|  |  |  |
| --- | --- | --- |
| **Student Ids** | Comments | Mark /100 |
| 998449 904858 | * Classes   Something resembling an OO design although no individual card class. Very messy code with randomly named variables and no commenting. The program goes wrong after 2 turns. At least the shuffle and deal methods seem to work. Your screen dumps are OK. Not a bad attempt for a first program.   * Inheritance   Program seems to work OK. Polymorphism correctly implemented. Over complex code for moveBlack and moveWhite and again poor commenting. No need for additional ‘take’ argument. Ludicrous complexity in AdvancedPlayer. There are much simpler ways to code a fairly complex strategy. Screendumps OK. Again a good effort overall. | 60  62 |
| 1207389  1098640 | * Classes   Well written and good design. Your program seems to work although I couldn’t test its ‘endgame’ functionality. You should have shown that in your screen dump. Good use of Arraylist and the Comparable interface in your design   * Inheritance   Good functionality demonstrated although the game seems to reach a stalemate with the AdvancedPlayer algorithm. Well coded and I liked the separate method to check for out of bounds error. Would have been really neat to use an exception. Good work. | 75  75 |
| 1087793 1098622 | * Classes   Some functionality but you have not submitted any screenshots to demonstrate that the game runs to completion. Strange design. Why does Player extend the DeckofCards class? I went through a complete design of this exercise in class so you clearly weren’t attending! Your cut and deal methods seem to work OK. Your design of the Card and DeckofCards classes are very inefficient.   * Inheritance   Your source code would not compile so no functionality demonstrated. You clearly have not been attending lectures where I went through the design of the program. You have submitted a bit of code for which I can give you some credit. | 50  10 |
| 1081899 1095531 | * Classes   Full functionality. Your design is accurate, fully object oriented and well implemented. Good code, well commented. Avery good effort for a first exercise. Clear example output produced.   * Inheritance   Fully functionality for basic and advanced player. I wasn’t too keen on you adding extra code to the Player class and it shouldn’t be necessary. Why is the advancedPlayer\_WillBeJumped method not in the advanced player class? There is nothing to stop a BasicPlayer object calling it also! Nice screenshots and again a good effort. | 85  75 |
| 1198505 | * Classes   Functionality OK. It seems to play the interactive game. It would have been good for you to code an algorithm to automatically determine a winning hand. Nice implementation using collections. No card class and operating on strings is more awkward. A good effort overall   * Inheritance   Very good functionality and efficiently coded algorithms. Correct use of polymorphism. Good work. | 75  80 |
| 1132398 1052974 | * Classes   A very bloated design. I am not sure what some of the classes you have created do! Your DeckofCards class is very inefficient. Why does it store multiple arrays of cards? No need for a separate Win class. Why is the main method in DeckOfCards? Your program runs OK and exhibits some functionality although there are some errors in the output you display (the same card twice) and you have not demonstrated the win condition. Rather messy code.   * Inheritance   No functionality as your game immediately crashes. There is no array out ofbound check which explains the problem. Some code written but no implementation of polymorphism. A poor effort. | 55  37 |
| 1095307 | * Classes   Excellent work. Full and extended functionality including autonomous play. Only criticism is that the hand should be sorted into suits when displaying.   * Inheritance   Full functionality. Nicely coded and correct use of polymorphism. | 97  90 |
| 1037736 1037571 | * Classes   Partial functionality only. The game hangs after a couple of hands are played. Bloated design of the DeckOfCards. Only1 array of cards required. You obviously didn’t attend lectures as I outlined a clear design of the application which you have not followed. At least your shuffle, deal and cut methods seem to work.   * Inheritance   No functionality. Again it seems like you didn’t attend the lecture where I outlined the design. You have not overridden the movePiece method of Player and haven’t implemented polymorphism. A poor effort. | 45  25 |
| 0915954 0900107 | * Classes   No functionality as the main program did not compile. Some of your design is sensible although the Card and DeckOfCards class designs (which I went through in a lecture) are very bloated. You should have at least written a main program to demonstrate the deal, shuffle and cut method.   * Inheritance   Again no functionality as the program crashes on launch. It seems you have completely changed the graphics program code that I gave you and you have attempted to write a different program. There doesn’t seem to be any use of inheritance/polymorphism which is what the exercise was about. Again I went through the exercise in detail in lectures which you should have attended | 30  15 |
| 1011105 1079901 | * Classes   Rather limited functionality. A bloated design for the DeckOfCards. Not necessary to store multiple decks or players hands. You have at least implemented cut and shuffle and deal methods but not fully tested them. You clearly have not attended lectures where I went through the design required for the main classes of this application. No example output submitted. A very poor effort  Inheritance  No functionality. Your moveBlack and moveWhite methods in Draughtboard contain no out of bounds checks which probably explains why the program crashes before a move is made. Not much code in the Player classes. Again you should have attended the lectures where I went through the design and implementation of this exercise. | 25  20 |
| 1039038 1031491 | * Classes   Good class design for the Card, deck and game classes. I couldn’t verify functionality as the instructions on how to use the commands were not clear. You were expected to submit example program output. Coding in the main game class is poorly laid out.   * Inheritance   Full functionality. Correct use of polymorphism. The moveBlack and moveWhite methods in the Draughtboard class seems overly complex. They should not decide on which move to make. They simply update the board array in response to the movePiece methods in the player classes. No example screendumps submitted. | 55  65 |
| 1097820 1014415 | * Classes   Some functionality. You are storing cards as strings which is inefficient. I went through this in lectures. No attempt to implement the game but you have demonstrated some simple manipulations of the card deck. You have not really implemented an object oriented application as you only have a single class and a main program so no object interactions.   * Inheritance   No functionality as the program doesn’t compile. You have submitted some code in the Draughtboard class which looks OK. | 50  20 |
| 0771658 | * Classes   Good functionality. Card class OK but DeckOfCards over complex and inefficient. No need for multiple decks. Good interaction between the objects and sensible output displayed on the console.No output submitted so don’t know if the game runs successfully to completion   * Inheritance   No functionality as the program doesn’t compile. You have submitted minimal code. | 65  10 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |