

A. LinkedList Exercise

```
public class Node {
    String data;
    Node next;
}

public class LinkedList {
    Node head;
}
```

1. Implement a LinkedList method named **create** that defines an empty linked list.
2. Implement a LinkedList method named **isEmpty** that determines whether a list is empty.
3. Implement a LinkedList method named **isFull** that determines whether a list is full.
4. Implement a LinkedList method named **clear** that clears the list.
5. Implement a LinkedList method named **duplicate** that creates a duplicate (deep copy) of the list.
6. The method `previousElement` has the following prototype:

String previousElement(String target);

The method will return the value (data) of the element preceding the element with a data that corresponds to target or null (if no preceding element exists).

B. Map Exercise

```
Map<String,ArrayList<String>> courses;
```

`studentCourses` maps a student's name to a list of courses he/she takes.

1. Using the `HashMap` class define an appropriate map object for `studentCourses`.
2. Define a method that adds a course to the list of courses associated with a student. If there is no list of courses associated with the student, then one should be created and inserted in the map.
3. Define a method that prints, for each student in the map, the set of courses the student is taking.
4. Define a method that determines whether a particular course is being taken by a particular student.
5. Define a method that determines whether a particular course is being taken by any student.
6. Run your code through the debugger so you can see how the `HashMap` object looks like.
7. The most popular course is defined as the one with the largest number of students. Define a method that determines the most popular course (for simplicity you can assume there is only one).