School of Computer Science & Software Engineering
Assignment 2 (Individual)

CSCI213 – JAVA PROGRAMMING AND APPLICATIONS
April 2016 – June 2016

UOW Moderator: Dr Lei Ye
Lecturer: Mr. Aaron Yeo
Email: simcsci213@gmail.com
Weightage: 10% of Subject
Submission: 1st May 2016 2355hrs Wollongong time to Moodle
Demo: 5th May 2016 during Lab 4

CSCI213 Assignment 2: Create A Simple Java Game Application

1. Overview

This assignment aims to establish a basic familiarity with the JDK development system and its associated on-line Java API class documentation. Students should apply the appropriate fundamental programming concepts (such as variables, constants, arrays, strings, methods, selection and repetition constructs etc.) and make use of appropriate Java API classes (such as Scanner, PrintWriter, String etc.) that they have learnt to solve the given problem.

2. Objectives

On completion of this assignment a student should be able to write simple Java application that:

- Makes use of selection and repetition constructs to achieve desired outcomes
- Stores data to and reads data from arrays
- Generates output to and reads input from the console window
- Reads data from and writes data to text file
- Manipulates string using Java API "String" class
- handles basic errors
- Applies object-oriented concepts

3. Scope

This assignment is based on individual effort. You are required to design, develop and test a one player Java game application named "Fishing Pair".

Besides providing the required functionalities, your program should incorporate appropriate error handling. Comments are also to be inserted to improve program clarity. Before you start coding your program, you are strongly advised to carry out proper problem analysis and program design. You are required to use JDK 1.5 developer version or later.

4. Requirements

4.1 Background

Fishing Pair is a turned based card game that can be played by two to five players.
The game uses a standard deck of 52 French playing cards.
A standard deck of 52 French cards consist of four suite; clubs (♣), diamonds (♦), hearts (♥) and spades (♠). Each suite consists of 13 cards with the following symbol; A,2,3,4,5,6,7,8,9,10,J,Q, K.

The objective of the game is to collect as many matching pair of cards with the same symbol as possible. The player with the highest number of matching pair of cards wins the game.

Game Procedure
The game starts by dealer shuffles the deck of cards.
The dealer then place 15 cards facing down on the playing table.

The game starts from player 1, then to player 2 and so on and end with the dealer then to player 1 again until all matching pair are collected.

For each play, each player can choose to reveal the symbol of any two cards on the table.
If the two cards symbol matches, the player collect and keep the two cards.
The dealer will deal another two cards (if there are cards left in the deck) to replace the two empty slots.
The player continues with the game until an unmatching pair appears.

If the two cards symbol does not match, the player lose the turn and can

1. Choose to turn back the cards OR
2. Choose to request the dealer to collect the two cards and place them at the end of the deck and replace the two empty slots with two new cards.

The next player continues with the game.

If there are no more card on the deck, the player may not request option 2

Game end:
At the end of each game, the player with the highest number of matching pair cards win the game.
If there is a tie, there will be no winner for the game.
If the human player wins the game, the human player score increase by 1.

The game will continue with a new round unless the human player chooses to leave the game.

4.2 Minimum required Functionalities

Develop a Java program for the two-players Fishing Pair game described above.

The game starts by the player logging into the game. The players' data (login in, SHA-256 password, last login date and score) are stored in players.dat file from assignment 1.
(The texts in **bold** are data input by the player)

```
FISHING PAIR GAME LOGIN
--------------------------------------------------------------------------------
Enter Login name> IcePeak
Enter Password > password
```

Upon logging in, the number of days since last login will be displayed.
The last login date of the player then is updated to the current date.

```
FISHING PAIR GAME
--------------------------------------------------------------------------------
Welcome back IcePeak, we have missed you for 10 days!
```

The player can choose the number AI players (for enhancement only)

```
FISHING PAIR GAME
--------------------------------------------------------------------------------
Welcome back IcePeak, we have missed you for 10 days!
How many player you want to play with ? (1 to 4) : 1
```

The default AI player is the dealer.

```
--------------------------------------------------------------------------------
Game starts
Dealer shuffles deck.
--------------------------------------------------------------------------------
Dealer deals cards on table
--------------------------------------------------------------------------------
<Card 1> <Card 2> <Card 3> <Card 4> <Card 5>
<Card 6> <Card 7> <Card 8> <Card 9> <Card 10>
<Card 11> <Card 12> <Card 13> <Card 14> <Card 15>

IcePeak Matching Pair: 0
Dealer Matching Pair: 0
Deck of cards: 37 cards
--------------------------------------------------------------------------------
IcePeak’s Turn
--------------------------------------------------------------------------------
Choose first card number to reveal: 2
Choose second card number to reveal: 5

<Card 1> <Ace of Spade> <Card 3> <Card 4> <Ace of Heart>
<Card 6> <Card 7> <Card 8> <Card 9> <Card 10>
<Card 11> <Card 12> <Card 13> <Card 14> <Card 15>

It is MATCH!, IcePeak has fished a Pair!.
--------------------------------------------------------------------------------
Dealer deals two new cards
--------------------------------------------------------------------------------
<Card 1> <Card 2> <Card 3> <Card 4> <Card 5>
<Card 6> <Card 7> <Card 8> <Card 9> <Card 10>
<Card 11> <Card 12> <Card 13> <Card 14> <Card 15>

IcePeak Matching Pair: 1
Dealer Matching Pair: 0
Deck of cards: 35 cards

--------------------------------------------------------------------------------
IcePeak’s Turn
--------------------------------------------------------------------------------
Choose first card number to reveal: 4
Choose second card number to reveal: 11

<Card 1> <Card2> <Card 3> <3 Heart> <Card5>
<Card 6> <Card 7> <Card 8> <Card 9> <Card 10>
<5 Club> <Card 12> <Card 13> <Card 14> <Card 15>

It is not a match :(
1. Turn back the two cards
2. Replace the two cards
Enter choice: 1
--------------------------------------------------------------------------------
Dealer’s Turn
--------------------------------------------------------------------------------
<The game goes on> ……….
--------------------------------------------------------------------------------
IcePeak wins !
IcePeak total score is now 11!

Next Game? (Y/N):Y
```

And the game continues until the player exits the game.

Updates of player’s score and last login date

After each game, the program should update the score and last login date for that player in the players.dat text file immediately.
For example if the current date is 2016-4-9 and IcePeak old score is 10 and IcePeak wins the current game, the file should be updated to:

For example

```
BlackRanger|21a57f2fe765e1ae4a8bf15d73fc1bf2a533f547f2343d12a499d45643453ad4|2016-1-18|10
BlueKnight|e765e4456e4f1ae4ae8bf15d73f435535e4a56f441f2556315a23646473e3454|2016-2-22|15
IcePeak|343a4d56b453c76e5e1ae54a8bf15d73fc1bf2a533f547f2343d19c0592044d4|2016-4-9|11
GoldDigger|bf2a536446464643e32335b3eddff2233433f547f2343d12a49343345ab53c4d|2016-3-8|22
```

Error Handling

Your program should be able to handle error situations like where a player enter wrong password. You should look out for other possible exceptions and handle them too.

5. Submission

A complete submission requires the following items:

1. Report
2. Program in Zip file for execution
3. Program listing in a single word file (for turnitin check)

Missing in any of the above items is consider non-submission.
Late submission for any above items is consider late submission.

Program file header

For each Java program file, provide the header as shown

```
/*
 * CSCI213 Assignment 2
 * --------------------------
 * File name: (state name of .java file)
 * Author: (State student name in FULL)
 * Student Number: (State UOW student number)
 * Description: (A brief description of this class)
 */
```

Any late submission of work must be accompanied by an application for Special Consideration, requested via SOLS. Unless an extension is granted, any late submission will receive a penalty of 25% of its total worth per day including weekends, and will result in zero mark being recorded on or after the 4th late day. Request for extension with supporting document must be submitted to SIM administration for further consideration before the submission date and the tutor must be informed. Extension will be granted on a case-by-case basis.

Report requirements

The report to be submitted consists of the following sections:

1. Cover page – clearly state your name and UOW student's number.
2. Content Page
3. Classes design
4. Test-run of Program: You have to provide screen outputs to show the correct execution of your program according to the requirements stated.
5. Error Handling: List down and explain the errors and exceptions that your program can handle. Provide screen captures here.
6. Enhancement: List down and explain the enhancement that your have done. Provide screen captures here
7. Conclusion and Reflection

Testing requirements

Make sure that you are able to run your program using the command (***java FishingPairGame***) directly from the command prompt from the project folder.

In the players.dat, you should have the following FOUR players' data.

Login Name	Password	Last login date	Scores
player1	p1	2016-1-11	10
player2	p2	2015-12-2	12
player3	p3	2016-2-7	15
player4	p4	2016-3-15	9

Refer to Report guideline and marks allocation document for detailed report requirements.

6. Plagiarism

The University's policy on copying does not allow you to copy software as well as your assessment solutions from another person. Copying of another person's work is unacceptable. It is the responsibility of all students that their assessment solutions are their own work. You must also ensure that others do not obtain access to your solutions for the purpose of copying a part of them. **Where such plagiarism is detected, both of the assessments involved will receive ZERO mark.**

7. Evaluation Criteria (total 50 marks)

Refer to Report guideline and marks allocation document