

# 48024

# Applications Programming

# Assignment 1

## Topics:

OO Design, Standard Patterns, Lists

## Learning Outcomes:

This assessment task addresses the following subject learning objectives (SLOs): 1, 2 and 3

## Due date:

11:59PM Monday the 7th of May

## Weight:

30%

## Individual Work

All work is individual. You may discuss ideas, approaches and problems, but you should write every line of code yourself except for code copied from the lecture notes, lecture code or lab code. You **MUST NOT** let another student see your solution code, and you **MUST NOT** look at another student's solution code. More information about Academic Misconduct can be found at:

<http://www.gsu.uts.edu.au/rules/student/section-16.html>

## Overview



### Movie Kiosk – A Movie Renting System

You are a Software Engineer at a Movie Rental company and you have been asked to write the software for their latest piece of technology, a fully automated movie renting system.

This is one of latest strategies implemented by the company that aims to increase profits and makes their products more accessible to their customer base.

However, they are struggling to find a capable programmer that can build a robust application that is fully functional and easy to use.

This requirement is urgent, as competitor companies have heard the rumors of this state-of-the-art system and they have also started working on similar prototypes.

## Specification

The Movie Renting System (or Movie Kiosk) will consist of two main components, an administrative component, and a movie catalogue.

The administrative section will allow, through text-based menus, the addition and removal of customers, the addition and removal of movies in the catalogue, the display of customer records, the display of a customer's 3 favorite movies and the top-up of customer accounts.

The catalogue will store a list of all the movies in the kiosk, a list of the genres of those movies, and a list of the available movies. The catalogue will also allow display of the movies in a number of ways, including by availability, genre and year. Finally, the catalogue will also handle the renting and return of movies.

Each customer record will include the customer's name, their chosen ID number, their current balance, which movies they are currently renting and their full renting history.

Each movie record will include the movie's title, its genre and the year in which it was made.

## An aside

While reading the first part of the specification, you will notice there is a lot going on.

- How many functions did you identify?
- How many classes did you identify?
- What are the fields in each class?
- How many goals did you identify?
- How many patterns did you think of that might be applicable?

This assignment will be challenging and you will probably want to manage your time well.

- How long do you think it will take you to code the functions?
- How long do you think it will take you to code each goal?
- A good rule of thumb is to think of an estimate, and then multiply that number by 3 or 4!
- To manage your time well, you may need to figure out which parts of the assignment you can start early.
- Which parts can you start now?
- Which parts can you start in week 6?

If you complete parts in the same week that you learn the topics (while they are fresh in your mind), they will take less time to complete.

## The User Interface

Below is a sample I/O trace. The red text indicates user input (you are not expected to print color text). Not every conceivable scenario is shown below and you should submit your code to PLATE to see what specific scenarios are tested. You should also implement your solution in the same order as PLATE's test cases so that you can receive incremental feedback and marks as you progress.

```
Welcome to the Movie Kiosk! Please make a selection from the menu:
```

```
1. Explore the catalogue.
2. View your customer record.
3. Show you favourite movies.
4. Top up account.
5. Enter Admin Mode.
X. Exit the system.
Enter a choice: 5
Welcome to the administration menu:
1. List all customers.
2. Add a customer.
3. Remove a customer.
4. List all movies.
5. Add a movie to the catalogue.
6. Remove a movie from the catalogue.
R. Return to the previous menu.
Enter a choice: 2
```

```
Adding a new customer.
Enter a new ID: 101
Enter the customer's name: Jaime
Enter the customer's initial balance: 5
Customer added.
```

```
Welcome to the administration menu:
1. List all customers.
2. Add a customer.
3. Remove a customer.
4. List all movies.
5. Add a movie to the catalogue.
6. Remove a movie from the catalogue.
R. Return to the previous menu.
Enter a choice: 2
```

Adding a new customer.  
Enter a new ID: **102**  
Enter the customer's name: **Luke**  
Enter the customer's initial balance: **3**  
Customer added.

Welcome to the administration menu:  
1. List all customers.  
2. Add a customer.  
3. Remove a customer.  
4. List all movies.  
5. Add a movie to the catalogue.  
6. Remove a movie from the catalogue.  
R. Return to the previous menu.  
Enter a choice: **1**

The Kiosk has the following customers:  
101      Jaime      \$ 5  
102      Luke      \$ 3

Welcome to the administration menu:  
1. List all customers.  
2. Add a customer.  
3. Remove a customer.  
4. List all movies.  
5. Add a movie to the catalogue.  
6. Remove a movie from the catalogue.  
R. Return to the previous menu.  
Enter a choice: **5**

Adding a new movie.  
Enter the title of the movie: **Titanic**  
Enter the year: **1997**  
Enter the genre: **Drama**  
Enter price: **4**  
Added Titanic to catalogue.

Welcome to the administration menu:  
1. List all customers.  
2. Add a customer.  
3. Remove a customer.  
4. List all movies.  
5. Add a movie to the catalogue.  
6. Remove a movie from the catalogue.  
R. Return to the previous menu.  
Enter a choice: **5**

Adding a new movie.  
Enter the title of the movie: **Matrix**  
Enter the year: **1999**  
Enter the genre: **SciFi**  
Enter price: **4**  
Added Matrix to catalogue.

Welcome to the administration menu:  
1. List all customers.  
2. Add a customer.  
3. Remove a customer.  
4. List all movies.  
5. Add a movie to the catalogue.  
6. Remove a movie from the catalogue.  
R. Return to the previous menu.  
Enter a choice: **4**



The Kiosk has the following movies:

1997 Titanic Drama \$4  
1999 Matrix SciFi \$4

Welcome to the administration menu:

1. List all customers.
2. Add a customer.
3. Remove a customer.
4. List all movies.
5. Add a movie to the catalogue.
6. Remove a movie from the catalogue.
- R. Return to the previous menu.

Enter a choice: **R**

Welcome to the Movie Kiosk! Please make a selection from the menu:

1. Explore the catalogue.
2. View your customer record.
3. Show you favourite movies.
4. Top up account.
5. Enter Admin Mode.
- X. Exit the system.

Enter a choice: **1**

Welcome to the Catalogue! Please make a selection from the menu:

1. Display all movies.
2. Display all available movies.
3. Display all genres.
4. Display movies in a genre.
5. Display all movies by year.
6. Rent a movie.
7. Return a movie.
- R. Return to previous menu.

Enter a choice: **3**

The Kiosk has movies in the following genres:

Drama  
SciFi

Welcome to the Catalogue! Please make a selection from the menu:

1. Display all movies.
2. Display all available movies.
3. Display all genres.
4. Display movies in a genre.
5. Display all movies by year.
6. Rent a movie.
7. Return a movie.
- R. Return to previous menu.

Enter a choice: **6**

Enter a valid customer ID: **101**

Enter the title of the movie you wish to rent: **Matrix**

Movie rented.

Welcome to the Catalogue! Please make a selection from the menu:

1. Display all movies.
2. Display all available movies.
3. Display all genres.
4. Display movies in a genre.
5. Display all movies by year.
6. Rent a movie.
7. Return a movie.
- R. Return to previous menu.

Enter a choice: **2**

The following movies are available:

1997 Titanic Drama \$4



Welcome to the Catalogue! Please make a selection from the menu:

1. Display all movies.
2. Display all available movies.
3. Display all genres.
4. Display movies in a genre.
5. Display all movies by year.
6. Rent a movie.
7. Return a movie.
- R. Return to previous menu.

Enter a choice: **6**

Enter a valid customer ID: **102**

Enter the title of the movie you wish to rent: **Titanic**

You don't have sufficient funds to rent this movie.

Welcome to the Catalogue! Please make a selection from the menu:

1. Display all movies.
2. Display all available movies.
3. Display all genres.
4. Display movies in a genre.
5. Display all movies by year.
6. Rent a movie.
7. Return a movie.
- R. Return to previous menu.

Enter a choice: **R**

Welcome to the Movie Kiosk! Please make a selection from the menu:

1. Explore the catalogue.
2. View your customer record.
3. Show you favourite movies.
4. Top up account.
5. Enter Admin Mode.
- X. Exit the system.

Enter a choice: **4**

Enter a customer ID: **102**

Enter the top-up amount: **5**

Transaction complete.

Luke's balance was: \$3

Luke's current balance is: \$8

Welcome to the Movie Kiosk! Please make a selection from the menu:

1. Explore the catalogue.
2. View your customer record.
3. Show you favourite movies.
4. Top up account.
5. Enter Admin Mode.
- X. Exit the system.

Enter a choice: **1**

Welcome to the Catalogue! Please make a selection from the menu:

1. Display all movies.
2. Display all available movies.
3. Display all genres.
4. Display movies in a genre.
5. Display all movies by year.
6. Rent a movie.
7. Return a movie.
- R. Return to previous menu.

Enter a choice: **6**

Enter a valid customer ID: **102**

Enter the title of the movie you wish to rent: **Titanic**

Movie rented.

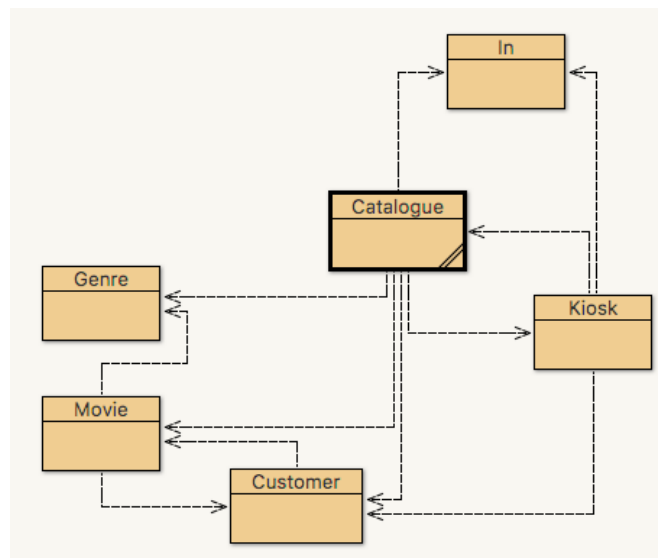
Welcome to the Catalogue! Please make a selection from the menu:

1. Display all movies.
2. Display all available movies.
3. Display all genres.
4. Display movies in a genre.
5. Display all movies by year.
6. Rent a movie.
7. Return a movie.
- R. Return to previous menu.

Enter a choice: **x**

## Requirements

- Your design will consist of exactly the following classes with the listed fields, declared as indicated. You may not add or remove classes or fields; however, you may add constructors, functions and procedures to complete your design (in fact, you will have to!). You should pay careful attention to the tests on PLATE, as these will help guide you with some (but not all) of these methods.
- To help visualize the design, a partial class diagram has been provided



- Classes – your design will consist of these 6 classes:
  1. Kiosk
  2. Catalogue
  3. Customer
  4. Movie
  5. Genre
  6. In (this is just the class you've been using throughout the labs to facilitate simpler I/O – just copy it over)



- Fields – the classes will have the following fields:

```
public class Kiosk {
    private Catalogue catalogue;
    private List<Customer> customers;
}

public class Catalogue {
    private Kiosk kiosk;
    private List<Movie> moviesAvailable;
    private List<Movie> moviesRented;
    private List<Genre> genres;
}

public class Customer {
    private int ID;
    private String name;
    private int balance;
    private List<Movie> currentlyRented;
    private List<Movie> rentingHistory;
}

public class Movie {
    private String title;
    private int year;
    private int price;
    private Genre genre;
}

public class Genre {
    private String name;
}
```

- The fields also have some additional requirements and structures:

Lists all have the abstract type of List<>, but must be instantiated with a concrete type that implements the List<> behavior (you will see two of these in week 6, you can choose either – you may also want to think about why you might do things this way).





- Constructors – the constructors of the class have the following requirements:
  1. All constructors initialize the fields of their class.
  2. The Kiosk constructor takes no parameters.
  3. The Catalogue constructor takes a single parameter, the Kiosk which it belongs to.
  4. The Customer constructor takes three parameters, the ID, name, (initial) balance, corresponding to the three fields identically named.
  5. The Movie constructor takes four parameters, corresponding to the title, year, genre and price, with the same types as the respective fields.
  6. The Genre constructor takes a single parameter, the name of the genre.
- toString() – several of the classes will have a toString() function, with the following formatting:
  1. `Customer.toString()` will produce a string of the form:
 

```
<ID>\t<name>\t$ <price>
```

e.g.

```
50      Homer Simpson      $ 15
```

Note: \t is the escape code for the tab character.
  2. `Movie.toString()` will produce a string of the form:
 

```
<year>\t<title>\t<genre name>\t$ <price>
```

e.g.

```
1997   Titanic      Drama      $ 4
```
  3. `Genre.toString()` will produce a string of the form:
 

```
<genre name>
```

e.g.

```
SciFi
```
- The main method of the program will be in the Kiosk class.
- Top-Ups: The Kiosk should also implement a top-up system. A customer should be able to top up their account by simply providing their ID and the top-up amount. When renting a movie, the price of the chosen movie will be deducted from the customer's account decreasing their balance. If the customer does not have sufficient funds, they will not be able to rent any movies.

## Advanced Requirements

To achieve a mark of >84, you must implement a favorite reporting function (or set of functions). This corresponds to an item in the main menu (see the trace above). Choosing this option should prompt the user for a valid customer ID, and then display the user's 3 most rented movies, in order of highest renting frequency to lowest. If the user has only rented less than three movies, it will show those movies alone. A sample partial I/O trace follows, again red text is used to indicate user input:

```
Welcome to the Movie Kiosk! Please make a selection from the menu:
```

```
1. Explore the catalogue.
2. View your customer record.
3. Show you favourite movies.
4. Top up account.
5. Enter Admin Mode.
X. Exit the system.
```

```
Enter a choice: 3
```

```
Enter a customer ID: 101
```

```
Jaime's favourite movies are:
```

```
1997   Titanic Drama   $4
1999   Matrix SciFi    $4
1991   The Silence of the Lambs      Crime   $4
```

## Expected Workload

The time to do the assignment to a credit/distinction level has been estimated at 25 hours for a student of average ability who has completed all the tutorial and lab exercises.

## Online Support

The Assignment 1 discussion board has been set up on UTSOnline so that students can ask questions, and other students can reply. The course coordinator will only post a reply only if the student response was wrong, or in the case of correcting a mistake in the assignment specification.

You must not post or share Java code to the discussion board. The board is there to help you, not to provide the solution. Posting your code is academic misconduct and will reported. Each time this rule is violated, the code will be removed and replaced with a comment of the form: "Strike 1: Posting code". After 3 strikes, the discussion board will be deleted because it did not work.

FAQs (Frequently Asked Questions) and their answers will be posted on PLATE alongside the assignment specification. If you have a question, check the FAQ first; it may already be answered there. You should read the FAQ at least once before you hand in your solution, but to be safe check it every couple of days. Anything posted on the FAQ is considered to be part of the assignment specification. The FAQ will be frozen (no new entries) two days before the due date; no questions will be answered after it is frozen.

If anything about the specification is unclear or inconsistent, contact the subject coordinator who will try to make it clearer by replying to you directly and posting the common questions and answers to the FAQ. This is similar to working on the job, where you ask your client if you are unsure what has to be done, but then you write all the code to do the task. Email [Jaime.Garcia@uts.edu.au](mailto:Jaime.Garcia@uts.edu.au) to ask for any clarifications or corrections to the assignment.

## PLATE Marking

Your solution is marked for correctness by PLATE (<https://plate.it.uts.edu.au/>) by comparing the output of your system to the output of the benchmark system. You can submit a solution to PLATE many times; I urge you to do this, so you receive credit for your work. Submission takes the same form as for the labs, you must package your assignment in a JAR file, including the source code. PLATE will test the features of your program in a certain order: Classes and fields, then constructors, then goals from basic to advanced. PLATE cannot test the more advanced goals until the basic goals are working. To receive marks, you must pass PLATE's test cases in the order in which PLATE tests them.

Your code is marked by software, so you can get a good mark by fooling or spoofing the software. If you spoof a task worth N marks, you receive a penalty of 2\*N marks.

## Test Cases

For testing purposes, please make sure that you have a database of movies and customers as follows:

Title	Year	Genre	Price
Matrix	1999	SciFi	3
Titanic	1997	Drama	4
The Silence of the Lambs	1991	Crime	3
Jurassic Park	1993	SciFi	4
Terminator 2	1991	SciFi	3

ID	Name	Balance
101	Jaime	10
102	Luke	10
103	William	1

## Submission and Return

Your solution is to be submitted to PLATE at <https://plate.it.uts.edu.au/> under Applications Programming / Assessments / Assignment 1, in the same manner as your labs – package in a JAR file including the source code. Your provisional mark and feedback is generated immediately each time you submit to PLATE. However, analysis of spoofing, plagiarism, collusion and general cheating is done in the two weeks following the due date. If you are suspected of Academic Misconduct, I will forward your case to the Misconduct Committee and will notify you by email.

There is no scheduled late submission period. An extension of up to one week may be given by the subject coordinator before the due date; you have to supply documentary evidence of your claim. An extension CANNOT be given after the due date. You may also apply for special consideration for reasons including unexpected health, family or work problems. More information about how to apply for special consideration can be found at: <http://www.sau.uts.edu.au/assessment/consideration.html>.

## Marking Scheme

The marks for the assignment are divided into the following functionality components (note that individual tests may test several functionality components, and a functionality component may be tested by several tests):

Functionality Component	Mark Allocation
Fields	8
Constructors	6
Main Menu	5
Admin Menu	5
Catalogue Menu	5
Add Customer	5
Add Movie	5
Remove Customer	5
Remove Movie	5
Show Movies	5
Show Movies by Genre	5
Rent	5
Return	5
Customer Record	5
Favorites	16
Account Top-up	10

This adds to a mark out of 100, at makes up 30% of your final assessment mark.