

Project 2 -Linked List

1. Design a java interface called **IDedObject** which has following abstract functions.

```
int getID() //Returns the ID of the object
void printID() //Prints the details of the ID
```

2. Design a java class **Magazine** that implements IDedObject interface and has the following class variables:

int magazineID

String magazineName

String publisherName

Implement suitable constructors, a printID function that prints all the variable values in separate lines and getID() function that returns the magazineID. Add any access and other functions necessary.

{Note all your data should be private and methods public inside the class}

3. Design a generic singly linked list java class (Give an appropriate name) to hold objects of the generic type <Anytype>. This AnyType should extend IDedObject. The linked list class must implement following member functions:

A constructor which generates an empty list .

```
void makeEmpty(); //empties the linked list
AnyType findID(int ID); // Get the generic type to get the particular id and
// returns AnyType. Don't remove the object from the list.
// returns null if the list is empty or ID not found.
boolean insertAtFront(<AnyType> x); // insert at front of list or return false if that ID
// already exists
AnyType deleteFromFront(); // delete and return the record at the front of the list or
// return null if the list is empty
AnyType delete(int ID); // find and delete the record with the given ID or returns
// null if it isn't found
void printAllRecords(); // print all elements in the order they appear in the linked
// list. if list is empty print appropriate message.
```

4. Write a java class to have the main function, create a linked-list to hold magazine type objects. It should prompt the user to choice list of the following operations.
- i. Make Empty
/// This Option makes the Linked list empty
 - ii. Find ID
/// print all details of the magazine ID, if it is in the list , if not print appropriate message
 - iii. Insert At Front
/// Get the magazine details from the user and add it to the front of the list
 - iv. Delete From Front
///Print the first item on the list and then delete it.
 - v. Delete ID
/// Print particular IDed item and then delete it.
 - vi. Print All Records
// Print all the records in the list
 - vii. Done
///Quit the program. For every other option after task completion display menu again.

Your program must read from the keyboard and print to the screen.

Sample run

:

Operations on List

1. Make Empty
2. Find ID
3. Insert At Front
4. Delete From Front
5. Delete ID
6. Print All Records
7. Done

Your choice: 3

Enter Magazine ID: 1111

Enter Magazine Name: People

Enter Publisher Name: People

...

Magazine Added

Operations on List

1. Make Empty

2. Find ID
3. Insert At Front
4. Delete From Front
5. Delete ID
6. Print All Records
7. Done

Your Choice: 2

ID No: 1111

1111

People

People

Operations on List

1. Make Empty
2. Find ID
3. Insert At Front
4. Delete From Front
5. Delete ID
6. Print All Records
7. Done

Your Choice: 4

1111

People

People

First item deleted

Operations on List

1. Make Empty
2. Find ID
3. Insert At Front
4. Delete From Front
5. Delete ID
6. Print All Records
7. Done

Your Choice: 7

Done.

etc

Additional Instructions:

Follow all project instructions given in elearning page. Include a README.txt file explaining the IDE that you used. Permissible IDEs -> Netbeans, Eclipse and IntelliJ; Other IDEs need permission from TA before project submission. If no IDE used and the program need to be executed on the command prompt, give elaborate instructions on how to run and execute your program. Your README file should also state what works and what doesn't work and contain sample run of your program.

Zip your files into one file before submitting, have the 'readme' file outside of zip file (elearning lets multiple files in one submission). Use good style and layout and comment your code well.