Beijing-Dublin International College

SEMESTER 1 FINAL EXAMINATION - (2017/2018)

School of Computer Science

COMP2005J Object Oriented Programming Programming Exam

Prof. Pádraig Cunningham
Dr. Seán Russell*

Time Allowed: 180 minutes

Instructions for Candidates:

Answer all questions.

BJUT Student ID:________________ **UCD Student ID:**________________

I have read and clearly understand the Examination Rules of both Beijing University of Technology and University College Dublin. I am aware of the Punishment for Violating the Rules of Beijing University of Technology and/or University College Dublin. I hereby promise to abide by the relevant rules and regulations by not giving or receiving any help during the exam. If caught violating the rules, I accept the punishment thereof.

Honesty Pledge:____________________________________ **(Signature)**

Instructions for Invigilators

Students are permitted to bring any printed notes and textbooks into the examination.

- All questions are based on the same program. Each question adds some new functionality to the code. You should not move to the next question until you have finished the one before it.
- You should only submit working code. If your code does not compile or work correctly you should comment it out. You will **fail** if you submit code that does not compile.
- To improve your grade you should submit well designed code with light comments and correct indentation.
- The main method of the program should be in a class named `Main`.
- There are bonus marks for providing good JUnit test cases for the classes you have written. This is not required to get 100%, but it will improve your score.
- There are files with some input data for the program on moodle. They are in a zip file called InputFiles.zip. Download these at the beginning of the exam. Internet access will be turned off after 10 minutes. If you have not downloaded the input files you will have to create them yourself. These files should be put in the project folder in eclipse, not in the src folder with your code. The password for the zip file is `"goodluck"` (without the quotes).
- The classes you define should have good encapsulation. If you need to access an instance variable you should add a getter or setter method.

The questions in this exam will all be based around the workings of a movie rental service. All questions will be associated with this concept.

Question 1: Customers

Customers will all have an account with the service and amount of credits that they have bought to rent movies with. Credits will be represented by an integer number and each movie may cost a different number of credits to rent. Each customer has been given a unique id (an integer number) to represent them, we will also remember their name and the amount of credit that they have left.

a. Implement a `Customer` class in Java to represent a single customer. The class should have a constructor and any required getter methods.

(10%)

b. Add a method to the `Customer` class called rent. This method should reduce the users number of credits by an amount specified in the parameter. If the user has enough credits, then the reduction should be made and the method should return true. If the user does not have enough credits in their account, the method should return false and not reduce the credits.

(5%)

c. Override the `toString` method in the `Customer` class, when called this method should return a String containing the customers name, followed by their id in brackets and their remaining credits. For example it should look like this `"Sean Russell (12) has 100 credits remaining"`.

(5%)

d. Add code to the main method of the program to read all of the customer data from the file `"1.txt"`. Each line of the file represents a single customer and the data is in the following format `<id> <credits> <name>`. After reading the data from the file a Customer object should be created for each and then printed to the screen. (There could be an unknown number of customers in the file so you should write your code to read the whole file) the maximum number of Customers in the file will be 1000.

   The output of the main method should look something like this:

```
Nicola Davies (759) has 86 credits remaining
Karen Lewis (657) has 101 credits remaining
Simon Poole (452) has 31 credits remaining
Emma White (294) has 119 credits remaining
...
Mary Bell (789) has 155 credits remaining
```

(10%)

(Question Total 30%)

Question 2: Movie Info

For each movie that can be rented, it is given a unique id (unique only for movies), a title and the number of credits it costs to rent.

a. Implement a `Movie` class in Java to represent a single movie. The class should have a constructor and any required getter methods.

(10%)

b. Override the `toString` method for the Movie class, when called this method should return a String containing the movies title, followed by the id and the cost. For example it should look like this `"I Am Number Four (2011) has id 204 and costs 9 credits"`.

(5%)

c. Add code to the main method (after the code from question 1) to read all of the movie information from the file `"2.txt"`. Each line of the file contains the information about one movie and the data is in the following format `<id> <cost> <title>`. After reading the data from the file a Movie object should be created for each and then printed to the screen. (There could be an unknown number of movies in the file so you should write your code to read the whole file) the maximum number of movies in the file will be 1000.

   The output of the main method should look something like this:

```
There Will Be Blood (2007) has id 684 and costs 7 credits
Remember the Titans (2000) has id 447 and costs 9 credits
...
Tinker Bell and the Legend of the NeverBeast (2015) has id
   338 and costs 6 credits
```

(10%)

(Question Total 25%)

Question 3: Rental Service

The movies and customers are all managed by a rental service, this service will have functionality like renting a movie to a customer and allowing customers to buy more credits. Before starting this question, you should read all of the questions and think about what data structure would be best for remembering the customers and movies. Choosing well will make the code much easier to complete.

a. Implement a class in Java to represent the rental service. This class should have data structures for remembering movies and customers. The data structures should be constructed in the constructor of the class.

(5%)

b. Add the methods `addCustomer` and `addMovie` to the rental service class. These methods should take a single parameter of a Customer or a Movie object and add that to the data structure to be remembered. The method does not need to return anything.

(10%)

c. Add the method `getCustomer` to the rental service class. This method should take a single integer as a parameter and return the customer object that has that id. You can assume that the id will always match one of the customer objects that has been added.

(5%)

d. Add the method `getMovie` to the rental service class. This method should take a single integer as a parameter and return the movie object that has that id. You can assume that the id will always match one of the movie objects that has been added.

(5%)

e. Add the method `rentMovie` to the rental service class. This method should take two integers as parameters. The first parameter is the id of a customer and the second parameter is the id of a movie they want to rent. If the customer has enough credits, then their credits should be reduced and the message ”XXXX rented YYYYY and has ZZ credits remaining” should be printed to the screen, and if the user does not have enough credits to rent the movie, then the message ”XXXXX did not have enough credits to rent YYYYY”. Here XXXX represents the customers name, YYYYY the title of the movie and ZZ how many credits the customer has now.

(10%)

f. Add code to the main method (after the code from question 2) to create a rental service object. All of the customer and movie objects created in question 1 and 2 should be added to the rental service. You should then add code to read the file `"3.txt"`, this contains a set of movie rental transactions. Each line in the file contains two numbers, the first is a customer id and the second is a movie id. These should be read and used as parameters to the rentMovie method.

The output of the main method should look something like this:

```
Virginia Hamilton rented Harold & Kumar Go to White Castle
   (2004) and has 175 credits remaining
...
Stewart Brown did not have enough credits to rent Before
   Sunset (2004)
```

```
Emma White rented Up (2009) and has 98 credits remaining
Irene Dickens rented Gone Girl (2014) and has 154 credits
   remaining
Jennifer McDonald rented Crystal Fairy & the Magical Cactus
   (2013) and has 146 credits remaining
```

(10%)

(Question Total 45%)

Submission

Create a single **zip** file containing your entire project. This file should be named using your UCD student number.