



CENTRAL
MICHIGAN UNIVERSITY

Department of Computer Science

Academic Year	2026
Semester	<input type="checkbox"/> Fall <input checked="" type="checkbox"/> Spring <input type="checkbox"/> Summer
Course Code - Name	CPS 240 – Object Oriented Programming, A&D
Instructor	Dr. Razi Iqbal
Assessment	Assignment 1
Deadline	January 26, 2026

Assignment 1

The main purpose of this lab is to test your knowledge of LinkedLists.

Instructions:

- You are required to submit your source (.java) files on Blackboard.
- Students having exactly similar code will get a straight 0.
- Comment on your code appropriately. Failing to do so would result in losing marks.
- The deadline for submission of this assessment is **January 26, 2026**.
- A late penalty of 10% per day late will be applied, in the absence of a medical note, to a maximum of 2 days late. After 2 days, the assignment will not be accepted.

Rubric

- Correct creation of LinkedList class that splits properly (5 marks)
 - Node Class Creation (0.5 marks)
 - LinkedList Insert Method (1 marks)
 - LinkedList Print Method (0.5 marks)
 - LinkedList SplitAndPrint Method (3 marks)

Question

In this assignment, you are provided with a `LinkedList` of students, where each student has a name, GPA, and academic status. Your task is to split this `LinkedList` into two separate `LinkedLists`: `Passing_Students` and `Failing_Students`. A student is considered Passing if their status is "Pass" and Failing if their status is "Fail".

You are required to write a Java program that creates a Java class `Student` with three data members `String` name and `double` gpa and `String` status. This class should have at least one parameterized constructor and a `toString()` method that returns the name, age and status of the student. Please refer to the expected output for more details.

You are also expected to create a class `LinkedList` that implements a singly linked list which should have at least 3 methods:

- `public void insertElement(Student student)` which inserts a student into the `LinkedList`
- `public void print()` which prints the elements of the `LinkedList`
- `public void splitAndPrint()` that splits the `LinkedList` into two separate `LinkedLists` namely, `Passing_Students` and `Failing_Students`. You have to traverse through the `LinkedList` and check if the status of the student is `Pass`, insert that student into the `Passing_Students` else insert into `Failing_Students`. Finally print both the Linked Lists (`Passing_Students`, `Failing_Students`) by calling the `print` function from each linked list.

Don't forget to create a `Node` class in `LinkedList.java` that contains your `LinkedList` class.

You are provided with `Driver` class that has a `main` method which generates the list of students using `Student` class and calls methods as required.

Don't worry too much about Big-O notation in this exercise. Concentrate on creating a `LinkedList` with the required functions to get a better understanding of creating and traversing through a `LinkedList`.

Below is the expected output of the program:

```
All Students:
Name: Alice, GPA: 3.5, Status: Pass
Name: Bob, GPA: 1.9, Status: Fail
Name: Charlie, GPA: 2.8, Status: Pass
Name: Diana, GPA: 1.5, Status: Fail
Name: Ethan, GPA: 3.2, Status: Pass

Passing Students:
Name: Alice, GPA: 3.5, Status: Pass
Name: Charlie, GPA: 2.8, Status: Pass
Name: Ethan, GPA: 3.2, Status: Pass

Failing Students:
Name: Bob, GPA: 1.9, Status: Fail
Name: Diana, GPA: 1.5, Status: Fail
```

The Code for **Driver** class with **main** method is provided below.

```
public class Driver {

    public static void main(String[] args) {

        LinkedList students = new LinkedList();

        // Creating Student objects
        Student s1 = new Student("Alice", 3.5, "Pass");
        Student s2 = new Student("Bob", 1.9, "Fail");
        Student s3 = new Student("Charlie", 2.8, "Pass");
        Student s4 = new Student("Diana", 1.5, "Fail");
        Student s5 = new Student("Ethan", 3.2, "Pass");

        // Inserting students into the LinkedList
        students.insertElement(s1);
        students.insertElement(s2);
        students.insertElement(s3);
        students.insertElement(s4);
        students.insertElement(s5);

        // Printing original list
        System.out.println("All Students:");
        students.print();

        // Splitting and printing passing and failing students
        students.splitAndPrint();
    }
}
```