

## COM410 Software Development II

### Assignment 2: "Higher/Lower" Card Game

Submission Deadline: 12:00pm (Noon) on Tuesday 12<sup>th</sup> April 2022 (Week 12)

This assignment carries 60% of the overall coursework mark for the module.

#### Introduction

**Higher/Lower** is a single-player card game that uses a standard shuffled deck of playing cards from which a number of cards have been dealt face-down. The object of the game is to turn over the first card revealing its value and then to predict whether the next card to be turned over is higher or lower in value than the previous. For example, if the game consists of 5 cards, the initial state of play might be as shown below, where the 4♥ is visible and the remaining 4 cards are face-down from left-to-right.

4♥ 

Your score = 0

The player is now invited to say whether they think that the next card will be higher or lower than a 4. In this instance they might guess "**higher**" and the next card is revealed as the 9♣. As the user's guess was correct, the game continues.

4♥ 9♣ 

Correct guess, your score = 1

This time the user guesses that the next card will be lower than a 9, but when the card is turned over, it is revealed to be the 10♠. As they have guessed incorrectly, the game is over, and their final score is confirmed.

4♥ 9♣ 10♠ 

Incorrect guess, your final score = 1

If the player successfully turns over all cards with correct guesses, the game automatically ends. Where consecutive cards are the same value, for example Q♥ followed by Q♣, then the player is guaranteed to lose as the second Queen is neither higher nor lower than the first one.

In this game, the Ace is the highest value available so that the values in order from lowest to highest are 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A.

Note that your application will be a pure text version – with keyboard input. There is no requirement for a graphical or mouse-based solution.

#### Required Functionality

There are 7 levels to which the submission can be completed – attracting increasing rewards.

**Level 1:** The facilities to create a new game are present and the game can be set up in its initial state, with 5 cards dealt from a shuffled deck, one of which is shown face-up

(i.e. its value is on display) with the remaining cards face-down. You can use any appropriate text representation for the cards

- Level 2:** The basic game mechanism is in place. Users can select “higher” or “lower” when each card is displayed, and the next card is revealed or the game ends depending on their choice.
- Level 3:** A full playable game is available. The application is able to prompt the player when the game is won (all cards successfully guessed) or lost (a wrong guess has been made).
- Level 4:** Before the game begins, the user is prompted for the number of cards to be used. For example, if the user chooses 10 cards, then one is displayed face up and 9 correct guesses need to be made for a winning game.
- Level 5:** The user can “stick” by declining to guess either “higher” or “lower”. In this case the game is won, and the user is awarded a score according to the number of correct guesses made so far. A high-score table is displayed after each game showing the top 5 winning performances so far.
- Level 6:** On completion of a game (whether the player has won or lost) the application is able to replay the game guess by guess, with the user prompting each replayed move by a keypress.
- Level 7:** The application is able to play a complete game in “demonstration mode”, where all 52 shuffled cards are dealt. In demonstration mode, the user’s only input is to press a key to begin the next move and the demonstration continues until an incorrect guess is made or all 52 cards have been successfully revealed. A separate “Demonstration” high score table should show the computer’s top 5 demonstration performances.

## Deliverables

The deliverables for this assessment are as follows:

- 1. “HigherLower” Application** – A Zip file containing all code developed for the application. The easiest way to create this is to Zip the contents of the **src** folder of your Java project. Each Java file should also be provided as a PDF document.
- 2. Design and Development Document** – A PDF document that contains a written account of the design and development decisions made during the development of your application. You should provide evidence of your consideration of data structures used and algorithms developed, with justification for your decisions and rejection of alternatives.
- 3. Testing Document** – A PDF document that describes your testing regime with details of any test classes developed. This document should also describe steps you have taken to ensure the security of your implementation.
- 4. Demonstration Video** – An individual video of no more than 5 minutes duration, demonstrating a walkthrough of your application running and a walkthrough of your

code base. You do not have to appear on video in person, but a voiceover commentary is mandatory. Any screen recording package can be used **as long as the video is in MP4 format**. Please note that videos with a duration in excess of 5 minutes will be penalised according to the following scheme.

<b>5 minutes or less</b>	no penalty
<b>5 minutes + 10%</b>	reduction in total mark by 10%
<b>5 minutes + &gt;10% - 20%</b>	reduction in total mark by 15%
<b>5 minutes + &gt;20% - 30%</b>	reduction in total mark by 20%
<b>5 minutes + &gt;30% - 40%</b>	reduction in total mark by 25%
<b>5 minutes + &gt;40%</b>	maximum total mark achievable is 40%

### Restriction

The Collection classes from the Java Class Library **MUST NOT** be used within a submission. Any lists, stacks, queues, bags or other data structures must be implemented from scratch as part of the submission. Java **Arrays** can be used but **ArrayLists** are **not** available. **Elements of the application that make use of the Collection Classes will not count towards the submission and will not contribute to the mark.**

### Submission

All 4 deliverables must be submitted to the Assignment 2 link on Blackboard by the due date. Late submissions can only be accepted through the standard EC1 process.

### Feedback

Feedback on each final submission will be provided in written form under the categories presented in the marking criteria that follows this specification. All results and feedback will be returned by **Friday 20th May 2022**.

### Plagiarism Declaration

By making a submission, you will be deemed to have made the following declaration of ownership. Source: <http://www.ulster.ac.uk/academicservices/student/plagiarism.pdf>.

*"I declare that this is my own work and that any material I have referred to has been accurately and consistently referenced. I have read the University's policy on plagiarism and understand the definition of plagiarism as given in the [course/subject] handbook. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file."*

## Submission Assessment Grid

Student:					
	Poor (<40%)	Satisfactory (40-59%)	Good (60-69%)	Excellent (70%+)	Mark
<b>Design and planning (15 marks)</b>	Limited understanding of fundamental ADT design concepts. Minimum set of operations defined. Lack of discussion of ADT design and/or poor justification for ADT design choices	Some understanding of fundamental ADT design concepts shown. Adequate set of operations defined. Adequate data storage representation design demonstrated.	Majority of ADT operations defined with appropriate data storage representation design demonstrated. Detailed discussion of ADT design with good justification for design choices.	Demonstrates excellent awareness of alternatives for ADT selection and good justification for decisions made. Most efficient representations for ADTs selected and clearly justified.	
<b>Implementation (15 marks)</b>	Limited knowledge of ADT development techniques shown. Minimum set of ADT operations demonstrated. Major issues with implementation of data storage representation.	Some knowledge of ADT development techniques. Adequate set of ADT operations clearly demonstrated. Some issues with implementation of data storage representation.	Good knowledge of ADT development techniques. Majority of ADT operations clearly demonstrated. Minor issues with implementation of data storage representation.	Excellent knowledge of ADT development techniques shown. Complete set of ADT operations clearly demonstrated. Insightful discussion of most technically challenging aspect	
<b>Functionality (40 marks)</b>	<b>Less than L1.</b> A game that is not playable with significant components missing or not working.	<b>L1.</b> The facilities to create a new game are present and the game can be set up in its initial state, with 5 cards dealt from a shuffled deck. <b>L2.</b> The basic game mechanism is in place. Users can select “higher” or “lower” when each card is displayed, and the next card is revealed or the game ends depending on their choice. <b>L3.</b> A full playable game is available. The application can prompt the player when the game is won (all cards removed) or lost (a wrong guess has been made).	<b>L4.</b> Before the game begins, the user is prompted for the number of cards to be used. For example, if the user chooses 10 cards, then one is displayed face up and 9 correct guesses need to be made for a winning game. <b>L5.</b> The user can “stick” by declining to guess higher or lower. In this case the game is won, and the user is awarded a score according to the number of correct guesses made so far. A high-score table is displayed after each game showing the top 5 winning performances so far.	<b>L6.</b> On completion of a game (whether the player has won or lost) the application is able to replay the game guess by guess, with the user prompting each replayed move by a keypress. <b>L7.</b> The application can play a complete game in “demonstration mode”, where the user’s only input is to press a key to reveal the next move and the demonstration continues until an incorrect guess is made or all 52 cards have been successfully revealed. A separate Demonstration High Score Table is provided.	
<b>Testing (20 marks)</b>	No evidence of testing or no evidence that testing has been effective.	Details of basic testing are present but no evidence of a structured approach. Limited documentation of the testing process.	Some test classes have been provided, but with less than full coverage. Incomplete testing documentation.	All main classes have test classes that give proper attention to the range of potential input values including edge cases. A comprehensive write-up that fully documents the testing process.	
<b>Video (10 marks)</b>	No video or no voiceover.	Video is present but is short or is missing either a demonstration of the application or a walkthrough of the main code elements.	Video demonstrates functionality and discusses the main code elements – but does not provide any insight beyond what can be seen by running the software.	Video provides additional insight into the development process. Excellent demonstration of working functionality.	
		<b>Initial Mark:</b>	<b>Deductions:</b>	<b>Total Mark:</b>	