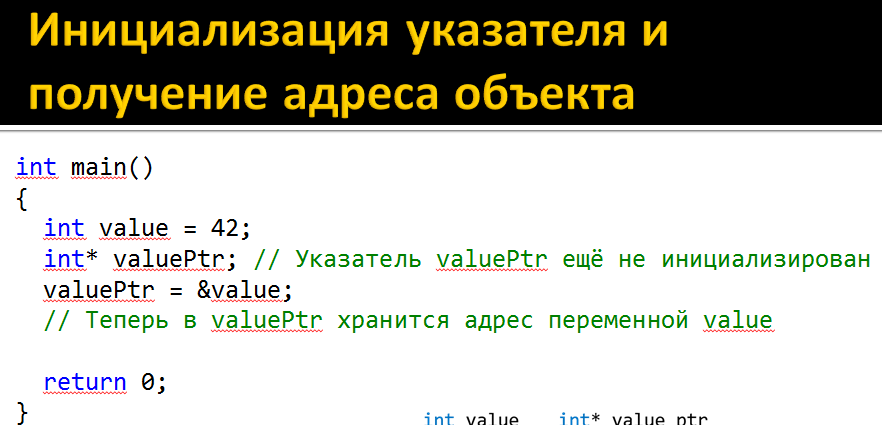
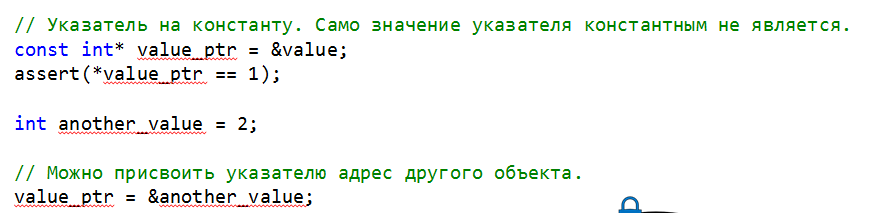
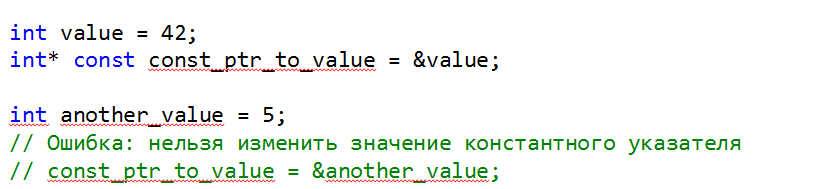
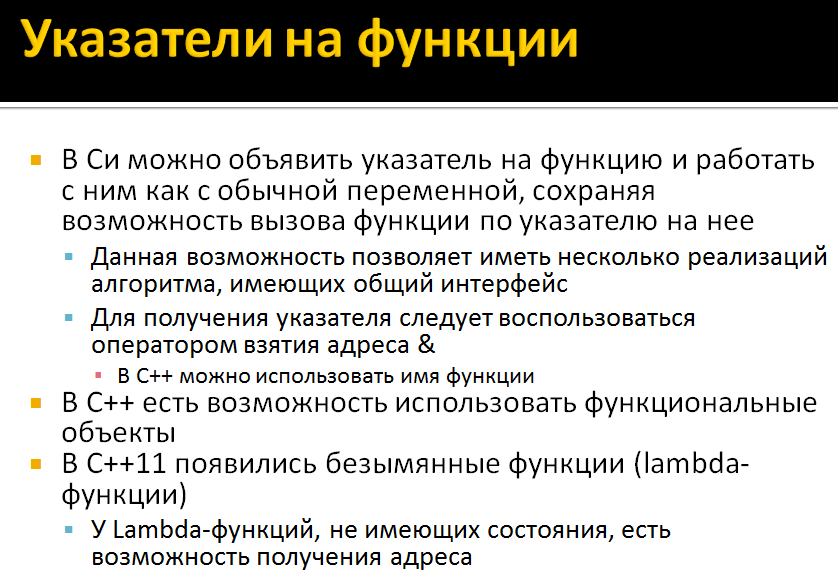
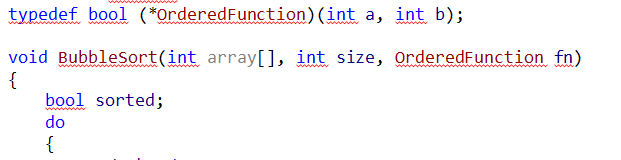
