|  |  |  |
| --- | --- | --- |
| **Student IDs** | Comments | Mark |
| 0795807 1038455 | * Classes   Some functionality. Deal, cut, shuffle work OK. No implementation of the full game. Your code has a sensible class structure. Inefficient implementation of a playing card. Using strings is not necessary. You could have included a simple method to display the card in a user friendly way especially as they are represented as strings! Arguably the Deal method should be part of the Player class. Overall a reasonable effort for a first program.   * Inheritance   Full functionality with an implementation of AdvancedPlayer. It seems to work OK. Your code layout is poor and not at all readable with few comments. Your BasicPlayer class should have extended the Player class but you included it within Player instead. Not a big issue. You correctly applied inheritance to AdvancedPlayer. Overall well done. A good effort. | 17/30  21/30 |
| 1018750 088406 | * Classes   Good functionality. The basic methods for the Deck class work fine. I couldn’t find program output displayed for the Game. Your code compiled and ran OK but the interface is awkward and I didn’t check it running to completion. You have used some quite sophisticated code and container structures in your Game which weren’t probably necessary. Overcomplex class structure. No need for separate suit and value classes. Overall though well done and a nice piece of work for your first attempt.   * Inheritance   Full functionality with implementation of AdvancedPlayer. You have applied polymorphism correctly. Your code is complex and not sufficiently commented. Also your submitted output is inadequate with a single screendump which doesn’t show me very much. But you have clearly understood the requirements of the lab. | 23/30  18/30 |
| 1052959 958465 | * Classes   I could find no working main program so nothing to demonstrated. You have submitted minimal code and no example output. What you did submit does compile OK.   * Inheritance   Some functionality here but the white pieces don’t move correctly. You have applied polymorphism correctly in your program. The code you have submitted is quite well laid out but overcomplex and almost totally uncommented! I can find no example output such as screendumps demonstrating your working program. | 5/30  15/30 |
| 996025 1002938 | * Classes   Full functionality. A good effort at getting the game working. Nicely laid out classes with sensible delegation of methods to appropriate classes. You make life difficult for yourself and it’s also inefficient to represent the cards as strings. Well laid out code but there could be more commenting. Your submitted example program output is clear.   * Inheritance   Only the basic player implemented but it seems to exhibit correct functionality. Correct application of polymorphism. Your code is fine and well laid out. Good program output submitted with some reasonably clear screendumps. Nice effort overall. | 24/30  20/30 |
| 1061734 1028285 | * Classes   Almost zero code submitted. You submitted a class file in place of the DeckofCards source file so I couldn’t take a look at the code. It does have some functionality though. No example program output submitted either.   * Inheritance   Nothing submitted | 10/30  0/30 |
| 997142 1023688 | * Classes   Full functionality for the Deck class and the game. Nicely constructed classes with sensible delegation of methods. Your code is readable and well layed out with reasonable use of comments. Efficient implementation of the Card class. In your example output, it would have been nice to have seen a game played to completion.   * Inheritance   Nothing submitted | 24/30  0 |
| 1033082 | * Classes   Your code doesn’t compile and is untidy and not especially readable. No example output submitted. Some attempt as a class design but you have obviously found the exercise difficult.   * Inheritance   Again your program doesn’t compile. I can’t understand why you have made life difficult for yourself by rewriting my DraughtBoard class. I suspect you haven’t attended lectures as I went through how to tackle both labs in the lectures. Some code submitted but very untidy and unreadable. I’m not sure what the png file you included is supposed to represent. | 7/30  7/30 |
| 1057007  1044925 | * Classes   Full functionality. Seems your game runs OK but you have not demonstrated any basic deck of cards methods such as cut and shuffle. Your code is compact and you have implemented rather strange relationships between classes. Why is DeckOfCards extended from Player?? Inefficient implementation of the Card class. No need to use strings. I coudn’t find any program output submitted. A reasonable effort at the first assignment.   * Swing and GUIs   I couldn’t get your image game to run (although it compiles). It simply displays a blank form. I think you may have to come and see me to explainwhat it’s supposed to do and I may revise the mark. You have not followed the specification as you were supposed to make use of supplied code to load an image. I suspect you have used a drag and drop IDE to implement this assignment. You should have also submitted some example program output allowing me to better judge what you have achieved. You seem to have implemented a lot of classes for a simple exercise. Most of the code should have been added to the GUI code provided. | 17  17 |
