|  |  |  |
| --- | --- | --- |
| **Student Ids** | Comments | Mark /30 (functionality + code) |
| 1100161 | **Classes**: No output presented for cut, or shuffle so unable to judge correct functionality. Cards dealt correctly for game. Game crashes after a few plays with an array out of bound error. Sensible selection of classes for game of gin rummy. None-optimal card class. Use of ArrayList for the game. Generally neatly written code which is well commented.  **Inheritance**: No polymorphism implemented. All functionality in the player class. Pieces move and jump correctly. Code reasonably well laid out. No screenshots indicating the game terminates successfully. | 10+8=18  12+4=16  Total=34/60 |
| 1164518 1177244 | **Classes:** Text file output supplied indicating shuffle/deal and an interactive game working correctly. Crashes if incorrect selection made. Needs more robustness. Very neatly laid out code. Can’t understand inheritance relationship used.  **Inheritance**: Full functionality with advanced player implemented. Moves and jumps working correctly. Well written code with polymorphism implemented. Some good use of exceptions in the board class. A nice piece of work. | 13+8=21  18 + 9=27  Total = 48 |
| 1188775 | **Classes:** Some functionality. Cut shuffle and deal working. No implementation of the game made. Card class over complex but ToString() method good. No necessary to have several card arrays in the DeckOfCards class. This confuses classes and objects. The Player class is sensible. Test class OK with simple user interface.  **Inheritance:** Full functionality although multiple jumps doesn’t work. Correct use of polymorphism. Well coded although needs more commenting. | 11+4=15  14+6=20  Total=35 |
| 1289495 | **Classes**: Full functionality. Cut, shuffle and deal working. Game also seems to be working. A lot of uncommented code written. Good class selection although arbitrary use of statics. Some redundant members of classes which is inefficient .Not sure what the Poker class is for.  **Inheritance**: Full functionality. Moves and jumps working OK. Correct use of polymorphism. Advanced and Basic player code seems similar. No commenting but code well laid out. | 15+6=21  15+6=21  Total=42 |
| 1153609 1169776 | **Classes:** Full functionality. Game runs OK. No screen dump to verify if automatically detects winning hand. Card class strings should be static. Good use of ArrayList. Hand objects should be part of Player objects. Code OK but needs more and clearer commenting.  **Draughts:** No functionality**.** The board is displayed but and 1 or 2 moves are made and then the program hangs**.** No implementation of polymorphism. All functionality in the Player class. | 15+6=21  5+3=8  Total=29 |
| 1193801  1216608 | **Classes:** Some functionality although game crashes after a few plays. Poorly laid out code with functionality in the Deck’s deal method for some strange reason. Not object oriented and the design given in the lectures has not been followed. Use of generic ArrayList. Some code commenting but very poorly and messily laid out making it unreadable.  **Draughts:** Not quite full functionality. Multiple jumps not implemented and some moves are skipped. But single jumps seem to work OK. Code added to the Player class which was supposed to be abstract. No separate Basic player. The set method not necessary as this should be done in the constructor. Polymorphism implemented for the Advanced player. | 13+3=16  13+5=18  Total=34 |
| 1289484 | **Classes:** Some functionality but much of the submitted code does not compile. A very messy and complicated design with classes that have little or no functionality and which are named like methods (eg Discard). The design laid out in the lectures has not been followed. A poor effort.  **Inheritance:** No real functionality. The board displays but no moves are possible. Polymorphism has not been implemented and the submitted code has compilation errors. Again the design does not follow that laid out in lectures. Why does the Player class extend the Draughtboard class? | 9+3=12  2+3=5  Total=17 |
| 1044925 | **Classes:** Some functionality. Shuffle cut and deal working. No game functionality. Not sure what the *Output* member of the Card class is doing. The suit and rank members are redundant as they are computed in public methods. Why 2 Card arrays in the Deck class? Sensible implementation of Player.  **Inheritance:** Fullfunctionality except for multiple jumps not working. Correct implementation of polymorphism. Well laid out code with commenting. Nice use of private methods in the Player classes. | 11+5=16  16+8=24  Total=40 |
| 1160485 1150589 | **Classes:** Seems like full functionality for the game although the shuffle and cut methods of the deck have not been demonstrated. From the example output shown,it seems that a winning hand was ignored. Well laid out code with sensible class selection but most instance fields have been made public thus ignoring encapsulation so this is not a fully object oriented solution.  **Inheritance**: Seems like full functionality although can’t detect any difference in strategy between basic and advanced players.Polymorphism has not been implemented and it doesn’t look like you understand it. Did you attend the lecture where I went through this lab exercise and explained it? Over complex design. Why the BasicB and BasicW inheriting from the Basic player. No need for separate white and black player classes! | 15+6=21  14+5=19  Total=40 |
