

Простейшая программа

```
#include <iostream>
using namespace std;
const double Pi = 3.1415;
int main()
{
    double r;
    cin >> r;
    double S = Pi*r*r;
    cout << "Площадь круга = "<<S<<endl;
}
```

Явное указание пространств имен

```
std::cin >> r;
double S = Pi*r*r;
std::cout << S << std::endl;
```

Инициализация в стиле initializer lists

```
double d {3.14};
```

Ввод/ вывод

```
cin >> a >> b >> c;
cout << a << b << c;
```

Основные типы

```
short int      (2) // или short
unsigned int   (4) // или unsigned
int            (4)
float          (4)
double         (8)
bool           (1)
char           (1)
unsigned char  (1)
string
```

Автовывод типов

```
auto r = 2.5;
auto z = sqrt(y);
```

Константы

```
const int i {5};
double const Pi = 3.1415;
```

Неявные преобразования типов

```
int i = 3.7; // i == 3
char c = 128;
c = 'z' - 2;
bool b = -1; // true
int i = true; // i == 1
```

Основные операции

```
i << 3 // i shl 3
i >> 2 // i shr 2
a += 2; // a = a+2;
b = a++; // t = a; a++; b = t;
b = ++a; // a++; b = a;
7/3 // 7 div 3
7%3 // 7 mod 3
(i<0 || i>2) // или
(i>=2 && i<=3) // и
!(i>2) // не
& | ^ ~ // побитовые and, or, xor, not
min = a<b ? a : b; //условная операция
a = b = c; //множественное присваивание
```

Условный оператор

```
if (a<b)
    min = a;
else min = b;

if (i==0 && i!=1)
    i++;
else i--;

if (x<y)
{
    double t {x};
    x = y;
    y = t;
}
```

Оператор выбора

```
switch (i)
{
    case 1:
        cout << 1;
        j++;
        break;
    case 2:
    case 3:
        cout << 2;
        break;
    default:
        cout << 3;
}
```

Операторы цикла

```
i = 5; j = 0;
while (i>0)
{
    i--;
    j++;
}

do
{
    i++;
    j--;
} while (i<5);

for (int i=0; i<10; i++)
    cout << i << " ";

for (double d=0; d<10; d+=0.2)
    cout << d << " ";
```

Операторы break и continue**Функции**

```
void f(int i, int j)
{
    cout << i*j << endl;
}

int abs(int a)
{
    return a>0 ? a : -a;
}
```

Ссылки

```
Ссылка - другое имя объекта:
int a = 5;
int& ra = a;
ra = 3;
```

Передача параметра по ссылке

```
void swap (int& a, int& b)
{
    auto v = a;
    a = b;
    b = v;
}

int c,d;
swap(c,d);
```

Стандартные функции

```
#include <cmath>
abs(x)
floor(x) - ближайшее целое <= x
ceil(x) - ближайшее целое >= x
sin(x) cos(x) tan(x)
exp(x) log(x) log10(x)
pow(x,y)
sqrt(x)
```

Генерация случайных чисел

```
#include <iostream>
#include <cstdlib>
#include <ctime>
const int size = 10;
const int maxNumber = 100;
```

```
int main()
{
    srand( (unsigned)time( NULL ) );
    for (auto i=0; i<size; i++)
        cout << rand()%maxNumber << " ";
    system("pause");
}
```

Заголовочные файлы**Что могут содержать заголовочные файлы**

- Объявление переменных
- Объявление функций
- Определение констант
- Определение типов (typedef)
- inline-функции
- Другие директивы #include

Чего не могут содержать заголовочные файлы

- Определение переменных
- Определение функций

Одномерные массивы

```
int a[5]; // a[0],...,a[4]
int mm[] {1,2,3,1}
```

Цикл foreach по массиву

```
int mm[] {1,2,3,1};
for (int& x: mm)
    x++;
for (auto x: mm)
    cout << x;
```

Является ли массив симметричным

```
int a[10] {1,2,3,4,5,5,4,3,2,1};
bool f = true;
int i = 0, j = 9;
while (i<j)
    if (a[i++] != a[j--])
    {
        f = false;
        break;
    }
```

Передача массива как параметра

```
bool contains(const int a[], int n,
              int k)
{
    for (int i=0; i<n; i++)
        if (a[i]==k)
            return true;
    return false;
}
```

Строки в стиле C (null terminated strings)

```
char s[] = "C++";
char s1[80] = "";
cout << s; //выводится содержимое до \0
cin >> s; //ввод до первого пробела
cin.getline(s1,80);
```

Двумерные массивы

```
int c[3][4];
int ff[2][3] {{1,1,1},{2,3,4}};
int ff1[2][3] = {1,1,1,2,3,4};
```

Вывод двумерного массива

```
for (int i=0; i<3; i++)
{
    for (int j=0; j<4; j++)
        cout << c[i][j] << ' ';
    cout << endl;
}
```

Определение типов

```
typedef unsigned char byte;
typedef int IArr[3];
```

Структуры

```
struct complex
{
    double re,im;
};
complex c1 {0,1}, c2 {1,2};
c1.re ++;
c1.im = 3;
c2 = c1;
```

Перечисления в стиле C

```
enum MyType {A,B,C}; // A=0; B=1; C=2
enum YourType {D=2,E,F=0};
// D=2; E=3; F=0
MyType m = A;
```

Перечисления в новом стиле

```
enum class Color { Red, Green, Blue };
Color c = Color::Blue;
```