| 1006041 861692 | * Classes   Very good piece of work. Full functionality and an autonomous game. I think your class structure is not quite accurate. A lot of code in the DeckOfCards class should be in Player. The selection of which card to play for example. However, you have got the program working which is a good achievement. Also the example output is clear.   * Inheritance   Full functionality. Well written code and example screendumps shown. You should have created a BasicPlayer class as an extension from Player rather than adding code to the Player class which is simply meant to be an abstract base class. Apart from that, you have correctly applied polymorphism. The code is fairly well layed out and commented. I couldn’t see a large difference in performance between the basic and advanced players (when I ran the program, the basic player won!). | 26  23 |
| 813396  971302 | * Classes   Plagiarism   * Inheritance   Program doesn’t compile and no example output submitted. You have applied polymorphism and inheritance correctly and submitted a fair amount of code. | 0  9 |
| 1032122 1037736 | * Classes   Doesn’t compile. A lot of messy code presented. No use of private instance variables. Why have you got a class called ‘main’?   * Inheritance   You haven’t applied polymorphism as was the intention of the exercise. There are no inheritance relationships in your program! You have achieved full functionality so you will get credit for that. No example output has been submitted. Your program is well laid out but lacks adequate commenting. Also you have put a lot of code in the Draughtboard class which should be in the player class. It’s obvious that you haven’t attended lectures as all of this was explained. | 5  13 |
| 1011100 1010842 | * Classes   A fairly basic implementation with some functionality. Using strings to represent a card is inefficient. You created a Player class (which looks OK) but it’s not used. You could have passed a reference to a Player object in your Deal method to update the player’s hand. This would have enhanced your functionality. However, not a bad first attempt. Your example program output is fine.   * Inheritance   You haven’t applied polymorphism as was the intention of the exercise. There are no inheritance relationships in your program! Did you have attend lectures as all of this was explained. You have achieved full functionality so you will get credit for that. Your example program output is very clear. Your code layout is complex with not a lot of commenting. | 17  16 |
| 1015448 1067142 | * Classes   A fairly basic implementation with some functionality. Using strings to represent a card is inefficient. Your submitted example output is OK. It’s not especially sensible to put the hand of the 4 players in the Deck class. Why not create a separate Player class for this. This would have significantly enhanced your program and been a good step towards an OOP solution. Overall not a bad first attempt.   * Inheritance   Partial functionality. You haven’t implemented the advanced player and your basic player moves seem a bit strange. But it’s almost working. You have applied polymorphism and inheritance correctly. Your example program output is very clear. Your Player code is neat and well commented although your moveBlack and moveWhite methods in the DraughtBoard class are incomplete which could explain the incorrect functionality of the basic player. Not a bad effort overall though. | 17  18 |
| 1036948 87168 | * Classes   Full functionality it seems (although I didn’t play the game to completion). Your submitted example program output shows a couple of screendumps. You have essentially written a C program. No use of OOP design which is evidenced by the large number of static methods! A lot of complex code submitted but not much commenting (except to comment things out!). You haven’t demonstrated the deal or cut methods in the Deck class. I’m not sure why you have to use 2 dimensional arrays to represent a Deck! You still need to make the transition from 1E to 2E judging by this program!   * Inheritance   Full functionality and nice screenshots submitted. Your advanced player implementation looks like a good extension of the basic player algorithm and comprehensively won against a basic player the 2 times I tried it! Again your code is complex but more commenting has been used than your previous assignment. A much better effort this time and you have correctly applied polymorphism and inheritance. | 15  23  -5 for 1 day late |
| 1037820 1049430 | * Classes   Some functionality. Your game class doesn’t compile. But you have demonstrated some basic classes including a cut method. Your code is rather untidy and not much commenting. An inefficient representation of a card as a pair of strings. Your example program output (a screen dump) is not easy to read and not very informative.   * Inheritance   Only partial functionality as your game crashes with an array out of bound exception. Your code in the DraughtBoard class is not complete which is why your program crashes. Your BasicPlayer code is well laid out and you have the correct implementation of inheritance and polymorphism. Only one simple screendump has been included as example output. | 15  14 |
