Он не нужен, потому что в нём не требуется высвобождать никакие ресурсы.

Ссылка на поток возвращается для того, чтобы можно было рекурсивно всё вывести в консоль (cout << a << b << c << … << - чтобы работала данная запись)

Параметр 'c' передается как константная ссылка, потому что при передаче объекта класса по значению будет создана его копия, что может быть излишним и неэффективным, особенно для больших объектов. Передача по ссылке позволяет избежать создания копии объекта, что экономит память и повышает производительность. Использование константной ссылки гарантирует, что объект c не будет изменен внутри функции.

Квалификатор const служит для указания того, что данный оператор является const-методом. Это означает, что при вызове этого оператора он не будет изменять состояние текущего объекта Complex, на котором он вызывается. Таким образом, использование const после скобок обеспечивает гарантию, что выполнение данного оператора не приведет к изменению состояния объекта, на котором он применяется.

Inline-функции - это функции, которые компилятор может решить вставить непосредственно в место их вызова в коде, вместо того чтобы вызывать их в обычном режиме. Они используются для увеличения производительности за счет сокращения накладных расходов на вызов функции. Ключевое слово inline подсказывает компилятору, что функция должна быть вставлена в каждое место ее вызова. Это может привести к увеличению размера кода, но может улучшить производительность при вызове небольших функций.

Динамические объекты и массивы объектов в C++ создаются с помощью операторов new и delete. Оператор new выделяет память для объекта или массива объектов в куче и возвращает указатель на эту память.

После использования динамически выделенной памяти необходимо освободить её с помощью оператора delete. Это важно, чтобы избежать утечек памяти.

Уничтожение динамических объектов происходит с помощью оператора delete. Когда объект уничтожается с помощью delete, вызывается деструктор объекта (если он определен), а выделенная для него память освобождается. При удалении массива объектов с помощью delete[], вызывается деструктор для каждого элемента массива в обратном порядке, а затем освобождается память, выделенная для всего массива. Утечки памяти могут возникнуть, если забыть освободить динамически выделенную память с помощью delete после завершения использования объекта или массива.