

COM498 Algorithms and Data Structures

Assignment 2: Individual Development Exercise

Submission Deadline: 12pm (Noon), 16th December 2022 (Week 12)

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

Introduction

STREAK is a one- or two-player card game that uses a standard deck of playing cards where each player is dealt a hand of cards. The object of the game is to have the longest possible streak of consecutive ranks when the hand is sorted into ascending order of rank. The number of points awarded to a player is the length of the streak achieved, with a bonus of one point if the streak is of cards of a single colour and a bonus of two points if the streak is of cards of a single suit.

For example, for a sorted hand of 5 cards, scoring might be as follows

Hand	Score	Commentary
5♠ 10♦ J♣ Q♠ A♦	3	The consecutive 3-card streak consisting of 10, Jack, Queen
5♦ 10♠ J♣ Q♠ A♦	4	The consecutive 3-card streak consisting of 10, Jack, Queen plus a bonus point for a run of a single colour
5♦ 10♠ J♠ Q♠ A♦	5	The consecutive 3-card streak consisting of 10, Jack, Queen plus two bonus points for a streak of a single suit

After the cards have been dealt and sorted, a player can attempt to improve their longest streak by nominating a card from their hand to be exchanged for one dealt from the remaining cards in the deck. The nominate and exchange process can be repeated for as many times as cards that were originally dealt. For example, in a 5-card game, up to 5 cards can be exchanged; in a 6-card game, up to 6 cards can be exchanged, and so on.

In a one-player game, a single hand of cards is dealt and played, but a two-player game consists of three hands where the winner is the player with the highest cumulative score after all hands have been played.

Challenge

This task is to produce a computer-based version of **STREAK** for **1 or 2 players** and a **hand size of between 5 and 10 cards**. After the number of cards in a hand has been selected, the hand is dealt, and the player's cards are sorted by ascending rank and are revealed. A player can then nominate a card to be replaced by one dealt from the deck in a process that can be repeated a number of times up to the size of the hand. In a 2-player game, the first player makes all their nomination and exchange selections and receives their score before the second player takes their turn.

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

For example, a 2-player game with 5 cards might proceed as follows (note that a fresh shuffled deck is generated for each hand that is dealt):

HAND 1:

Player1

2♠ 3♦ 5♠ 7♣ K♣	(longest streak is 2)	Player1 exchanges K♣
2♠ 3♦ 5♠ 6♣ 7♣	(longest streak is 4 – including colour bonus)	Player1 exchanges 2♠
3♦ 5♠ 6♠ 7♣ A♦	(longest streak is 4 – including colour bonus)	Player1 exchanges A♦
3♦ 5♠ 6♠ 7♣ J♦	(longest streak is 4 – including colour bonus)	Player1 exchanges J♦
3♦ 5♠ 6♠ 6♦ 7♣	(longest streak is 4 – including suit bonus)	Player1 exchanges 6♦
3♦ 4♦ 5♠ 6♠ 7♣	(longest streak is 5)	All 5 exchanges done

Player2

6♦ 7♦ 10♣ K♣ A♣	(longest streak is 4 – including suit bonus)	Player2 exchanges 10♣
2♣ 6♦ 7♦ K♣ A♣	(longest streak is 4 – including suit bonus)	Player2 exchanges 2♣
6♦ 7♦ Q♣ K♣ A♣	(longest streak is 5 – including suit bonus)	Player2 sticks

Current score

Player1 5 Player2 5

HAND 2

...

HAND 3:

...

Final score

Player1 10 Player2 8

Required Functionality

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

Level 1: A single player is dealt 5 cards from a shuffled deck which are displayed sorted in ascending order of rank.

Level 2: The application automatically reports the length of the longest streak of successive ranks.

Level 3: The user can change up to 5 cards, one at a time, with the aim of creating the longest possible streak of successive rank values.

- Level 4:** A number of points is calculated as the length of the longest streak. A bonus point is awarded if the longest streak consists of cards of the same colour, while two bonus points are awarded if the longest streak consists of cards of the same suit.
- Level 5:** The number of cards (both that to be dealt and the maximum number that can be changed is variable in the range 5-10. The number of cards is selected by the player before the hand is dealt.
- Level 6:** A 2-player game is provided where the first player plays out their turn and receives a score before the second player begins. The winner of a 2-player game is that with the highest total score after 3 hands are played.
- Level 7:** A high-score table holding the name and score for the 5 best single-hand performances (i.e., the 5 best streaks when colour and suit bonuses are included) is implemented. The high-score table can be viewed on demand before and after each match.
- Level 8:** Following a single-player game (only), the player is offered the opportunity to view a replay in which each hand and choice of card to be replaced is displayed.

Deliverables

The deliverables for this assessment are as follows:

1. **“Streak” 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.

5 minutes + 10%	no penalty
5 minutes + 10% - 19%	reduction in total mark by 5%
5 minutes + >20% - 29%	reduction in total mark by 10%
5 minutes + >30% - 39%	reduction in total mark by 15%
5 minutes + >40% - 49%	reduction in total mark by 20%
5 minutes + >=50%	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. **Elements of the application that make use of the Collection Classes will not count towards the submission and will not contribute to the mark.**

Note: The Java **Array** type can be used, but **ArrayList** is not available.

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 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 design concepts shown. Minimum set of operations defined Lack of discussion of ADT design and/or poor justification for ADT design choices made	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 made	Demonstrates excellent awareness of alternatives for ADT selection and good justification for decisions made. Most efficient representations for ADTs selected and clearly justified.	
Implementation (15 marks)	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	
Functionality (40 marks)	Less than L1. A game that is not playable with significant components missing or not working.	L1. A single player is dealt 5 cards from a shuffled deck which are displayed sorted in ascending order of rank. L2. The application automatically reports the length of the longest streak of successive ranks. L3. The user can change up to 5 cards, one at a time, with the aim of creating the longest possible streak of successive rank values.	L4. A number of points is calculated as the length of the longest streak. A bonus point is awarded if the longest streak consists of cards of the same colour, while two bonus points are awarded if the longest streak consists of cards of the same suit. L5. The number of cards (both that to be dealt and the maximum number that can be changed is variable in the range 5-10. The number of cards is selected by the player before the hand is dealt.	L6. A 2-player game is provided where the winner is that with the highest total score after 3 hands are played. L7. A high-score table holding the name and score for the 5 best single-hand performances is implemented. The high-score table can be viewed on demand before and after each match. L8. Following a single-player game (only), the player is offered the opportunity to view a replay in which each hand and choice of card to be replaced is displayed.	
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:	

