



## Fundamentals of Programming (FSP)

Year 1 (2014/15), Semester 1

### SCHOOL OF INFOCOMM TECHNOLOGY

Diploma in Financial Informatics

Diploma in Information Technology

## ASSIGNMENT

Due on 15<sup>th</sup> August 2014 (Friday), 6.00 pm

Duration:	2 weeks (25 <sup>th</sup> July – 15 <sup>th</sup> August 2014)
Weightage:	30% of Module
Individual/Team/Both:	Individual
Format:	Programming (30%) Walkthrough Test (70%)

#### Penalty for late submission:

NO late submission shall be entertained after the Walkthrough Test.

There are a total of 9 pages (including this page) in this handout.

#### **WARNING**

*If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this assignment. Disciplinary action will also be taken. Similar action will be taken for the student who allows other student(s) to copy his/her work.*

## 1. OBJECTIVE

This assignment assesses the student's ability to apply relevant programming concepts (**including parallel array** manipulation) to develop a simple application using the Java programming language.

## 2. SCOPE

Merlion iGames (MiG), a company dealing in three categories of items: manga, anime, and video games, requires a system to manage its rental system.

You are assigned to develop a simple Java application program to maintain information about these items.

Some samples of the information maintained are shown in Figure 1:

Category	Title	Serial No.	Rental Charge	Status
Manga	The Renegade	2222	9.75	On Loan
Anime	Flying Pigs	1111	12.40	Available
Video Game	Mortar ConBet	4444	23.45	On Loan
Video Game	5D Warcraft	7788	24.80	On Loan
Video Game	Bot Gains	108	44.55	Available
Anime	Alley Barber	4096	36.00	On Loan
Manga	Bokuno Kokorowa	5566	13.95	Available
Anime	Condo Zeroes	9413	16.60	Available
Video Game	Sun Zi in SupraNova	777	100.00	On Loan
...	...	...	...	...

*(Note: The status is actually stored as a boolean value in the application as follows:*

- *true, if the item is available for loan;*
- *false, if the item is on loan.)*

**Figure 1 – Sample MiG Information**

The assignment consists of “**Basic Requirements**”, “**Additional Requirements**” and “**Descriptive Section**” as described in sections 3 to 5. **You SHOULD complete the basic requirements BEFORE proceeding with the additional requirements and descriptive section.**

For this assignment, you are expected to do the following :

- understand the problem completely and plan your program layout before you start coding your program;
- break your program into smaller and simpler parts;
- implement and test a part at a time;
- **use methods wherever appropriate.**

The required resources for this assignment are:

- ♦ Java SE 8 (or higher version)
- ♦ jGrasp 2.0 (or higher version)

### 3. BASIC REQUIREMENTS (to be submitted by Wednesday, 6<sup>th</sup> August 2014)

The system should provide the following **basic** features:

- **Initialisation**

The program should **initialise the data for the first 3 items** shown in Figure 1.

- **Display main menu (repeatedly)**

When the program is run, it should display the main menu as shown in Figure 2. When the user enters an option from 1 to 5, the program will process the option accordingly. After the option has been processed, the program will display the main menu again and the process is repeated until the user chooses option 0 to exit.

```
MENU
=====
[1] List items
[2] Display available items
[3] Enquire average rental of items on loan
[4] Add new item
[5] Update status of an item
[0] Exit
Enter your option:
```

**Figure 2 - Main Menu**

- **List items**

This option allows the user to view all the items in the system, as shown in Figure 3:

```
List items
=====
Item  Category  Title                Serial No.  Rental Charge  Status
1     Manga     The Renegade         2222       9.75          On Loan
2     Anime     Flying Pigs          1111       12.40         Available
3     Video Game Mortar ConBet        4444       23.45         On Loan
```

**Figure 3 – List items**

- **Display available items**

This option allows the user to view the items which are currently available for rental, as shown in Figure 4.

Item	Category	Title	Serial No.	Rental Charge	Status
1	Anime	Flying Pigs	1111	12.40	Available

**Figure 4 – Items available for rental**

- **Enquire average rental of items on loan**

This option allows the user to find out the average rental charge for items on loan. See Figure 5. (Note: This part of your implementation should call a method that computes average.)

```
Average rental of items on loan
=====
The average rental for all items on loan is $16.60.
```

**Figure 5 – Average rental of items on loan**

- **Add new item**

This option allows the user to add a new item to the application, as shown in Figure 6. Note that a new item has a status of “Available”.

```
Add new item
=====
Category (1-Manga 2-Anime 3-Video Game): 3
Title: 5D Warcraft
Serial No.: 7788
Rental Charge : $24.80
One new item added.

List items
=====
Item  Category      Title              Serial No.  Rental Charge  Status
1     Manga          The Renegade      2222        9.75          On Loan
2     Anime            Flying Pigs       1111        12.40         Available
3     Video Game       Mortar ConBet     4444        23.45         On Loan
4     Video Game       5D Warcraft       7788        24.80         Available
```

**Figure 6 – Add new item**

- **Update status of an item**

This option allows the user to change the status of an item. The user needs to select the item no. from a displayed list. The status is then updated automatically. A sample screenshot is shown in Figure 7.

```
Update Status
=====
List items
=====
Item  Category      Title              Serial No.  Rental Charge  Status
1     Manga          The Renegade      2222        9.75          On Loan
2     Anime            Flying Pigs       1111        12.40         Available
3     Video Game       Mortar ConBet     4444        23.45         On Loan
4     Video Game       5D Warcraft       7788        24.80         On Loan

Enter item no.: 3

List items
=====
Item  Category      Title              Serial No.  Rental Charge  Status
1     Manga          The Renegade      2222        9.75          On Loan
2     Anime            Flying Pigs       1111        12.40         Available
3     Video Game       Mortar ConBet     4444        23.45         Available
4     Video Game       5D Warcraft       7788        24.80         On Loan
```

**Figure 7 – Update new item**

- **Validation** (and feedback)

The program should handle ALL invalid entries by the user, e.g. invalid numeric data, adding entries when the array is out of storage space, duplicate items, etc. If the user made a mistake in the entry, the program should inform the user via appropriate feedback for him to make corrections.

- **Program documentation, indentation and appropriate use of methods**

The program should be properly documented, indented and should make use of methods where appropriate.

#### 4. ADDITIONAL REQUIREMENTS (to be submitted by Friday, 15<sup>th</sup> August 2014)

The **additional** features are listed below. All the basic features should still work even after these are incorporated.

##### A1 Changing the name of an item's category

This option allows the user to change the name of the category of an item; e.g., changing the category of "Flying Pigs" from "Anime" to "Manga". The user needs to select the serial no. of the item from a displayed list, and then enter the new category when prompted.

##### A2 List the items as grouped by categories

This option allows the user to view the information for all the items, grouped by category; e.g.,

Category	Title	Serial No.	Rental Charge	Status
Manga	The Renegade	2222	9.75	On Loan
Manga	Bokuno Kokorowa	5566	13.95	Available
Anime	Flying Pigs	1111	12.40	Available
Anime	Alley Barber	4096	36.00	On Loan
Anime	Condo Zeroes	9413	16.60	Available
Video Game	Mortar ConBet	4444	23.45	Available
Video Game	5D Warcraft	7788	24.80	On loan
Video Game	Bot Gains	108	44.55	Available
Video Game	Sun Zi in SupraNova	777	100.00	On Loan

##### A3 Remove an item

This option allows the user to delete an item. The user needs to specify the item no. when prompted. **The implementation of this option should be consistent with those of the basic features and the other additional features.**

**Notes:**

- You should not use any static variables in your program.
- You should think carefully what input is required for each option if there is any.
- You should design your own output.
- You should perform all possible data validation to make your program robust.
- You should implement the additional features only **AFTER** all the basic features have been fully implemented and tested to be fully working.
- **NO MARKS** could be awarded for the additional features if all the basic features have **NOT** been fully implemented (and fully working).
- No additional marks will be awarded for any extra additional features.
- Tutors will not entertain any questions on the implementation of the additional features.

**5. DESCRIPTIVE SECTION**

After your system is implemented, it was decided that the program should now **maintain the serial no. for the items in running order, starting from the value 1**. That is, the first item stored in the application should start with the serial no. 1. The next item added should have the serial no. 2, and so on. Accordingly, Figure 3 – List items should look like:

```
List items
=====
Item  Category      Title           Serial No.      Rental Charge    Status
1     Manga          The Renegade    0001            9.75             On Loan
2     Anime           Flying Pigs     0002            12.40            Available
3     Video Game      Mortar ConBet   0003            23.45            On Loan
```

Discuss the changes you will make to your program.

**Note:** you are not required to implement this feature in your program.

## 6. PROGRAM TEMPLATE

```

/*=====
 * FSP 2014 Assignment
 * Student ID :
 * Student Name :
 * Module Group :
 *
 * Features done:
 * 1. ...
 * 2. ...
 *
 * Descriptive Section:
 * ...( type your answer for the descriptive section here ) ...
 *
 *=====*/
import java.util.*;

public class S10009999Assignment
{
    public static void main(String[] args)
    {
        final int MAX_ITEMS = 15; // maximum number of items
        final String[] categoryTypes = {"Manga", "Anime", "Video Game"}; // the 3 types
        String[] categories = new String[MAX_ITEMS]; // array for the categories
        String[] titles = new String[MAX_ITEMS]; // array for the titles
        int[] serialNos = new int[MAX_ITEMS]; // array for the serial numbers
        double[] rentalCharges = new double[MAX_ITEMS]; // array for the rental charges
        boolean[] statuses = new boolean[MAX_ITEMS]; // array to store the statuses
        int itemCount = init(categories, titles, serialNos, rentalCharges, statuses);

        // ADD YOUR STATEMENTS HERE....

    }

    // DEFINE OTHER METHODS HERE....

}

```

**Figure 9 – Program Template**

## 7. DELIVERABLES

Name the class **S10009999Assignment** where **10009999** represents your student ID.

You are required to create a folder called 'Assignment' in the **FSP network folder** ([\\ictspace\fsp](#)) and submit your work into that folder **in 2 stages**:

### **Stage 1 – Wednesday, Week 16 (6<sup>th</sup> August 2014, 8:30am)**

#### **All the basic requirements (.java and .class files)**

Create a folder called '**Stage 1**' in your 'Assignment' folder. Upload your work for all the basic requirements in the folder. Complete the *block comment* at the top of the program template stating your **student number, name, group and features done**.

**Stage 2 – Friday, Week 17 (15<sup>th</sup> August 2014, 6:00 p.m.)**

**Whole application (.java and .class files)** that you have written for the assignment.

Create a folder called '**Stage 2**' in your 'Assignment' folder. Upload your work for the whole application in the folder. **Remember to provide your answer for the descriptive section in the block comment at the top of the program.**

Your tutor will grade your work based on the soft-copy that you have submitted into the folder.

**8. WALKTHROUGH**

There will be a walkthrough, a written test, to be conducted on **Monday, 18<sup>th</sup> Aug 2014**. The purpose of the walkthrough is to test your understanding of your work. You will be required to answer questions relating to the assignment.

In the walkthrough, you will be asked to give short, written answers to some questions about your assignment. These questions will assess your basic understanding of the code that you are handing in. It is also possible that you will be called to perform a demonstration cum explanation of your work.

If you fail to display adequate understanding of your own program, **your final grade for the assignment can be down-graded by up to two letter grades** (e.g., from A to C).

**You MUST BRING A HARDCOPY OF YOUR PROGRAM PRINTOUT with line numbers for the walkthrough. The printout is to be submitted together with the walkthrough. Ensure that there is no hand-writing or any marking on your listing.**

**The schedule for the walkthrough is as follows:**

**Date: 18<sup>th</sup> Aug 2014**

**Time: 10:00 a.m.**

**Venue: Blk 53 Level 8**



## 9. ASSESSMENT CRITERIA

This assignment constitutes 30% of this module.

Performance Criteria for grading the assignment is as described below. Marks awarded will be based on **program code** as well as student's degree of understanding of work done as assessed during the **walkthrough**.

### A Grade

- ◆ Program implements the *Basic Requirements with input validation* successfully
- ◆ Program implements all three *Additional Requirements* successfully
- ◆ Excellent answer for the modification needed for the descriptive section
- ◆ Program demonstrates good design with the correct use of methods
- ◆ Program provides strong evidence of good programming practice
- ◆ Program has been tested adequately
- ◆ Score at least a 'A' in the walkthrough test

### B Grade

- ◆ Program implements the *Basic Requirements with input validation* successfully
- ◆ Program implements two *Additional Requirements* successfully
- ◆ Correct answer for the modification needed for the descriptive section
- ◆ Program attempts to use methods
- ◆ Program provides sufficient evidence of good programming practice
- ◆ Program has been tested adequately
- ◆ Score at least a 'B' in the walkthrough test

### C Grade

- ◆ Program implements the *Basic Requirements with input validation* successfully
- ◆ Program implements one *Additional Requirement* successfully
- ◆ Satisfactory answer for the modification needed for the descriptive section
- ◆ Program provides some evidence of good programming practice
- ◆ Program has been tested adequately
- ◆ Score at least a 'C' in the walkthrough test

### D Grade

- ◆ Program implements the *Basic Requirements* successfully
- ◆ Program has been tested adequately
- ◆ Score at least a 'D' in the walkthrough test

### NOTE

- *Evidence of good programming practice include the use of meaningful variable names, proper indentation of code, appropriate and useful comments, adoption of standard naming conventions etc.*
- *Basic Input validation refers to the checking of the inputs entered by the user.*