " << y2 << "= "<< fun(x2, y2);
```

```
return 0;
```

```
}
```



Function Overloading in C++