| 1289255 | **Classes:** Cut, shuffle and deal and also the game correct output has been demonstrated. 3 instance fields of Card are not necessary. Incorrect use of static in Deck class and String[] suits in the Deck should be in the Card class. The PlayGinRummy class contains all the functionality for the game. There should be a separate Player class as outlined in the lectures. Code layout satisfactory but needs more commenting.  **Inheritance:** Full functionality. Move and jumps working OK**.** No need for separate white and black player classes. Polymorphism correctly implemented. Nicely laid out code although a distinct lack of commenting making complex code difficult to understand and maintain. | 15+6=21  15+6=21  Total=42 |
| 1082605 1113273 | **Classes:** Functionality not clear as computer hand not displayed so not sure if it runs to a winning hand. However, the human player functionality seems OK**.** Inefficient design of Card class. Some instance fields not referenced in the class. Over complex design with too many classes. A Discard class not required. This functionality should be in the Deck class. You are confusing classes and methods! Over use of inheritance. Sorting the hand is sensible but you have made sorting more difficult by a poor design of the Card class. Good effort with the coding. More commenting would help its readability.  **Inheritance:** Functionality OK. Advanced player not implemented but basic player works to completion. Polymorphism correctly implemented. Unnecessary to have separate white and black player classes. Well laid out and commented code. | 15+6=21  13+6=19  Total=40 |
| 1312199 | **Classes:** Functionality seems OK for cut, shuffle and deal and also playing the game interactively. It doesn’t seem that the player objects have much functionality. They should determine the ‘play’ and hence adjust their hand of cards. Well laid out code but a definite lack of commenting.  **Inheritance:** Moves and jumps working correctly. The scoring doesn’t seem to work correctly and has probably been re-implemented. Also the game continues after all pieces have been taken. Polymorphism correctly implemented Well laid out code and good commenting. | 14+6=20  14+7=21  Total=41 |
| 1180936 1215645 | **Classes:** Good functionality demonstrated. Useful screenshots also presented. The Deck class contains the functionality to play the game. This is not correct design. The deck class is a single deck of cards. Game functionality should be in the game class. Apart from that, a good selection of classes. Well laid out and commented code.  **Inheritance:** Minimal functionality. The program crashes after a single incorrect play. It seems that polymorphism has been correctly implemented. However, there is no need for separate black and white player classes. | 15+7=22  5+5=10  Total=27 |
| 1115064 | **Classes:** Some functionality demonstrated but the Deck class crashes after the cards are cut. Shuffle and deal not demonstrated. Use of Strings in the Card class is very inefficient. This was explained in lectures. Untidily laid out code but there is some commenting. Player and game classes implemented but they contain little functionality.  **Inheritance:** Can’t find anything submitted. | 8+4=12  0  Total=12 |
| 1205221 1138870 | **Classes:** Good functionality demonstrated. Useful screenshots presented but nothing to show if winning hand achieved. The class getInputFromUser is unnecessary and bloats the design. It is obviously an action and hence should be a method of a class! Use of Strings in the card class very inefficient. The Deck class has correct functionality. The game class has functionality that should be delegated to the player class. This was outlined in the lectures.  **Inheritance:** Minimal functionality as the game halts after a couple of plays. Polymorphism correctly implemented although the Player class should have been left with no functionality. Generally well laid out code with good commenting | 14+6=20  8+5=13  Total=33 |
| 1072835 1036854 | **Classes:** Cut, deal and shuffle demonstrated for the deck class. The printInfo method of card could be much simpler with a bit of thought. The deckOfCards class does not need 3 Card arrays. This demonstrates a misunderstanding about classes, objects and object oriented programming. An efficient design for these classes was outlined in lectures.  **Inheritance:** No functionality as no piece is moved. No need for separate Black and White player classes. No implementation of advanced player although it looks like polymorphism has been correctly implemented. | 12+5=17  6+5=11  Total=28 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |