**Project 2**

**Title**

**Black Jack 2**

**Course**

**CSC-17a**

**Section**

**45434**

**Author**

**Cory Chesebro**

**1 Introduction**

**Black Jack is a simple game where there is a dealer and at least one player.**

**The game is card-based meaning there can be one or more decks involved, many casinos implement the use of multiple decks to prevent card counting which is a technique used to gain an advantage by a player. The goal of the game is to reach a value of 21, which means you win by default, or to have a higher value than everyone else with out going over 21. The player starts with two cards and can “hit” or “stand”. A hit signals the dealer to give the player another card and a stand signals the dealer to stop giving the play more cards. Each numbered card is worth its face value and each faced card is worth 10 except for the ace. The ace which can be valued at either 1 or 11; which ever is in the favor of the player.**

**2 Game play**

**In my game, the player is presented with the rules and given the option to play or to not play. Subsequently after the player is asked if they would like to bet. If they would like to bet, they are then asked how much they would like to bet before receiving their cards. After they enter a valid amount they receive their first two cards and then after are given the option to hit or stand. If they choose to hit they are given another card and given the option to hit or stand again unless their value is over 21, then the player busts and the round is lost for the player. If the player does not bust, the player is given to option to hit until they signal to stand. After the player stand the dealer draws their cards and then you compare card values determining the winner.**

**3 Development summary**

**This is my fourth time programming the black jack game and arguably the hardest. The game is not complex enough to fulfill all the necessary requirements and some requirements caused me to do things in a total unnecessary manner. Regardless the program does work and includes the majority of the required concepts. Challenges I came across were getting the dealer and player access to the deck variable which resided in the Game class. The solution was friendship! No but really, the base class of the Dealer and User class had a friend ship with the Game class allowing access to the global game deck granting the Dealer and User the ability to take cards from the deck in a realistic manner. There were 52 cards in the deck and as a player took a card, the corresponding card was removed from the deck.**

**4 References**

**References included:**

**Stackoverlow.com**

**Cplusplus.com**

**Geeksforgeeks.com**

**cboard.cprogramming.com**

**en.cppreference.com**

**ibm.com**

**5 Requirements**

**Cross Reference for Project 2**

**You are to fill-in with where located in code**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Chapter** | **Section** | **Topic** | **Where Line #''s** | **Pts** | **Notes** |
| 13 |  | Classes | Dealer.cpp, Game.cpp,  Player.cpp,  User.cpp |  |  |
|  | 1 to 3 | Instance of a Class | Main.cpp line 17 | 4 |  |
|  | 4 | Private Data Members | Game.h line 30 | 4 | Never Public |
|  | 5 | Specification vs. Implementation | Game.cpp, Game.h | 4 | .h vs. .cpp files Always split |
|  | 6 | Inline |  | 4 |  |
|  | 7, 8, 10 | Constructors | Game.h lines 18 & 19 | 4 | Overloading |
|  | 9 | Destructors | Game.h line 21 | 4 |  |
|  | 12 | Arrays of Objects |  | 4 |  |
|  | 16 | UML | Doxygen html files | 4 |  |
|  |  |  |  |  |  |
| 14 |  | More about Classes |  |  |  |
|  | 1 | Static |  | 5 |  |
|  | 2 | Friends | Game.cpp line 28 | 2 |  |
|  | 4 | Copy Constructors | Game.cpp line 19 | 5 |  |
|  | 5 | Operator Overloading | User.cpp line 48 | 8 | Overload 3 operators |
|  | 7 | Aggregation | Game.h line 23 | 6 |  |
|  |  |  |  |  |  |
| 15 |  | Inheritance | Dealer.h line 15 |  |  |
|  | 1 | Protected members | Player.h line 30 | 6 |  |
|  | 2 to 5 | Base Class to Derived | Dealer.h line 15 | 6 |  |
|  | 6 | Polymorphic associations |  | 6 |  |
|  | 7 | Abstract Classes | Player.h | 6 |  |
|  |  |  |  |  |  |
| 16 |  | Advanced Classes |  |  |  |
|  | 1 | Exceptions |  | 6 |  |
|  | 2 to 4 | Templates | Dealer.h line 14 | 6 |  |
|  | 5 | STL |  | 6 |  |
|  |  |  |  |  |  |
|  |  | Sum |  | 100 |  |

Page 1