Diver Class

public static void main(String[] args)

1. Short Description: Takes input from user and outputs Students and courses

2.Parameter Description:

Student: Instance of student Class

Course: Instance of Course class

Courselist: Instance of CoureCollection Class

Studentlist: Instance of StudentCollection Class

3.Precondition: Student, Course, CourseCollection, StudentCollection classes must exist.

4.Postcondition: Must output both collections

5.The Throws List: when input is incorrect outputs “Invalid Input”.

Design: Takes input from user and sets id, name, name, gpa, and credits. Then displays StudentCollection when called.

Course Class

1. Short Description: contains setters and getters for course name, gpa, and credits.

2.Parameter Description:

Name: Stores the name of the inputted course

Gpa: Stores the inputted gpa

Credits: Stores the inputted credits

3.Precondition: none

4.Postcondition: The class can be used by the Driver class

5.The Throws List: none

Design: Class takes elements inputted by user and sets the elements of name, gpa, and credits to the input. Class is called and stored in student and then called and stored in linked list by StudentCollection.

Student Class

1. Short Description: contains getter and setters for id and student name.

2.Parameter Description:

Id: Stores inputted ID.

Name: Stores inputted Name.

3.Precondition: none

4.Postcondition: The class can be used by Driver and StudentCollection Classes.

5.The Throws List: none

Design: Class takes elements inputted by user and sets the elements of id and name to the input. Class is then called and stored in linked list by StudentCollection.

StudentNode Class

1. Short Description: Contains getters and setters for data and linked.

2.Parameter Description:

Data: Stores a student class instance.

Link: keeps track of node position

3.Precondition: Student Class must exist.

4.Postcondition: StudentCollection must be able to use it

5.The Throws List: none

Design: Takes an instance of student and stores it in a node. StudentCollection then adds node into linked list.

CourseCollection Class

Add

1. Short Description: adds a course to a student

2.Parameter Description:

Courses: Holds an array of Course Objects.

Count: increases the size of the array for every new object added.

3.Precondition: Course Class must exist.

4.Postcondition: Student Class must be able to use it.

5.The Throws List: none

Design: Inserts a course object into the array and increases the count.

Remove

1. Short Description: removes a course from a student

2.Parameter Description: Same as add

3.Precondition: Target must exist and be a String.

4.Postcondition: Desired target is removed from array.

5.The Throws List: none

Design: Removes specific Course object using input of the courses name within course object to remove it from the array and decrease the count.

Search

1. Short Description: Searches for a course by name

2.Parameter Description: same as add and remove

3.Precondition: target must exist and be a string

4.Postcondition: Desired target is removed form array

5.The Throws List: none

Design: Searches for a Specific object using input to find a Course object with the course name within the object.

StudentCollection Class

Add

1. Short Description: Add a student object to linked list

2.Parameter Description:

Head: keeps track of the head of the linked list

Numberofnodes: number of nodes in the linked list

Element: A student object

3.Precondition: Element must be a student object

4.Postcondition: A new Student object is added to the linked list.

5.The Throws List: none

Design: Adds a new Student object at the head of the linked list.

Remove

1. Short Description: removes a student

2.Parameter Description: Same as add

3.Precondition: target must be an integer.

4.Postcondition: The student with equal ID must be removed.

5.The Throws List: none

Design: Removes a specific student with an ID that is equal to the target.

Search

1. Short Description: searches for a student

2.Parameter Description: same as add and remove

3.Precondition: Target must be an integer.

4.Postcondition: Must display the student with equal ID.

5.The Throws List: none

Design: Searches for a student with an ID that is equal to the target.