1 - Тип данных - объявление переменных, которое влияет на то, какие значения эта переменная может иметь в будущем. Также определяет место, занимаемое переменной в памяти.

2 - Класс определяет формат (описание) некоторых данных и работу (поведение) над этими данными. Из объявления класса можно получить различное количество объектов класса (переменных типа «класс»). Каждый объект класса определяется конкретным (на данный момент) значением внутренних данных (переменных), которое называется состоянием объекта.

3 – 1 – исходный код

2 – компилятор

3 – объектный код

4 – 1 – библиотечный код

4 – 2 – код запуска

5 – компоновщик

6 – исполняемый код

4 - Структурная декомпозиция – частный случай процедурной декомпозиции, при структурной декомпозиции запрещены «циклические» или обратные вызовы подпрограмм. Схема структурной декомпозиции иерархическая и напоминает дерево, перевернутое корнем вверх. На верхнем уровне, в корне дерева, находится основная программа, которая вызывает подпрограммы следующего уровня, а эти подпрограммы, в свою очередь, вызывают только подпрограммы более низких уровней.

5 - Декомпозиция в ООП – разбиение всей задачи на совокупность классов и объектов, с соответствующими свойствами и методами.

6 – Формализация – метод исследования содержание объекта с помощью выявления закономерностей и связей между элементами его формы. В процессе формализации задача представляется в виде формальной системы. В этом методе форма абстрагируется от содержания. Такая модель позволяет исследовать структурные закономерности происходящих в ней процессов.

7 - В объектно-ориентированном программировании конструктор класса — специальный блок инструкций, вызываемый при создании объекта. С помощью конструктора можно при создании объекта инициализировать аргументы класса, заполняя их заданными значениями.

8 - Подпрограммы-функции, возвращающие результат (число, символьную строчку и т.д.), который можно использовать в основной программе. Виды – процедуры и функции.

9 – Параметры подпрограммы – аргументы, передаваемые подпрограмме для работы с конкретными значениями. Виды параметров - формальные и фактические. Способы передачи – по значению, по ссылке, по имени.

10 – Да, допускается.

11 - Статическая линковка запускается в момент создания исполняемого файла, в её результате из нескольких объектных файлов получается один исполняемый. Динамическая линковка запускается в момент создания процесса или после его создания. Она подключает нужные библиотеки динамически в процессе работы.

12 – Ассоциация, Наследование, Агрегация, Зависимость.

13 - Инкапсуляция предназначена для изоляции аргументов и методов абстракции (протокол/интерфейс) от их реализации. На практике это означает, что класс должен состоять из двух частей: интерфейса и реализации. В реализации C++ обеспечивается механизм сокрытия, позволяющий разграничивать доступ к различным частям класса.

14 - Наследование позволяет перенять одному классу ( дочернему ) аргументы и методы другого класса ( родительский ).

15 – Полиморфизм - это способность объекта использовать методы производного класса, который не существует на момент создания базового.