Простейшие программы

```
#include <iostream>
int main()
{
  std::cout << "Hello, World!\n";
}
```

```
#include <iostream>
using namespace std;
const double Pi = 3.14159265;
int main()
{
  double r;
  cin >> r;
  double S = Pi*r*r;
  cout << "Площадь круга = " << S << endl;
}
```

Явное указание пространств имен

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

using объявления

```
using std::cout;
cout << S << std::endl;
```

Ввод / вывод

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

Основные типы

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

Автовывод типов (C++11)

```
auto r = 2.5;
auto z = 'z';
auto s = "str";
```

Инициализация и неявные преобазования типов

```
int i = 3.7;  // i == 3 - warning:
  сужающее преобразование (сп)
char c = 128; // warning (сп)
c = 'z' - 2;
bool b = -1;  // true - warning (сп)
int j = true; // j == 1
```

Инициализация в стиле C++11

```
double d { 3.14 };
auto s { "C++" };

int i { 3.7 };  // error!
char c { 128 }; // error!
bool b { -1 };  // error!
int j { true }; // OK
```

Константы

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

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;
}

auto add(int x, int y) {
  return x+y;
}
```

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

Генерация случайных чисел

```
#include <iostream>
#include <ctime>
#include <random>
#include <functional>

int main() {
  default_random_engine engine(time(0));
  uniform_int_distribution<> dist{1,10};
  for (int i = 0; i < 100; i++)
    cout << dist(engine) << " ";
}
```

Структуры

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

struct Person
{
  string name;
  int age;
};
Person p {"Иванов",23};
p = {"Попова",21};
```

Кортежи (C++11)

```
#include <tuple>
tuple<int,char,string> t{4,'z',"C++"};
auto x = make_tuple(4, 'z', "C++");
auto s = get<2>(x);
int i; char c; string s;
tie(i, c, s) = t; // распаковка
```

Перечисления в стиле C

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

Перечисления в стиле C++11

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

Векторы

```
#include <vector>
using namespace std;
vector<int> a(5); // a[0],...,a[4]
vector<int> mm {1,2,3,1};
```

Цикл foreach по вектору

```
for (int& x: mm)
  x++;
for (auto x: mm)
  cout << x << " ";
```

Является ли вектор симметричным

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

Передача вектора как параметра

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

initializer_list (C++11)

```
#include <initializer_list>

using namespace std;

void print(initializer_list<int> il) {
  for (int x : il)
    cout << x << " ";
}

print({ 2,4,6,8 });
```

Строки string

```
#include <string>
using namespace std;
string s = "C++", s1;
s1 = s;
s[0] = s[s.size()-1];
s = s + "17";
cout << s << endl;
cin >> s; //ввод до первого пробела
getline(cin,s); //ввод до конца строки
int i = stoi("123");
string s = to_string(i);
```

Передача строки как параметра

```
bool starts_with(const string& s, char c)
{
  return s[0] == c;
}
```

Определение типов (C++11)

```
using byte = unsigned char;
using IntVec = vector<int>;
```

Заголовочные файлы

Что могут содержать заголовочные файлы

```
Объявление переменных
Объявление функций
Определение констант
Определение типов
inline-функции
Другие директивы #include
```

Чего не могут содержать заголовочные файлы

```
Определение переменных
Определение функций
```