CS201 – PROBLEM SOLVING & PROGRAMMING II
PROGRAM 5 (GROUP PROJECT)
PLANNING DOCUMENT 2 – DATA ACTIONS

NAME(S): Kenneth Kakie, Sylvia Tang

1. Think about those in the community you are helping and the data (classes) you have decided upon. Which data structure(s) (linked lists, queue, or stack) will be best to track this data? Explain this below:

TWO linked lists with client class, commission class
Helps People who need way to store Clients

2. What other actions (functions) that you will need to perform to complete the outcomes you are trying to achieve. Think about the data that needs to be available for the function (preconditions) and what output or data update will be performed as a result of your function (postconditions). List as many things as you can think of in the table below:

FUNCTIONS	**PRECONDITIONS** What data that needs to be available?	**POSTCONDITIONS** What is updated or produced from this function?
addNewCommission addCommission	commission ID	A new commission is added for the client
addClient	client list	Update client list with new client.
processRaw File	clientList commissionList infile (input).	calls process comm's & process client and push into vector accordingly.
save	clientList commisList	purges txt file & overwrites it with updated info.
mainHeader	none	prints available input options.
printMenu	clientList commList	prints menu
update	client list commList	calls all update functions. (ie update priority & update Price)
deleteCommission	client list Commission List	deletes a commission from the list.
updateActive Comms	None	updates the # of active comms to be the size of active comm vector.
Sort Price Sort Priority	sends in character input (y / n)	update menu to be sorted by price / priority.

3. Begin considering input & output from your program:
   a. What will the input look like (csv file? csv file & menu driven? menu-driven only?) The 'actions' to be taken on the data will help you decide & describe what needs to be included as part of your input.
   Write some sample data to be 'input' into your data structure of object types: Identifying character in front

   CSV storing raw data per comis. + client.

   csv for input file
   output updated csv for input next time (updated data)

   b. Consider how you will test your input, data, logic. What are some error checking opportunities you can add?
   if clients have same name, different ID,

   c. How will you verify that your program is accomplishing the intended goal.

   While loops that prevent leaving until suitable answer provided

   d. Group other functions – which are related or rely on another function to provide information? Ex: The basic class functions (accessors & mutators) and data structure functions (push, pop or enque, dequeue) need to be ready to go before any of the other functions are written. Reading a file or inputting from the keyboard will require logic to check for errors before adding the object to your data structure.

   Begin assigning these logical groupings to each group member. List these jobs below.

Kenneth	Sylvia
Client class	comis class
- print	- Print
	- Sort
	- filter
client LinkedList	comis L.L.

need to plan Scrums, [sprints]
[and] write it out physically
write who is responsible for what in todo list
Check Points

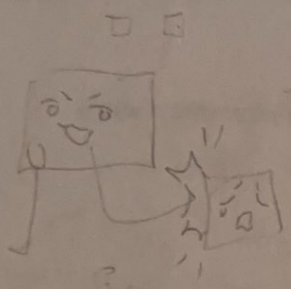
Dec Nov 20[illegible]

...use two linked lists, we need linked list to do

CS201 – PROBLEM SOLVING & PROGRAMMING II
PROGRAM 5 (GROUP PROJECT)
PLANNING DOCUMENT 1 – CLASS DATA

NAME(S): Kenneth Kaur, Sylvia Tang

1. Brainstorm as a team. Think about people in your community: what would be important or useful for them to track in a community organization or activity or to help your community.
For example, you could create a class to keep track of community events, club activities, athletic games or statistics, community leaders, performers or performances, health tracking, or another subject of your choice.
List ALL ideas discussed:



- Personal banking: keeping track of finances and physical cash/coins
- Pokemon: tracking cost of cards, creating sets, seeing how much each is worth
  rand().
- tracking clients & commissions: client class, comis class, diff prices per comis, diff discount rates per client
  Filter/prioritize VIP
  different sort options (alpha, comis. size, comis. cost, etc)

2. Choose the top 2 ideas as a team and list those below. Briefly describe why you think these 2 ideas would the best to implement.

Tracking clients/commissions.

Personal Banking

They're the most practical.

3. For your top choice, write a sentence (or two) about the intended goal of your program:
(ex: Our program will help those in my community to ..... We think this will be helpful because .....)

Tracking clients/commissions would be useful for artists and people who create things as their job to have a convenient way to track the many aspects of their job.

4. Brainstorm the data items you would need to track. Include the type of data for each (string, integer, etc.). Think about the relationships between the data. You will be expected to implement a base (parent) class and a linked list of objects of the base class.

class client

5. Organize this data into the UMLs for your classes. Include constructor, accessor, mutator information, copy constructors, destructors. It is understood that this may change for the final project.