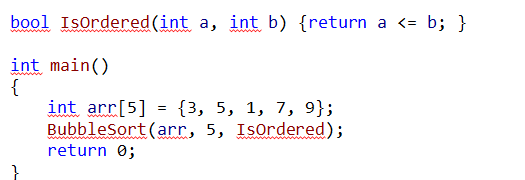
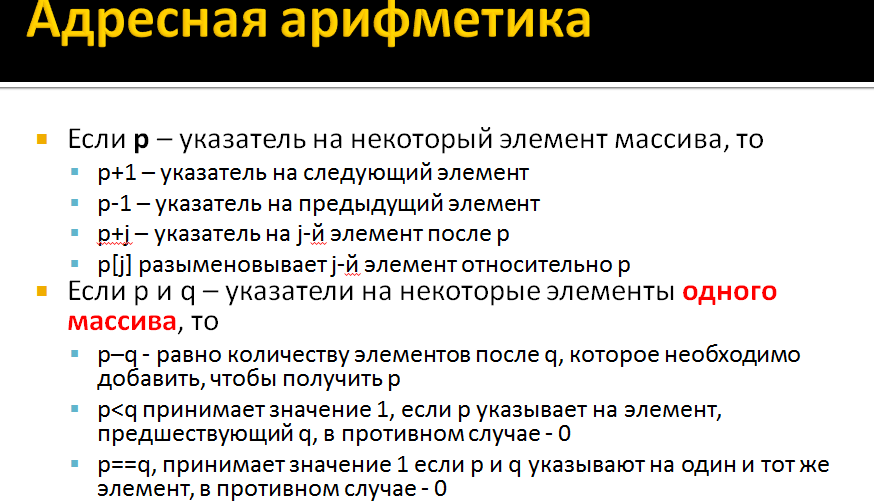
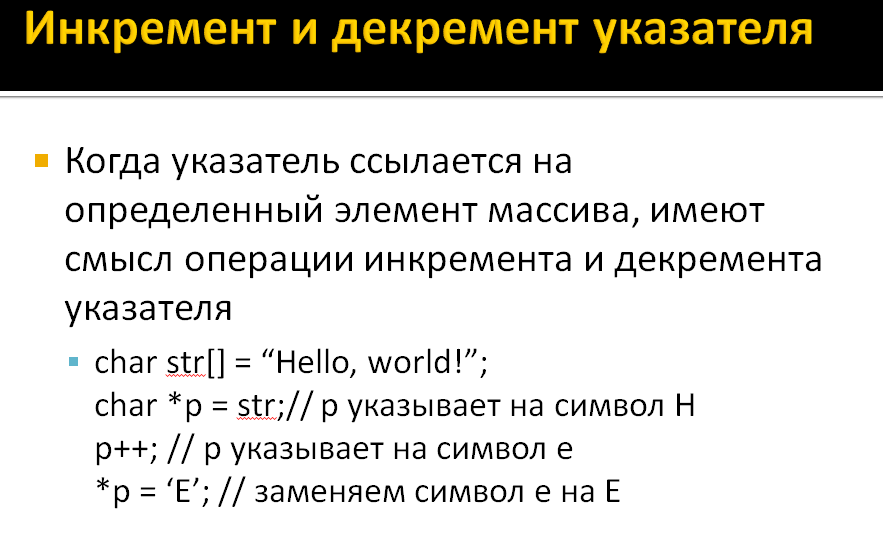
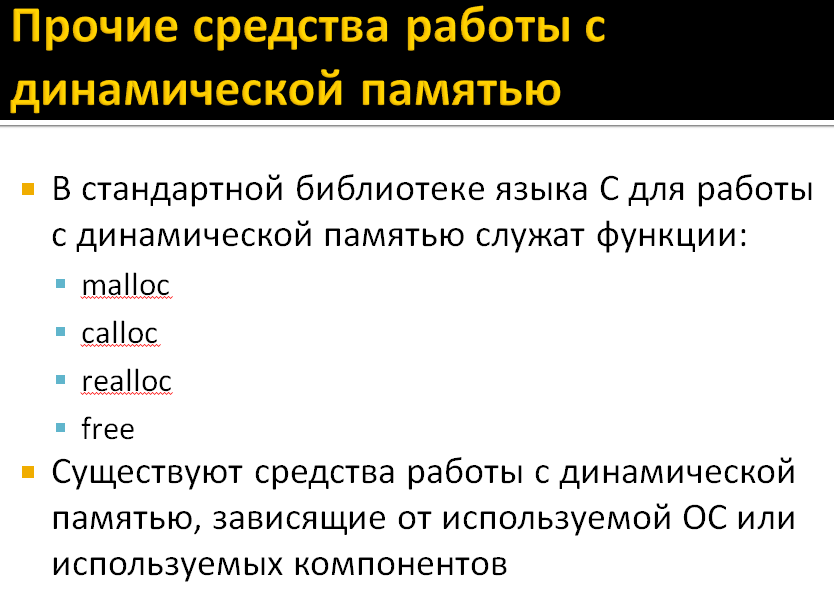
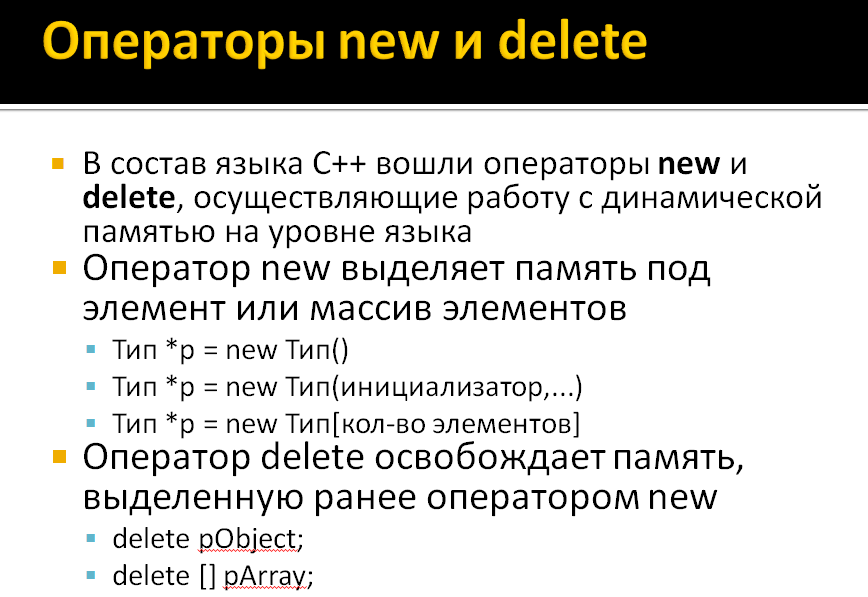
  
