Pseudocode for Vector Data Structure

**Open File And Read Data**

Function Openfile(Filepath)

Declare File

Open Filepath As File

If File Is Null

Print "Error: File Not Found."

Return Null

Endif

Return File

End Function

**Parse File Data**

Function Readfile(File)

Declare Courseslist As List

Declare Line

While Not End Of File

Read Line From File

Parse Line Into Coursedata

If Not Validateline(Coursedata)

Print "Error: Invalid File Format On Line" + Line

Return Null

Endif

Append Coursedata To Courseslist

End While

Return Courseslist

End Function

Function Validateline(Line)

Declare Coursedata As List

Split Line By Comma Into Coursedata

If Length Of Coursedata < 2

Return False

Endif

Return True

End Function

**Create Course Object**

Struct Course

Coursenumber As String

Name As String

Prerequisites As List

End Struct

Function Createcourse(Coursedata)

Declare Course As Course

Set Course.Coursenumber To Coursedata[0]

Set Course.Name To Coursedata[1]

Declare Prereqlist As List

For I From 2 To Length Of Coursedata

Append Coursedata[I] To Prereqlist

End For

Set Course.Prerequisites To Prereqlist

Return Course

End Function

**Store Course In Bst**

Function Insertcourseintobst(Tree, Course)

If Tree Is Null

Set Tree To Course

Else If Course.Coursenumber < Tree.Coursenumber

Call Insertcourseintobst(Tree.Left, Course)

Else

Call Insertcourseintobst(Tree.Right, Course)

Endif

End Function

Function Buildcoursetree(Courseslist)

Declare Bst As Tree

For Each Coursedata In Courseslist

Declare Course As Course

Set Course To Createcourse(Coursedata)

Call Insertcourseintobst(Bst, Course)

End For

Return Bst

End Function

**Print Course Information Pseudocode**

Function Printcourseinfo(Tree)

If Tree Is Not Null

Call Printcourseinfo(Tree.Left)

Print "Course Number: " + Tree.Coursenumber

Print "Course Name: " + Tree.Name

If Length Of Tree.Prerequisites > 0

Print "Prerequisites: " + Join Tree.Prerequisites With ", "

Else

Print "Prerequisites: None"

Endif

Call Printcourseinfo(Tree.Right)

Endif

End Function

Main

Function Main()

Declare Filepath As String

Set Filepath To "Path\_To\_Your\_Course\_File.Txt"

Declare File

Set File To Openfile(Filepath)

If File Is Null

Print "Error Opening File."

Return

Endif

Declare Courseslist

Set Courseslist To Readfile(File)

If Courseslist Is Null

Print "Error Reading File."

Return

Endif

Declare Coursetree

Set Coursetree To Buildcoursetree(Courseslist)

Print "Course Information:"

Call Printcourseinfo(Coursetree)

End Function

**Menu Options**

Function Displaymenu()

Print "Select an option:"

Print "1. Load file data into the data structure"

Print "2. Print all courses in alphanumeric order"

Print "3. Print course title and prerequisites"

Print "9. Exit"

Declare Choice As Integer

Input Choice

Return Choice

End Function

Function Main()

Declare Filepath As String

Set Filepath To "Path\_To\_Your\_Course\_File.Txt"

Declare File

Declare Courseslist

Declare Coursetree

Declare Choice As Integer

While True

Set Choice To Displaymenu()

If Choice = 1

Set File To Openfile(Filepath)

If File Is Null

Print "Error Opening File."

Return

Endif

Set Courseslist To Readfile(File)

If Courseslist Is Null

Print "Error Reading File."

Return

Endif

Set Coursetree To Buildcoursetree(Courseslist)

Print "Data successfully loaded into the structure."

Else If Choice = 2

If Coursetree Is Null

Print "Please load the file data first using option 1."

Else

Print "All Courses In Alphanumeric Order:"

Call Printcourseinfo(Coursetree)

Endif

Else If Choice = 3

If Coursetree Is Null

Print "Please load the file data first using option 1."

Else

Declare Coursenumber As String

Print "Enter the course number:"

Input Coursenumber

Call Printspecificcourseinfo(Coursetree, Coursenumber)

Endif

Else If Choice = 9

Print "Exiting the program."

Return

Else

Print "Invalid choice. Please select a valid option."

Endif

End While

End Function

**Print Specific Course Information**

Function Printspecificcourseinfo(Tree, Coursenumber)

If Tree Is Not Null

If Tree.Coursenumber = Coursenumber

Print "Course Number: " + Tree.Coursenumber

Print "Course Name: " + Tree.Name

If Length Of Tree.Prerequisites > 0

Print "Prerequisites: " + Join Tree.Prerequisites With ", "

Else

Print "Prerequisites: None"

Endif

Return

Else If Coursenumber < Tree.Coursenumber

Call Printspecificcourseinfo(Tree.Left, Coursenumber)

Else

Call Printspecificcourseinfo(Tree.Right, Coursenumber)

Endif

Else

Print "Course not found."

Endif

End Function

| **Operation** | **Vector** | **Hash Table** | **Binary Search Tree** |
| --- | --- | --- | --- |
| **File Read and Course Creation** | O(n) | O(n) | O(n log n) |
| **Searching for a Course** | O(n) | O(1) | O(log n) |
| **Sorting Courses** | O(n log n) | O(n log n | O(n |
| **Memory Usage** | Moderate to High | High | Moderate |

**Vector**

Advantages: Simple to implement, good for ordered data.

Disadvantages: Slow search times, especially for large datasets.

**Hash Table**

Advantages: Fast search times, efficient for large datasets.

Disadvantages: High memory usage, poor performance in worst-case scenarios.

**Binary Search Tree**

Advantages: Balanced performance for search, insertion, and deletion operations.

Disadvantages: Requires careful implementation to maintain balance.