Class Client
- vector<Commissions> ← Holds commission IDs
- string name
- double discount
- int numActiveComm
- int numTotalComm - int moSpent?
- int client ID
- ~~Pointer nextClient~~
+ void addComis() ← add to vector
+ void removeComis() ← removes from vector

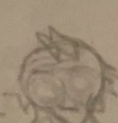
need a check, if client name isn't in List, then call constructor, add comis. to vector

sorts, to sort by client, progress, num comis, total comis, etc. time spent on it, price,

Also filter

in update progress

if progress = 100 call destructor & client Active--; if progress = -1, canceled comis so client Active--, Total comis--



Class Commissions
- int progress (0-99)
- int ID
- ~~pointer nextComis~~
- int clientID
- Double price
- Double TimeSpent (in hours)
- int Priority ← from calculation of price & client has spent
- bool isFinished — filter by false?
+ static int totalComisEver
+ static int totalActiveComis

From comis class comis object has field for client ID. Have secondary linked list for clients? Allows search, if client name isn't in client list, create new client obj.

By Nov 20th,

Kenneth: design basics of client class
Sylvia: design basics of commissions.

note, since two linked lists, we need linked list to be a parent of our classes? or, we need to design classes of linked list type? NO, data type is going to be of type comm3 or client, bc we declare in main "LinkedList1"

[would we have two separate code designs? NO using a template would be better. where the unique functions should be stored is under comisFunctions or clientFunctions.h

[How do you sort a linked list?
option 1: -create new list using createOrdered, which passes in other unordered list [if created in sound, will be deleted after printed so, have to sort every time]
option 2: - Create temp list, then sort temp, then overwrite original

↓ solved by [2]

[How do we structure raw input file? (two formats) CSV

1: "client", name, clientID, discount, numActiveComis, numTotalComis, comisID1, comisID2, ...
0 1 2 3 4 5
extra fields to be put into vector (variable amount of extra fields)

2: "commission", comisID, clientID, price, time spent, priority, isFinished, progress
any type identifier 2 1 5 6 3 4 7
int

- one processRawFile() function required, with if statement.
  If typeID == "client" : <Kenneth writes this>
  else if typeID == "commission" : <Sylvid writes this>
  ] These will be consolidated when forming actual code

PRE: It pulls in the whole line & takes first word. Passes remaining line through to correct place.

Need child class of linked for client & com

- Why not just one class per insted, no need to overcomplicate it with templates

When creating new comis, need to add it to client. If you know clientID, pass it in. Otherwise, add to newest client?

Sort in _ way
I call general print

Filter: print if _ field = inputted filter

Nov 20th: make a write to [save] our raw input file function
expound delete node functs [search for target node, del]
Need to delete ID in client class when comis is finished

Write print func

By nov 20:

Kenneth: creates read file cpp, creates template Linked list
creates client class

Sylvia: creates comics class, creates sorts, creates filters

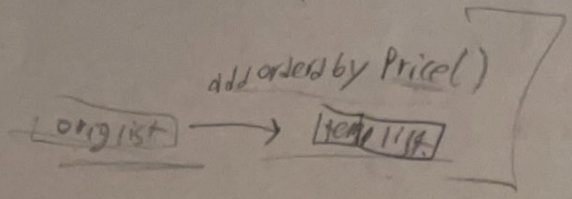
By nov 27:

Kenneth: update functions, prints

Sylvia: error checks

Draw the two lists & how they connect (Primary Key)

printPrice() {



~~Overwrite orig with temp list~~

}

[cout x.

[int x = 5

linkedCommission X;

X.printPrice() ← Prints sorted. Sorting is temporary

Write Print statements [Print all, Print Sorted]
Write delete Node
Calculating Priority

int x = "a"

CommisID
01 - A
02 -
03 -

- beef - shellfish
- chicken
- pork <-fish

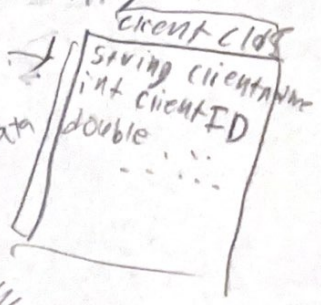
X.printPrice()

~~NOde<int>~~

~~data = 0;~~

NOde<client> D

data is a clientClass



main {

LinkedX;] const. called

data is the equivalent of + here

3] decon called

main

function(int x);

↓

function(D x);

template<typename D>

classA{

+
+
-
-

}

Node<D>

class<D>

.h] class B: public A{

.h] class C: public B{ }

;

wordList[i].word

wordstruct X;

X.word