1. Что обеспечивает идеальная передача и как она реализуется?

Идеальная передача дает возможность писать универсальные функции одновременно для разных видов ссылок у аргументов (это особенно важно, когда их [аргументов] несколько и различных сочетаний очень много). Реализуется за счет универсальных ссылок и функции std::forward, которая превращает lvalue-ссылку в rvalue, если в результате инстанцирования в функции на месте пробрасывающей принимается rvalue-ссылка.

2. Какая ссылка называется пробрасывающей или универсальной?

Пробрасывающей называется ссылка auto&& или T&& , если T — тип из шаблона, то есть в тех случаях, когда тип перед && вычисляется компилятором. При инстанцировании становится ссылкой любого вида, свойство rvalue внутри функции при необходимости восстанавливается вызовом std::forward.

3. В чем заключается идиома SFINAE применительно к шаблонам?

Идиома SFINAE предусматривает проверку совместимости шаблона с принимаемым типом, и если результат отрицательный, то вместо инстаницирования наиболее подходящей по сигнатуре, но неправильно работающей функции и ошибки компиляции продолжается поиск других вариантов этой перегруженной функции.

4. Как можно использовать вспомогательный шаблон enable_if?

Для проверки некоторого условия и инстанцирования с некоторым заданным типом только при его [условия] выполнении путем:
1) Упаковки типа возвращаемого значения функции (отсутствие типа — некорректная запись, а иначе тип получим из enable_if);
2) Добавления аргумента в функцию со значением по умолчанию (благодаря этому значению при невыполнении условия не получится инстанцироваться, потому что само оно есть, но типа у него нет, а иначе функция принимает какой-то аргумент с типом из enable_if, возможно, не используемый далее);
3) Добавления шаблона с типом-аргументом по умолчанию, упакованного enable_if (при невыполнении условия снова будет некорректная запись и ошибка при попытке инстанцирования);
4) Упаковки нешаблонного аргумента функции (аналогично (2), только в случае удачного прохождения проверки тип принятого аргумента является не шаблонным, а каким-то конкретным).

5. Какие правила вывода применяются при работе с шаблонами?

1) Если функция принимает T&, то тип T остается таким же, какого типа аргумент передан в функцию (с сохранением константности), и к нему прибаляется &, при этом если передана ссылка, то & & сворачивается до &.
2) Если функция принимает пробрасывающую ссылку T&&, то для lvalue-объектов T выводится как (переданный тип& (с сохранением константности)), и инстанцирование происходит для lvalue-ссылки (после сворачивания & && до &). Для rvalue-объектов T выводится как (переданный тип), в случае xvalue дополнительно происходит сворачивание && && до && и инстанцирование происходит для rvalue-ссылки.