**Переменные.** Переменные - это область памяти, в которой содержится значение данной переменной и которая имеет имя, по которому мы можем к ней обратиться.

**Идентификаторы.** Проще говоря - имена переменный, функций и т.д. Идентификатор может состоять из букв, цифр и специальных символов. Правила составления идентификаторов:  
1. начинается с буквы или с нижнего подчеркивания (пример: abc, a1, \_a);  
2. регистр букв имеет значение (если переменная названа а, то попытка использовать её как А обернётся ошибкой);  
3. компилятор различает только первые 31 символ;  
4. идентификатор не может совпадать с зарезервированными словами.  
**Важно.** Если вы назвали одну переменную локальную переменную а, то вы не можете назвать так другую локальную переменную в этой же функции, т.к. это приведет к ошибкам.

**Зарезервированные слова.** Ключевые слова, которые используются в С++. К ним относят: базовые типы данных, пользовательские типы данных, модификаторы типов, операторы и классы памяти.

**Константы.** Константа - это переменная, которая имеет постоянное, неизменное значение. Попытка изменить значение константы в процессе выполнения программы обернётся ошибкой. Для объявления константы нужно перед типом переменной указать ключевое слово const:  
const int a = 1;

**Структура оперативной памяти.** Память состоит из ячеек, каждая из которых имеет свой адрес. Адреса расположены по порядку и записаны в виде двоичных чисел. Как правило, одна ячейка содержит один байт информации. Один байт состоит из 8 бит. То есть каждая ячейка состоит из 8 разрядов. Байт является минимальной единицей, к которой мы можем обратиться. Комбинации из нескольких (как правило, четырёх или восьми) байтов называются словами. Слова иногда используются командами в качестве единицы для обращения.

**Типы данных.** В С++ есть множество типов и даже есть возможность создавать свои типы. Очень важно, чтобы тип каждой переменной, функции и т.д. был известен компилятору в точке использования. Если тип не будет известен, то это приведёт к ошибкам. Разные типы данных имеют разные диапазоны значений, размер и допустимые операции.

| char | символьный |
| --- | --- |
| int | целочисленный |
| bool | логический |
| float | с плавающей запятой |
| double | двойной точности с плавающей запятой |

**Преобразование типов**Существует два варианта, как изменить тип данных у значения:  
1. Явное преобразование (с помощью специальных команд):  
2. Неявное преобразование — преобразование, которое происходит вынужденно из-за копирования значения в другую переменную с другим типом данных.

Преобразование типа данных из большего типа в меньший является опасным преобразованием, т.к. возможна потеря информации. Определить, какой тип является меньшим, какой - большим, можно по количеству байт, выделяемому на него (смотри таблицу типов данных в презентации).

Пример кода с преобразованиями типов данных:

#include <iostream>

int main() {

// Явное преобразование (конверсия)

double doubleNumber = 3.14159265;

int intNumber = static\_cast<int>(doubleNumber);

std::cout << "Явное преобразование (конверсия):" << std::endl;

std::cout << "doubleNumber: " << doubleNumber << std::endl;

std::cout << "intNumber (после преобразования): " << intNumber << std::endl << std::endl;

// Неявное преобразование (автоматическая конверсия)

int a = 5;

double b = a; // Неявное преобразование int в double

std::cout << "Неявное преобразование (автоматическая конверсия):" << std::endl;

std::cout << "a (int): " << a << std::endl;

std::cout << "b (double, после неявного преобразования): " << b << std::endl << std::endl;

// Преобразование char в int

char c = 'A';

int asciiValue = static\_cast<int>(c);

std::cout << "Преобразование char в int:" << std::endl;

std::cout << "c (char): " << c << std::endl;

std::cout << "asciiValue (int, после преобразования): " << asciiValue << std::endl;

return 0;

}