Montana Bazarragchaa

CMSC 204

```
Class CourseDBElement implements Comparable<CourseDBElement>:
    crn: int                // Course Registration Number (unique)
    courseID: String        // Course ID
    credits: int            // Number of credits
    room: String            // Room number
    instructor: String      // Instructor name

    Constructor(courseID, crn, credits, room, instructor):
        Set attributes

    Method hashCode() -> int:
        return String.valueOf(crn).hashCode()

    Method compareTo(other: CourseDBElement) -> int:
        return this.crn - other.crn

    Method equals(other: Object) -> boolean:
        return true if all fields match

    Method toString() -> String:
        return formatted string with all course info

Class CourseDBStructure implements CourseDBStructureInterface:
    hashTable: Array of LinkedLists<CourseDBElement>   // Hash table with chaining
    tableSize: int                                      // Size of the table
```

```
Constructor(n: int):                        // n is estimated number of courses
    tableSize = next 4K+3 prime after n / 1.5
    initialize hashTable of size tableSize

Constructor("Testing", size: int):              // For testing purposes
    tableSize = size
    initialize hashTable

Method add(element: CourseDBElement):
    index = element.hashCode() % tableSize
    if hashTable[index] is null:
        create new LinkedList
    add element to list if not already present

Method get(crn: int) -> CourseDBElement:
    index = hash of crn % tableSize
    search list at index for element with matching crn
    if found, return it
    else, throw IOException

Method showAll() -> ArrayList<String>:
    create list
    for each linked list in hashTable:
        for each element:
            add element.toString() to list
    return list
```

```
Class CourseDBManager implements CourseDBManagerInterface:
    cds: CourseDBStructure       // Main data structure

    Constructor():
        initialize cds with estimated size

    Method add(courseID, crn, credits, room, instructor):
        create new CourseDBElement
        cds.add(element)

    Method get(crn: int) -> CourseDBElement:
        return cds.get(crn)

    Method readFile(file: File):
        open file
        for each line:
            parse courseID, crn, credits, room, instructor
            call add method
        handle IOException if file not found

    Method showAll() -> ArrayList<String>:
        return cds.showAll()
```