  
Указатели объявляются подобно обычным переменным, только с использованием символа «звёздочка» \* после типа. Например, так выглядит указатель, способный хранить адрес объекта типа int.

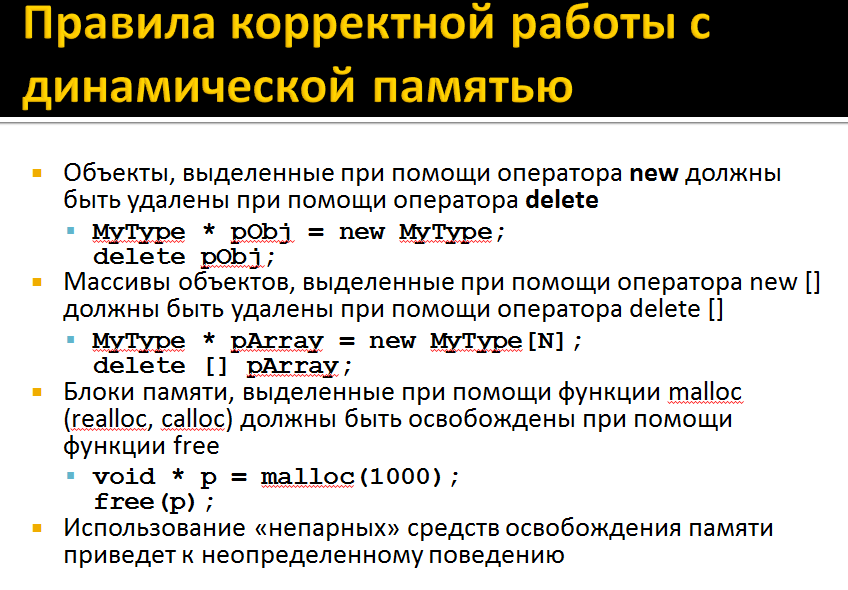
Переменная p может хранить адрес целого числа. Так как переменная-указатель p не инициализирована, использовать её для доступа к объекту нельзя. Объявление указателя выделяет память для хранения адреса на платформе, но не инициализирует эту область памяти.

