

## COM410 Programming in Practice

### Assignment 2: Individual Development Exercise

Submission Deadline: 12pm (Noon), 15<sup>th</sup> December 2023 (Week 12)

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

#### Introduction

**Make 11** is a one-player card game played against the computer. The game uses a standard shuffled deck of playing cards from which the player is dealt a hand of 5 cards. In each round of the game, the computer deals a card face-up from the deck and the player attempts to add a card from his hand such that the total rank score of the two cards is 11. In this game, the rank value of the picture cards (Jack, Queen, and King) is 10 and the value of the Ace is one. If the player is successful, a point is scored, and the player's card is replaced with one dealt from the deck. If a player can **Make 11**, they can also opt, in the same turn, to play any picture cards in their hand to exchange them for replacement cards dealt from the deck. If the player cannot **Make 11**, they can play a card of the same suit to allow the game to continue, but no point is scored. The game ends when a player is forced to play a card that does not **Make 11** and is not of the same suit.

For example, a game might proceed as follows.

Round	Player's Hand	Computer	Player	Score	Commentary
1	5♠ 10♦ J♣ Q♠ A♦	A♣	10♦	1	Player can <b>Make 11</b> . 10♦ is replaced in the player's hand
2	5♠ 9♥ J♣ Q♠ A♦	6♦	5♠	1	Player can <b>Make 11</b> . 5♠ is replaced in the player's hand
3	2♦ 9♥ J♣ Q♠ A♦	8♣	J♣	0	Player cannot <b>Make 11</b> but can play J♣ to continue. J♣ is replaced in the player's hand
4	2♦ 9♥ K♥ Q♠ A♦	9♠	2♦ K♥ Q♠	1	Player can <b>Make 11</b> . 2♦ is replaced in the player's hand. Player also opts to replace K♥ and Q♠
5	4♦ 9♥ 10♠ J♠ A♦	3♦	4♦	0	Player cannot <b>Make 11</b> but can play 4♦ to continue. 4♦ is replaced in the player's hand
6	3♥ 9♥ 10♠ J♠ A♦	7♣	3♥	0	Player has no moves that permit the game to continue, so plays any card from their hand and the game ends.

FINAL SCORE: 3

## Challenge

This task is to produce a complete implementation of **Make 11**. In each round, the player's hand of cards is displayed along with the card played by the computer. The player is invited to select one of their cards to **Make 11** with the computer's card. If a **Make 11** move is possible, the player may also select any picture cards from their hand to have them replaced with new cards from the deck. At the end of each round, the player's score for that round should be displayed, along with the current total score.

**Note:** A demonstration video on Blackboard will illustrate an example submission application running.

## Required Functionality

There are 8 levels to which the submission can be completed – attracting increasing rewards.

**Level 1:** A player is dealt 5 cards from a shuffled deck. The hand of cards is displayed in the console.

**Level 2:** The computer deals a card from the deck and the user selects a card from the hand to **Make 11**. If **Make 11** is achieved, the user scores one point and their card is replaced with one from the deck.

**Level 3:** The game continues until the player is unable to **Make 11**, when the total score is displayed.

**Level 4:** If a player is unable to **Make 11** but can play a card of the same suit as the computer, the game can continue, but no point is scored for that round.

**Level 5:** If the player can **Make 11**, they can opt to also play any picture cards in their hand so that they are replaced by cards from the deck.

**Level 6:** All user input is validated so that only valid input is possible. An empty deck causes the game to end.

**Level 7:** A high-score table holding the player's name and score for the 5 highest scores is implemented. If a new game results in a score that qualifies for the high score table, the player is asked for their name and the new entry is inserted into the appropriate place in the table. Where the new score equals the lowest score in the table, it should replace the existing entry. The high score table is made available for display before each new game is played and after a new high score entry is recorded.

**Level 8:** Following each game, the player is offered the opportunity to view a replay which displays for each round the player's hand, the computer's card, the card(s) played by the player and the outcome.

## Deliverables

The deliverables for this assessment are as follows:

1. **"Make-11" 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 with an associated commentary concentrating on the parts of the code for which you were responsible. You do not have to appear on video in person, but a voiceover commentary is mandatory. Videos should preferably be recorded using Panopto (available in the module area on Blackboard), though other packages may be used provided the video is in MP4 format. Please note that videos with a duration longer than 5 minutes will be penalised according to the following scheme.

<b>5 minutes + 10%</b>	no penalty
<b>5 minutes + 10% - 19%</b>	reduction in total mark by 5%
<b>5 minutes + &gt;20% - 29%</b>	reduction in total mark by 10%
<b>5 minutes + &gt;30% - 39%</b>	reduction in total mark by 15%
<b>5 minutes + &gt;40% - 49%</b>	reduction in total mark by 20%
<b>5 minutes + &gt;=50%</b>	maximum total mark achievable is 40%

## Submission

All 4 deliverables must be submitted to the Assignment 2 links 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 19<sup>th</sup> January 2023**.

## 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
Design and planning (15 marks)	Limited understanding of fundamental ADT and algorithm design concepts shown. Minimum set of operations defined Lack of discussion of ADT and algorithm design and/or poor justification for design choices made	Some understanding of fundamental ADT and algorithm design concepts shown. Reasonable approach to design but lacking detailed analysis and evidence of decision making.	Discussion of ADT and algorithm design with good justification for design choices made	Demonstrates excellent awareness of design alternatives and clear, logical justification for decisions made.	
Implementation (15 marks)	Limited knowledge of ADT and algorithm development techniques shown. Minimum set of ADT operations demonstrated. Major issues with implementation of data representation.	Some knowledge of ADT and algorithm development techniques. Adequate set of ADT operations clearly demonstrated. Some issues with implementation of data representation.	Good knowledge of ADT and algorithm development techniques. Majority of ADT operations clearly demonstrated. Minor issues with implementation of data representation.	Excellent knowledge of ADT and algorithm development techniques shown. Wide set of appropriate ADT operations clearly demonstrated. Insightful discussion of most technically challenging aspects.	
Functionality (40 marks)	Less than L1. A game that is not playable with significant components missing or not working.	L1. The player is dealt 5 cards from a shuffled deck. The hand of cards is displayed in the console.  L2. The computer deals a card from the deck and the user selects a card from the hand to <b>Make 11</b> . If Make 11 is achieved, the user scores one point and their card is replaced with one from the deck.  L3. The game continues until the player is unable to <b>Make 11</b> , when the total score is displayed.	L4. if a player is unable to Make 11 but can play a card of the same suit as the computer, the game can continue, but no point is scored for that round.  L5. if the player can Make-11, they can opt to also play any picture cards in their hand so that they are replaced by cards from the deck.  L6. All user input is validated so that only valid input is possible. An empty deck causes the game to end.	L7. A high-score table holding the player's name and score for the 5 highest scores is implemented.  L8. Following each game, the player is offered the opportunity to view a replay which displays for each round the player's hand, the computer's card, the card played by the player and the outcome.	
Testing (20 marks)	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.	
Video (10 marks)	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.	
		Initial Mark:	Deductions:	Total Mark:	