

# Object Oriented Programming 2021/22

## Project self-evaluation form

Oral discussion date Fri-24/6 12:00

Group number 23

Student Number	Student Name	Percentage of participation (must sum 100)	Expected mark (0-10 points)
93168	Ricardo Matias Febra	33.3%	9
93178	Rui Silva Ferreira	33.3%	9
93160	Pedro Soares Ricardo	33.3%	9

Mark	Command/feature	Correctly implemented	Implemented with faults	Not implemented	Prof notes
<b>UML</b>					
5/20	Tool used: <u>Visual Paradigm</u> Was it done with reverse Engineering? <u>No</u>				
<b>Basic game</b>					
1/20	Bet command	X			
1/20	Credit command	X			
1/20	Deal command	X			
1/20	Hold command	X			
<b>Strategy and statistics</b>					
3/20	Perfect strategy	X			
0.5/20	Advice command	X			
0.5/20	Statistics command	X			
<b>Modes</b>					
1/20	Debug mode Reading card and command files and running commands	X			
1/20	Simulation mode Shuffling/Re-shuffling, commands with perfect strategy and statistics	X			
<b>Documentation</b>					
1/20	Examples of debug files Examples to test the game in debug mode	X			
2/20	Java doc Packages, interfaces, classes, methods, fields		X		

<b>Visualization</b>			
1.5/20	Correct in example files without errors A correct output should give a correct info in all commands/advice/statistics	Prof notes:	
0.5/20	Correct in example files <u>with</u> errors A correct output should not crash and give information to the user	Prof notes:	
<b>Discounts</b>			
-3/20	Interfaces and polymorphism used incorrectly	Give here the name of all interfaces in your project: NA Give here the name of all abstract classes in your project: NA Give here the name of all polymorphic methods in your project: get_command and reset deck methods for the debug and simulation modes	Prof notes:
-2/20	Open-closed principle used incorrectly	How many packages? 2, main and videopoker Visibility of the attributes (choose all used): - ~ # + + for the lists of ranks and suits in the Card class; # for the attributes of the game and the variation in the VideoPoker and DoubleBonus Classes; - for everything else	Prof notes:
-1/20	Object class / collections used incorrectly	Which classes override equals? none Did you use any sorting method or sorted collection from Java? yes, to sort the hand in the order_deck method from the Deck class How is the deck shuffled? We utilized the shuffle method from Java collection Did you provide your own exceptions? no	Prof notes:
-1/20	Incorrect data structures	Collections used (apart from arrays): none	Prof notes:
-1/20	Prints outside the format	Professor notes:	
-0.5/20	Problems with the executable Incorrect MANIFEST.MF and JAR w/out java sources		
-0.5/20	Files submitted outside the format Other compression than .zip and incorrect folders		
-2 <sup>n</sup> /day	Projects submitted after the established date		