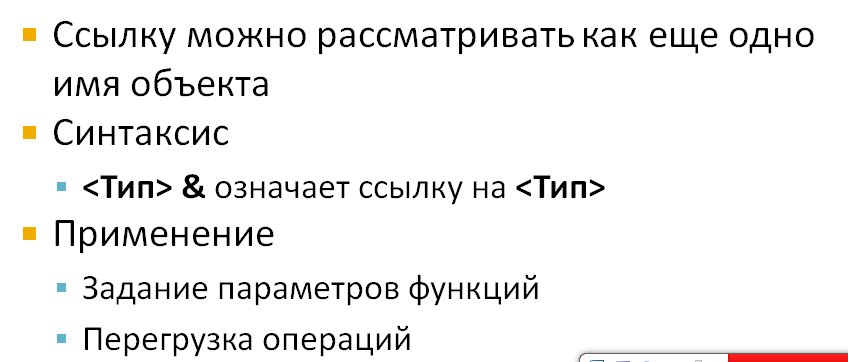
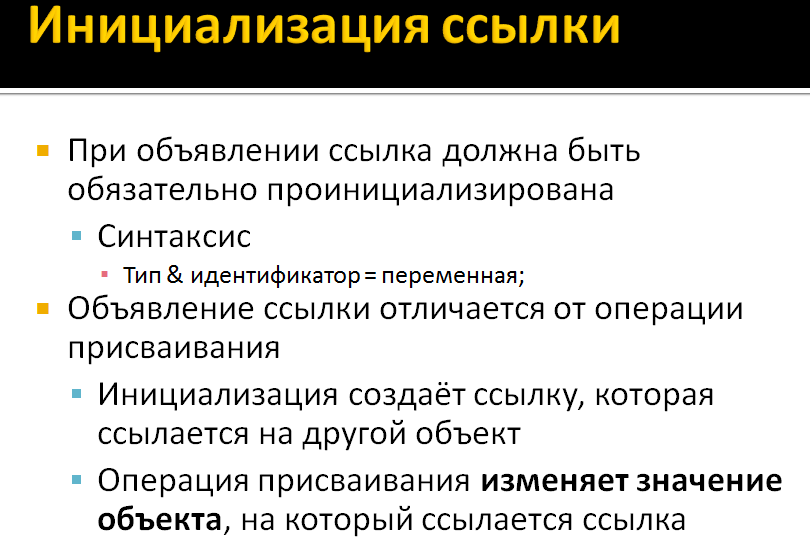
  
  
В C++ ссылки — не объекты. Они вводят новое имя для доступа к уже существующему объекту. Поэтому оператор &, применённый к ссылке, возвращает не указатель на ссылку, а указатель на сам объект.

