Vector Data

* Load course file
  + Open file
  + If return value is less than 1
    - Error file cannot be found
  + Else file is found
* Read each line
  + If there are less than 2 value in line
    - Return error
  + Else read parameter
    - If there are 3 or more parameter
      * Return
* Close file
* Course information
  + Create course structure
    - Create Course Id, Course Name and Prerequisite
      * Split lines
  + Create new course object
    - Course Number and Name
  + Each prerequisite
    - Add to course object prerequisites list
* Insert Course
  + Add course to vector data structure
  + Print course information – course Name
    - If course number equal the course number
      * Print course information
  + Close
* HashTable
* Open file
  + If vale is -1
    - File not found
      * Print error
  + Else
    - Read line
      * If there is less than 2 value in
        + Return error
  + Else
    - Read parameter
      * If there are 3 or more values parameter
      * If the 3rd or more parameter is first then it will continue
  + Else
    - Return
* Close file
  + Create hashtable class
  + Create method
    - Create a loop
  + Each line in the file of The 1st and 2nd value
    - Temp item will be created to hold the values
  + Else
    - 3rd value exist – add to current value
      * Insert method for each value
* Search/Print
  + Users ask for input
  + Assign input for key
    - If key is found
      * Print course information
    - Prerequisite course
      * Print prerequisite course information
* Close
* BinaryTree
* Open file by using fstream
  + If file open then continue
    - Else
      * Print Error
* Read each line of file
  + If less then two value (int/string) in line
    - Return error
      * Else if
        + Third or more then first line with prerequisite
      * Else
        + Continue to read file
        + Print all course at end of line

Create Course object

* if 1st and 2nd line are value
  + Add course Id and Course name
* If 3rd or more lines exist
  + Add prerequisites
* Create root that points to null
  + If root is null
    - Add course null
      * Else
        + Course number is less than leaf node

Add to left

* + - * + If course number is greater than

Add to right

* + - * Else
        + If course number is greater than root

Add to right

* + - Print
* If root is not null
  + Move left output
    - Else
    - Move right output
* close

Pseudocode Menu

* Create value
* create user prompt to enter option
  + std: cout<<
    - << “Enter Choice:”<< endl
    - << 1. Load Data. << endl
    - << 2. Print alphanumeric list all Courses <, endl
    - << 3. Print Course Title/Prerquisites << endl
    - <<9.Exit”<< endl
* Menu loop will continue until user exit
* Switch statement base on user chose
* Swich(choice)
  + Case 1. Loaddata
  + Case 2. PrintcourseList
    - SortList(course)
    - PrintList(course)
      * break
  + Case 3.
    - Print Course(courseId
      * break
  + Case 9.
    - Output “Goodbye”
      * Break
* hashTable
  + print Course information
  + Course identified
  + Request course ID
* Key is equal to course ID
  + If current node is same
    - Return course
    - Display course
  + Else
    - Check node
  + If null point to node
    - Return
  + If key equal course If
    - Return
    - Display course
* Tree

Reading file

* Open file by using fstream
  + If file open then continue
    - Else
      * Print Error
* Read each line of file
  + If less then two value (int/string) in line
    - Return error
      * Else if
        + Third or more then first line with prerequisite
      * Else
        + Continue to read file
        + Print all course at end of line
* Print out CourseList
  + SortList
  + Char = 50
  + Create char array
    - Sort
  + While char <50
    - Char = 1
    - If I <0; I<50
      * ++i
    - Else
      * Char = 1
    - Return

Vector Data:

* The run time for vector is O(n).
* Advantages
  + Simple
  + Fast access
  + Order
* Disadvantages
  + Not that fast

HashTable

* Advantages
  + Fast
  + Can hold large datasets
  + Direct access
  + The run time for hash table is O(n) and the run time for binary tree is O(n).
* Disadvantages
  + No order

Binary tree

* Advantages
  + Sort order
  + Search and retrieval
  + Run time is O(n)
* Disadvantages
  + Slower
  + complex

Based on the information would recommend the vector, because it is simple and fast access. The vector would be able to sort fast and print.