1. Перечислите все специальные функции-члены класса, включая перемещающие операции

Конструктор по умолчанию

Пользовательские конструкторы (для символа и для строки)

Деструктор

Операторы ввода, вывода, арифметические, логические

Копирующий и перемещающий операторы

Копирующий и перемещающий (copy/move) конструкторы

1. Приведите примеры операторов, которые можно, нельзя и не рекомендуется перегружать

-> [ ] ( ) + - \* / << >> = == и т.д. можно перегружать

& , | | не рекомендуется перегружать

:: . .\* нельзя перегружать

1. О каких преобразованиях следует помнить при проектировании операторов?

* Когда мы можем определять его как функцию член, а когда не можем
* Нужно обращать внимание на возвращающее значение
* Также на то, нужна нам rvalue или lvalue ссылка в аргументах оператора
* Порядок следования аргументов. Иногда он также должен быть согласован с введенными операторами преобразования типов

1. Опишите классификацию выражений на основе  перемещаемости и идентифицируемости

Lvalue – левый операнд (может быть переменная), идентифицируемый, но не перемещаемый, то есть может иметь адрес.

Rvalue – правый операнд (число 10, временная переменная, результат какой-либо функции), только перемещаемый.

Xvalue – и идентифицируемый и перемещаемый

Gvalue – только идентифицируемый

Prvalue - перемещаемый, но не идентифицируемый (например число)

1. Зачем нужны rvalue-ссылки?

Они позволяют осуществлять перемещение без копирования. Точно также продлевают время жизнь временным переменным и результатам функций.

rvalue ссылка ведет себя точно так же, как и lvalue ссылка, за исключением того, что она может быть связана с временным объектом, тогда как lvalue связать с временным (не константным) объектом нельзя.

Например:

A& a\_new = A( ); // ошибка

A&& a\_new = A( ); // работает

1. Почему семантика перемещения лучше копирования?

Перемещение более эффективно из-за того, что идет просто перенос адреса элемента в памяти, перемещается указатель на область памяти из одной переменной в другую. А при копировании (V1 копируем в V2) отдельно выделяется новая область памяти, на которую будет указывать V2, туда копируется всё, что находится в области под V1, если объекты большие, то это очень затратно по времени и по памяти.

1. Что делает функция std::move и когда нет необходимости явно ее вызывать?

Std::move преобразует тип переменной к типу rvalue, для того чтобы мы могли её перемещать.

Когда необходимо вернуть переменную определенную внутри функции и она имеет тот же тип, что и возвращаемый функцией, компилятор может сам преобразовать её к типу rvalue, но она не должна являться аргументом функции.

1. Кем выполняется непосредственная работа по перемещению?

Специальный перемещающий оператор или конструктор. Например, оператор Operator=( Example& par, Example&& par)

1. Когда может потребоваться пользовательская реализация специальных функций-членов класса?

Когда нет уверенности, что компилятор хорошо сможет сделать это сам. Например, когда нам нужно сделать глубокое копирование, а он делает поверхностное.

1. Для чего нужны ключевые слова default и delete в объявлении специальных функций-членов класса?

Default – специальная функция будет создана компилятором

Delete – этой функции не будет вообще. Её не создаёт ни пользователь, ни компилятор