1. **Преобразование типов аргументов. Аргументы по умолчанию. Встраиваемые функции. Отладка функций. Перегрузка функций.**

**Преобразование типов аргументов -** когда мы передаем аргументы в функцию, компилятор определяет, какой тип аргумента лучше всего соответствует типу параметра функции.

Возможные результаты:

1. Точное соответствие.
2. Соответствие с преобразованием типа.
3. Отсутствие соответствия.

**Аргументы по умолчанию** позволяют определить значения, которые будут использованы, если при вызове функции не указаны значения для этих аргументов.

Но! Аргументы по умолчанию передаются только после обычных аргументов.



**Встраиваемые функции** - ключевое слово inline, используемое в простых функциях. Если при вызове обычной функции выделяется пространство в стеке для ее выполнения, то встроенная функция выполняется в коде так же, как если бы это функции не было, а все команды исполнялись бы прямо в месте вызова.

**Отладка функций** - тут хз что конкретно.

* Либо просмотр в компиляторе последовательность атомарных действий последовательно по мере исполнения кода.
* Либо создание контрольных точек, в которых либо с помощью cout выводить ключевые значения.
* Либо использовать условную компиляцию:
  + #'ifdef DEBUG (или другой вариант: #'if DEBUG > 4 - или другое логическое выражение)
    - То, что находится здесь будет компилироваться только в том случае, если существует DEBUG, если он определен ранее с помощью #'define DEBUG. Но если его закоментировать, компилироваться не будет и может служить чисто для тестирования.
      * #'else (или другой вариант #'elif DEBUG > 4)
        + Здесь выполнится код, если нет определенного макроса DEBUG.
    - #'endif DEBUG

**Перегрузка функций** позволяет определить несколько функций с одним и тем же именем, но с разными параметрами.

Компилятор выбирает соответствующую версию функции на основе типов аргументов, переданных в вызове функции.

