1. Конструктор – это особый тип метода класса, который автоматически вызывается при создании объекта этого же класса (также его можно написать самостоятельно). Конструктор помогает настроить способ инициализации элементов класса или вызвать функции при создании объекта класса
2. Всего существует три типа конструкторов: 1) конструктор с параметрами; 2) конструктор без параметров; 3) конструктор копирования
3. Деструктор - это особый метод, необходимый для освобождения ресурсов, которые конструктор выделил объекту. Описать деструктор явным образом нужно в случае, если объект содержит указатели на память, выделяемую динамически, иначе при уничтожении объекта память, на которую ссылались его поля-указатели, не будет помечена, как свободная
4. Конструктор с параметрами используется для инициализации объекта требуемыми значениями; конструктор без параметров используется для создания «пустого» объекта; конструктор копирования используется для создания объекта, аналогичного тому, который уже существует
5. Конструктор копирования вызывается, когда новый объект создается путем копирования существующего: 1) при описании нового объекта с инициализацией другим объектом; 2) при передаче объекта в функцию по значению; 3) при возврате объекта из функции
6. Конструктор не возвращает значение. Нельзя получить указатель на конструктор. Класс может иметь несколько конструкторов с разными параметрами для разных видов инициализации. Конструктор, вызываемый без параметров, является конструктором по умолчанию. Тип параметров конструктора может быть любым за исключением этого же класса. Конструкторы не наследуются, их нельзя описывать с модификаторами const, virtual, static. Конструкторы глобальных объектов вызываются до вызова функции main
7. Деструктор не имеет аргументов и возвращаемого значения. Деструктор не наследуется. Деструктор не может быть объявлен как const или static. Деструктор может быть виртуальным
8. Р
9. Указатель this - это указатель, который хранит адрес конкретного объекта класса. Он присутствует в виде скрытого первого параметра в каждом методе класса (кроме статических методов). Типом этого указателя является имя класса. В методах класса, при необходимости, можно использовать this явно.
10. Методы, определенные внутри класса, всегда inline, поэтому им не обязательно явно прописывать inline. Для выносимых за пределы класса функций в случае необходимости inline писать обязательно
11. Конструктор не возвращает значения
12. Конструктор по умолчанию
13. void
14. Конструктор без параметров
15. Конструктор без параметров
16. Конструктор с параметрами

student(string surname, int age)

{

}

1. Конструктор с параметрами

student(string surname, int age)

{

}

1. Конструктор с параметрами

student(string surname, int age)

{

}

1. student()

{

void show()

{

cout << “” << endl;

}

}

1. p.set\_name(“Alexey”);