| 1037568 1019309 | * Classes   Your program compiles but crashes when I try to run it. No example output provided. The program is essentially procedural and not object oriented. You have a Card class but with no functionality. The game code is complex with little commenting and not very readable.   * Inheritance   Again a non object oriented implementation. You haven’t used inheritance or polymorphism. Your program seems to run OK to completion. Most of the code is in the Player class. No example output presented. You have submitted a lot of working code for which I have credited you but the code is not commented adequately. It would appear that you haven’t attended lectures where I explained the outline design for both labs! | 10  17 |
| 1027106 1052974 | * Classes   Full functionality with a lot of code presented. I haven’t checked that your game runs to completion. Your example screen output could have demonstrated this better. Good class design and correct delegation of methods to appropriate classes. Some inefficient coding in parts. The printHand method in Players for example. A good effort overall though for a first program.   * Inheritance   Partial functionality only with the pieces not moving correctly. A non object oriented implementation. You haven’t used inheritance or polymorphism. Your code is very insubstantial. This was the point of the exercise. It would appear that you haven’t attended lectures where I explained the outline design for this lab! No example output submitted. | 23  13 |
| 1067066  1023776 | * Classes   Full functionality. Your submitted output is good showing the game running to completion. An partial object oriented solution has been presented but more of the functionality about determining the correct plays should have been delegated to methods of the Player class. Your code is well commented on the whole. A very good effort for your first program.   * Inheritance   Full functionality. Your program runs OK to completion. Well laid out code and full use of commenting. You haven’t implemented the Advanced Player but nevertheless a good piece of work. You could have submitted a few more screenshots demonstrating your game running to completion. | 23  20 |
| 1020161 1008195 | * Classes   A very good effort. Your program seems to run to completion. You submitted some example output but it would have been nice to see output relating to the concluding stages of the game. Well structured classes. A lot of nice functionality in the Card class and correct delegation of functionality to the Player and Game classes. Your code is well commented although rather untidily laid out (incorrect indentation) in places. Well done overall for a first Java program.   * Inheritance   Mostly correct functionality although the game seems to continue after 1 colour has won! You haven’t used inheritance or polymorphism as was the intention of the lab exercise. Your submitted example output is OK but doesn’t really show me very much. No implementation of Advanced Player. | 23  15 |
| 1044462 1031844 | * Classes   Files not zipped! Some basic functionality demonstrated. A simple Card class has been implemented but using strings is inefficient. Your cut deal and shuffle methods seem to work OK. Some example program output submitted as a text file. You have at least included all the sensible classes necessary to play the game even though you have not managed any implementation of the game code.   * Inheritance   Your program executes OK and plays a game to completion but you haven’t implemented polymorphism or inheritance! You clearly didn’t attend lectures where this was explained. No example output submitted. Your code in the Draughboard class looks OK but the Player code is simplistic and not well laid out with lots of commented out sections. You have clearly struggled with this exercise. | 15  12 |
| 1022332 1045102 | * Classes   Your program runs OK and you have implemented the game. It’s not clear from your submitted output that a player does win as the final hand is not a winning one! Well laid out code and good commenting. Sensible use of classes but an additional class representing the game would complete the object oriented design (you have a Game class containing a main method only).  Your basic card class is OK but I can’t understand why you have an integer for the card value and a string for the suit! Overall a pretty good effort for a first program.   * Inheritance   Your program executes OK and plays a game to completion but you haven’t implemented polymorphism or inheritance! Your Player code is complex and full of commented out sections. The whole point of inheritance is to delegate the different levels of play to different classes thus avoiding ‘over coding’ a single class which you have done. I did explain this clearly in a lecture. Your submitted screen dumps were OK. | 22  15 |
| 1017701 956581 | * Classes   Basic functionality only. Your Card class represents a card object as a string which is rather inefficient. Your shuffle, cut and deal methods seem to work OK. No example program output submitted. Well done for using static string arrays to store names of suits and values. I have given you an extra couple of marks for that! (They should really be part of the Card class and used by a method to display a card). You incorporated your Player class into the deal method which is the first step towards an object oriented solution to the card game.   * Inheritance | 17 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |