Poker Dice

Instructions: Unless it has been done for you, print a copy of this document for your pair. As you work through it, write answers to any questions or prompts that are in **boldface**.

Your names:

Learning Objectives

Content Objectives

Parentheses below correspond to part of the knowledge units in the ACM's *Computer Science Curricula 2013*.

After completing this activity, students should be able to:

- Devise and implement linear-time algorithms (AL/Analysis of algorithms).
- Write JUnit tests (SDF/Development methods).

Process Objectives

After completing this activity, students should have improved their ability to:

- Program in a pair. [Management]

Model 1: Playing the Game

Open the Poker Dice project in Eclipse. Run PokerDice.jar to play the game. It's up to you whether you want to read the rules or play the game first, but do both.

Rules

Players: 2-5

Equipment: Five special dice for each player, with the faces 9, 10, Jack (J), Queen (Q), King (K), and Ace (A).

Object: Get the best *Poker* hand.

Play: Roll all of your dice. Decide which ones you want to keep and re-roll the rest. One more time, decide which ones (out of all five) you want to keep and re-roll the rest. You may choose to stop early and keep all three dice, but you get a maximum of three rolls. After all players have done this, the best hand wins.

Hands: The following hands are possible in *Poker Dice*. They are listed from best to worst.

Hand	Example	Description
5 of a kind	Q Q Q Q Q	All five dice match
4 of a kind	9 9 9 9 K	Four dice match
Full house	J J J 10 10	Three of one rank, two of another
Straight	9 10 J Q K	Consecutive sequence of five
3 of a kind	10 10 10 9 Q	Three of one rank
2 pair	9 9 J J K	Two each of two ranks
1 pair	J J 9 Q K	Two of one rank
High card	9 10 J K A	Anything else

If two players have hands in the same category, ties are broken by the values of the cards, with larger matching sets counted first. For example, Q Q Q J J beats 9 9 9 K K.

1. **Which hand is better: Q Q Q Q J or K K K Q Q?**
2. **Which hand is better: J J Q Q 9 or Q Q 9 9 A?**
3. **Does the program count 10 A Q K J as a straight?**
4. **Does the program sort each player's dice in increasing order, in decreasing order, or not at all?**

Model 2: Overview

Examine the various Java classes.

5. **The dice are displayed as 9, 10, etc., but they are represented differently in Hand. What number represents a Jack?**

6. **Examine the tests in HandTest and the comments on the methods in Hand. What value would be given to a hand whose `dice` array contained the values 0, 1, 4, 0, 1?**

Model 3: `fiveOfAKindScore`

7. **Examine `testFiveOfAKindScore` in HandTest. Does it contain an example of a hand that has five of a kind?**

8. **Does it contain an example of a hand that does not have five of a kind?**

9. **Does it contain every possible five of a kind?**

10. **If *n* is the length of the array `dice`, what is the order of the running time of the `fiveOfAKindScore` method?**

Model 4: `fourOfAKindScore`

11. **What is the order of the running time of `counts`?**

12. **What is the order of the running time of `fourOfAKindScore`?**

Complete `testFourOfAKindScore`. The working implementation of `fourOfAKindScore` should pass the test.

13. **Explain how you could add, remove, or change *one character* in `fourOfAKindScore` to make it fail your test.**

Model 5: Remaining methods

For each of the methods listed below, complete the corresponding test, fail the test, and then write code that passes the test. Each of these methods must run in time linear in the length of `dice`. This means that it's okay to iterate through `dice` a fixed number of times, but not (for example) to use a loop that considers each pair of indices.

- `fullHouseScore`
- `straightScore`
- `threeOfAKindScore`
- `twoPairScore`
- `onePairScore`
- `highCardScore`

Be sure to play the game (by running PokerDice.java) a few times after you're done to make sure the entire system is working correctly.