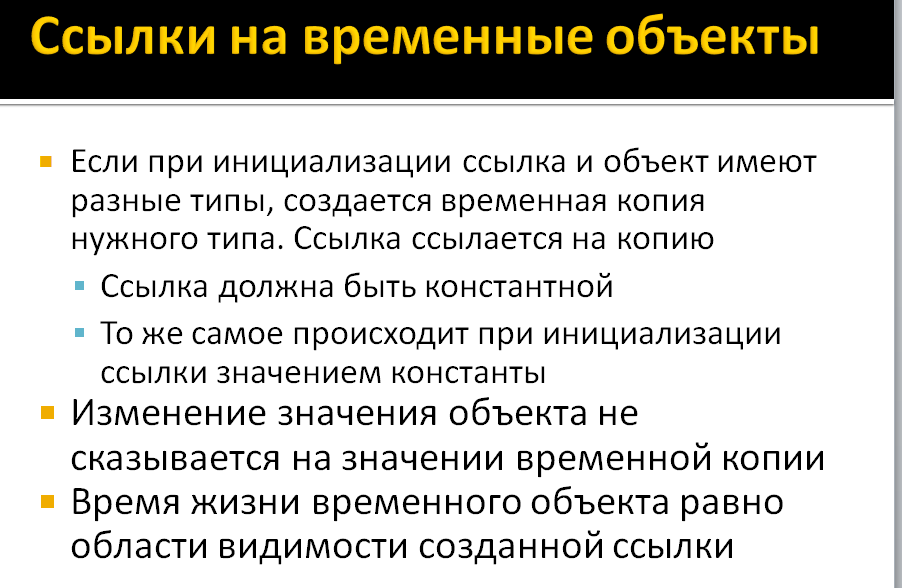
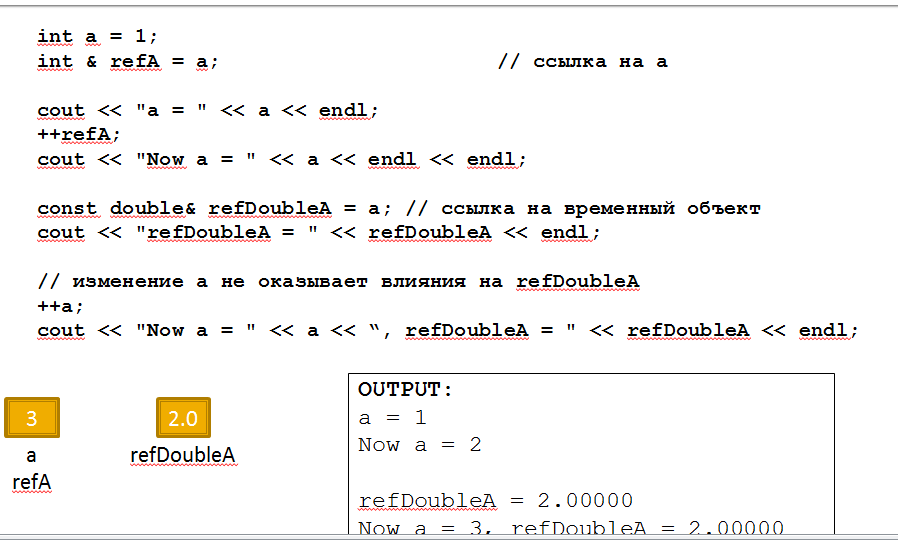
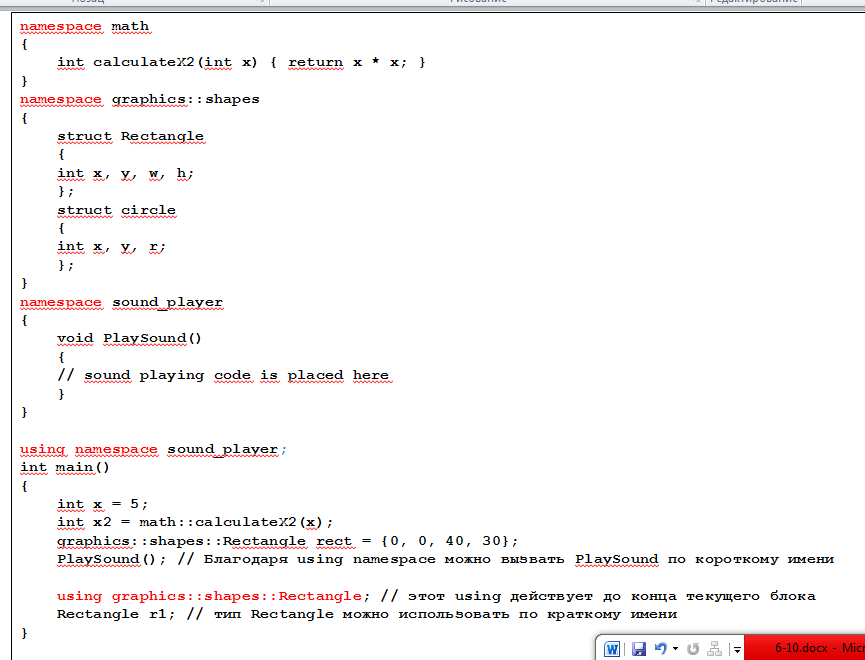
И переменная answer, и ссылка answer\_ref относятся к одному и тому же объекту. Поэтому взятие адреса ссылки равнозначно взятию адреса объекта.

