NATIONAL UNIVERSITY OF COMPUTING AND EMERGING SCIENCES
ASSIGNMENT 01 Data Structures

Submission: Assignments are to be done individually. Submissions that do not comply with the specifications given in this document will not be marked and a zero grade will be assigned. Write your name and e-mail id in a comment line in on top of each source file. You are required to submit a single zip file containing an archive of your code files on Google Classroom. You should name your zip as i21-XXXX.zip where i21-XXXX represents your student id.
Deadline: Deadline to submit project is **1st March 2024**, 11:00 PM. No submission will be considered for grading outside Google Classroom or after **1st March 204, 11:00 PM**. Correct and timely submission of project is responsibility of every student; hence no relaxation will be given to anyone.
Plagiarism: -100% marks in the project if any significant part of project is found plagiarized. A code is considered plagiarized if more than 20% code is not your own work.

DVD STORE

During holidays or on weekends, a family or an individual typically rents a DVD either from a local store or online. Therefore, write a program that does the following:

1. Rent a DVD; that is, check out a DVD.
2. Return, or check in, a DVD.
3. Create a list of DVDs owned by the store.
4. Show the details of a particular DVD.
5. Print a list of all the DVDs in the store.
6. Check whether a particular DVD is in the store.
7. Maintain a customer database.
8. Print a list of all the DVDs rented by each customer.

So, write a program for the DVD store. This example further illustrates the object oriented design methodology and, in particular, inheritance and overloading. The programming requirement tells us that the DVD store has two major components: **DVDs** and **customers**. You need to implement the details of these two components by maintaining various lists:

1. A list of all the DVDs in the store
2. A list of all the store's customers
3. Lists of the DVD currently rented by each customer

You need to develop the program in two parts.

In part 1, design, implement, and test the dvd component. In part 2, design and implement the customer component, which is then added to the dvd component developed in part 1. That is, after completing parts 1 and 2, you can perform all the operations listed previously.

The common things associated with a dvd are as follows:

1. Name of the movie
2. Names of the stars
3. Name of the producer
4. Name of the director
5. Name of the production company
6. Number of copies in the store

From this list, we see that some of the operations to be performed on the dvd object are as follows:

1. Set the dvd information—that is, the title, stars, production company, and so on.
2. Show the details of a particular dvd.
3. Check the number of copies in the store.
4. Check out (that is, rent) the dvd. In other words, if the number of copies is greater than zero, decrement the number of copies by one.
5. Check in (that is, return) the dvd. To check in a dvd, first we must check whether the store owns such a dvd and, if it does, increment the number of copies by one.
6. Check whether a particular dvd is available—that is, check whether the number of copies currently in the store is greater than zero. The deletion of a dvd from the dvd list requires that the dvd list be searched for the dvd to be deleted. Thus, we need to check the title of a dvd to find out which dvd is to be deleted from the list. For simplicity, we assume that two dvds are the same if they have the same title.

The primary operations on the dvd list are to check in a dvd and to check out a video. Both operations require the list to be searched and the location of the dvd being checked in or checked out to be found in the dvd list. Other operations such as seeing whether a particular dvd is in the store, updating the number of copies of a dvd, and so on also require the dvd list to be searched. To simplify the search process, we will write a function that searches the dvd list for a particular dvd. If the dvd is found, it sets a parameter found to true and returns a pointer to the dvd so that check-in, check-out, and other operations on the dvd object can be performed.

The customer object stores information about a customer, such as the first name, last name, account number, and a list of dvds rented by the customer. Every customer is a person. You have to design the class and described the necessary operations on the name of a person.
The basic operations on an object of type customer are as follows:

1. Print the name, the account number, and the list of rented dvds.
2. Set the name and the account number.
3. Rent a dvd; that is, add the rented dvd to the list.
4. Return a dvd; that is, delete the rented dvd from the list.
5. Show the account number.

The details of implementing the customer component are left as a free hand for you.

In the main program test the dvd object. Assume that the necessary data for the dvds are stored in a file. We will open the file and create the list of dvds owned by the dvd store. Write a function, to read the data from the input file and create the list of dvds. Also write a function, to show the different choices—such as check in a dvd or check out a dvd—that the user can make. Perform various operations such as 1: To check whether the store carries a particular dvd 2: To check out a dvd 3: To check in a dvd 4: To check whether a particular dvd is in stock 5: To print only the titles of all the dvd 6: To print a list of all the dvd 9: To exit

- Complete the design and implementation of the class customerType
- Design and implement the class customerListType to create and maintain a list of customers for the video store.
- Complete the design and implementation of the dvd store program. In other words, write a program that uses the classes mentioned above to make a dvd store operational.
- Redo the dvd store program so that the list of dvds, the list of customers, and the list of dvds rented by a customer are kept in a list container

NATIONAL UNIVERSITY OF COMPUTING AND EMERGING SCIENCES
ASSIGNMENT 01 Data Structures

Class DVDType	setDVDInfo NoOfCopiesInStock CheckIn CheckOut PrintTitle PrintInfo CheckTitle UpdateStock DVDType ….	15 Marks
Class DVDListType	videoSearch isVideoAvailable videoCheckOut videoCheckIn videoCheckTitle ….	15 Marks
class customerType	setName print firstName lastName accountNo TotalDvdsRented ….	15 Marks
List container	implementation	10 Marks