Задачи на второй полусеместр:

1. Прочитайте файл linkedList.cpp. Допустим, мы заполняем объект этого класса данными, а затем присваиваем один класс целиком другому, используя стандартный оператор присваивания:  
   list2 = list1;  
   Допустим, что впоследствии мы удалим объект класса list1. Можем ли мы все еще использовать list2 для доступа к введенным данным? Увы, нет, так как при удалении list1 все его ссылки были удалены. Единственное, что было известно объекту linklist про удаленный объект, это указатель на него. Но его удалили, указатель в list2 стал недееспособным, и все попытки получить доступ к данным приведут к получению мусора вместо данных, а в худшем случае — к зависанию программы. Один из способов избежать этих проблем — перегрузить оператор присваивания, чтобы он вместе с объектом копировал бы все его ссылки. Но тогда придется пройти по всей цепочке, поочередно копируя все ссылки. Следует также перегружать конструктор копирования. Чтобы была возможность удалять объекты linklist в main(), можно создавать их с помощью указателя и new. В таком случае проще будет проверять работу новых операций.
2. Для класса дроби, реализованного ранее, создайте пререгрузки операций +,-,\*,/, так чтобы компилятор верно трактовал записи вида fract1 = fract2 + fract3 и т.д. Создайте перегрузки операций, позволяющие прибавлять к дробям целые числа, так чтобы компилятор верно трактовал записи вида fract1 = fract2 + 5 и fract1 = 5 + fract2 и т.д. Для реализации второй перегрузки потребуется создать дружественную функцию. Реализуйте метод упрощения дополнительно при помощи дружественной функции.
3. Создайте базовый класс и два производных класса. Создайте функцию DisplayName(Base \*), которая принимает указатель на объект базового класса и выводит на консоль имя класса, который был передан по указателю. Создайте метод main, который позволит убедиться в работоспособности программы.
4. Создайте функцию swaps, обменивающую значения двух аргументов, посылаемых ей. Сделайте из функции шаблон, чтобы она могла использоваться с любыми числовыми типами данных (int, char, float и т. д.). Напишите main() для тестирования функции.
5. Прочитайте файл safeArray.cpp. Сделайте из класса safearray шаблон, так, чтобы он мог хранить любые типы данных. В секции main создайте по крайней мере два массива различных типов.
6. **(\*)** Класс дроби из упражнения 2 преобразуйте в шаблон, так, чтобы дробь могла хранить числитель и знаменатель различных типов. Упражнение вместо упражнения 7.
7. **!!(\*)** Преобразуйте класс vector, реализованный на практических занятиях к шаблону, так чтобы он мог хранить данные любого типа. Упражнение вместо упражнения 6.
8. !!В программу из пункта 5 добавьте класс исключений, чтобы индексы, выходящие за пределы массива приводили к генерации исключения. Сообщение об ошибке исключения должно содержать значение индекса, приведшее к сбою и действительный диапазон индексов массива.
9. Напишите программу, выводящую на консоль самое большое число Фибоначчи, которое может хранится на целевой машине и влазит в тип int.
10. В цикле запрашивайте у пользователя данные, состоящие из имени, отчества, фамилии и номера работника (типа unsigned long). Затем осуществите форматированный вывод в объект ofstream с помощью оператора вставки («). Не забывайте, что строки данных должны оканчиваться пробелами или другими разделителями. Когда пользователь сообщит об окончании ввода, закройте объект ofstream, откройте объект ifstream, прочитайте и выведите на экран все данные из файла, после чего завершите программу.
11. !!Основываясь на программе linkedList из упражнения 1, напишите программу, позволяющую пользователю выбрать одно из четырех действий нажатием соответствующей кнопки. Действия таковы:  
    1. добавить ссылку в список (от пользователя требуется ввести целое число)  
    2. показать данные по всем ссылкам из списка;  
    3. записать в файл данные для всех ссылок (создание или переписывание файла);  
    4. считать все данные из файла и создать новый список ссылок, куда и поместить их.  
    Первые два действия могут использовать методы, уже имеющиеся в LINKLIST. Требуется написать функции для чтения и записи файла. И для того, и для другого можно использовать один и тот же файл. В нем должны храниться только данные; нет никакого смысла хранить содержимое указателей, которые, возможно, уже не будут нужны во время чтения списка.
12. !!Для класса дроби перегрузите операторы извлечения (») и вставки («) в калькуляторе с четырьмя действиями. Имейте в виду, что операторы могут связываться в цепочки, поэтому при выполнении действий с дробями понадобится только одно выражение: cin »fracl » op »frac2;