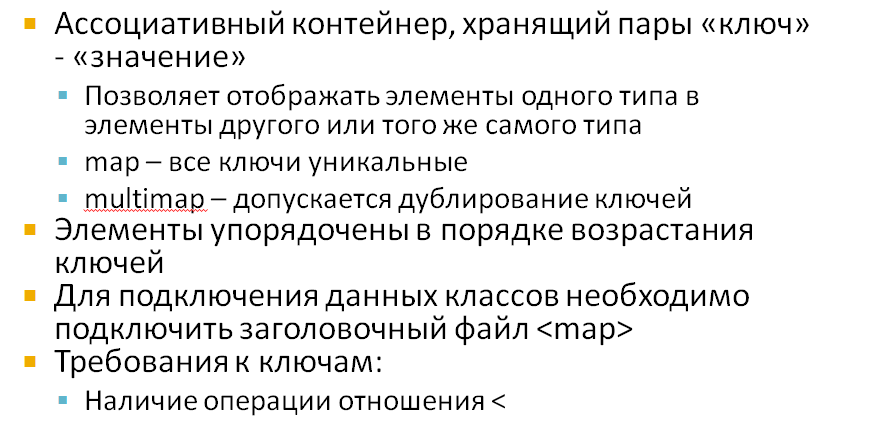
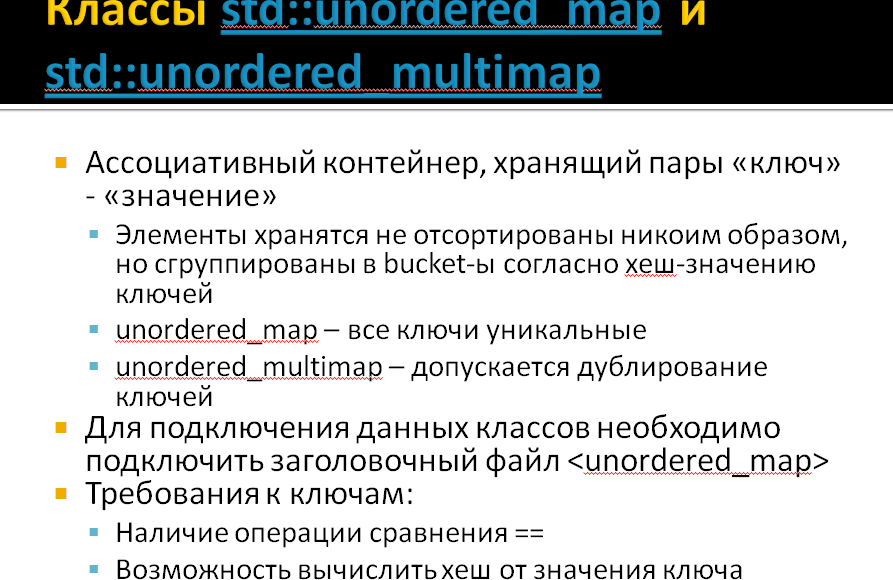
В отличие от ссылок, указатели могут в процессе жизни менять своё значение, храня в разные моменты времени адреса разных объектов. Простейший способ изменить значение указателя — присвоить ему адрес другого объекта:  
  
Указатель на константу.  
  
Константный указатель.  


  
  
Указатель на функцию.  
  
  
  
  
  
  
  
  




  
  
Ссылки.  
  
  
  
  


  
  
Deque – двусторонняя очередь.  
List – двусвязный список.  
  
  
  
  
  
  
  
  
Итераторы.  
