Project 1

Black Jack

(21)

Victor Alcaraz

CSC – 5

46090

07/21/2015

1 Introduction

The game of Black Jack, also known as 21, is a casino banking game. With 2-6 players, everyone playing goes against the dealer and not each other. Black Jack is played with a minimum of 1 deck of cards (52 cards).

The name of the game often refers to a player getting an ace of spades and a black jack, either clubs or spades, in his or her first two cards. In more recent times, the title of black jack was to refer to a total of 21, regardless of suit, on the players first two cards. This must occur without the dealer also getting the same total in his or her first two cards.

The game is dealt and played clockwise, which is starting with the person most left of the dealer. The objective of the game is to then get as close to or land right on 21 without going over. Everything above 21 is called a bust and therefore the player immediately loses and forfeits his or her bet.

Every number card is worth it's face value, but special cards such as A, K, Q, and J get a more special treatment. More information on playing the game will be explained in the **Rules and Regulations** section of the manual.

2 Summary of Work

This program contains roughly 300 lines of code, 15 variables , 2 functions, and 1 random number seed. Over the summer session of 2015, this project took about 1 week to finish. Choosing this game for my project was certain as I played the game of Black Jack quite a bit and understood the rules with ease.

This iteration of the game will not initialize betting. This means that “double downs,” “splitting,” or “insurance” will not be implemented. Also, this iteration will only have 1 player go against the dealer. No other players will be present. This program only runs 1 deck of cards with the deck reset every hand dealt. All cards are randomized. This means a random face value matched with a random suit.

At the beginning of the game, the player decides how many games he or she would like to play. From there, the dealer card is showed first, while the player card gets shown second. The hand is then played and repeated the number of times the player chose at the beginning. One small note: if the game is tied(“push”), then the game number gets repeated. In which case, the total number of games can exceed the initial number inputted by the player.

3 Rules and Regulations

As stated before, the cards are dealt in a clockwise manner, starting with the left most person from the dealer. The cards are dealt one at a time face up to each player. Two cards to initialize the game. The dealer gets one card face up and the second card face down. After all cards are dealt, the hands are played one player at a time, again, in a clockwise manner.

3.1 Win Condition

The goal to win the game is to get a higher sum of cards than the dealer without going over the sum of 21. This is achieved with commands: Hit or Stand. Hit meaning add one more card while Stand meaning stay with the cards the player currently has.

3.2 Card Values

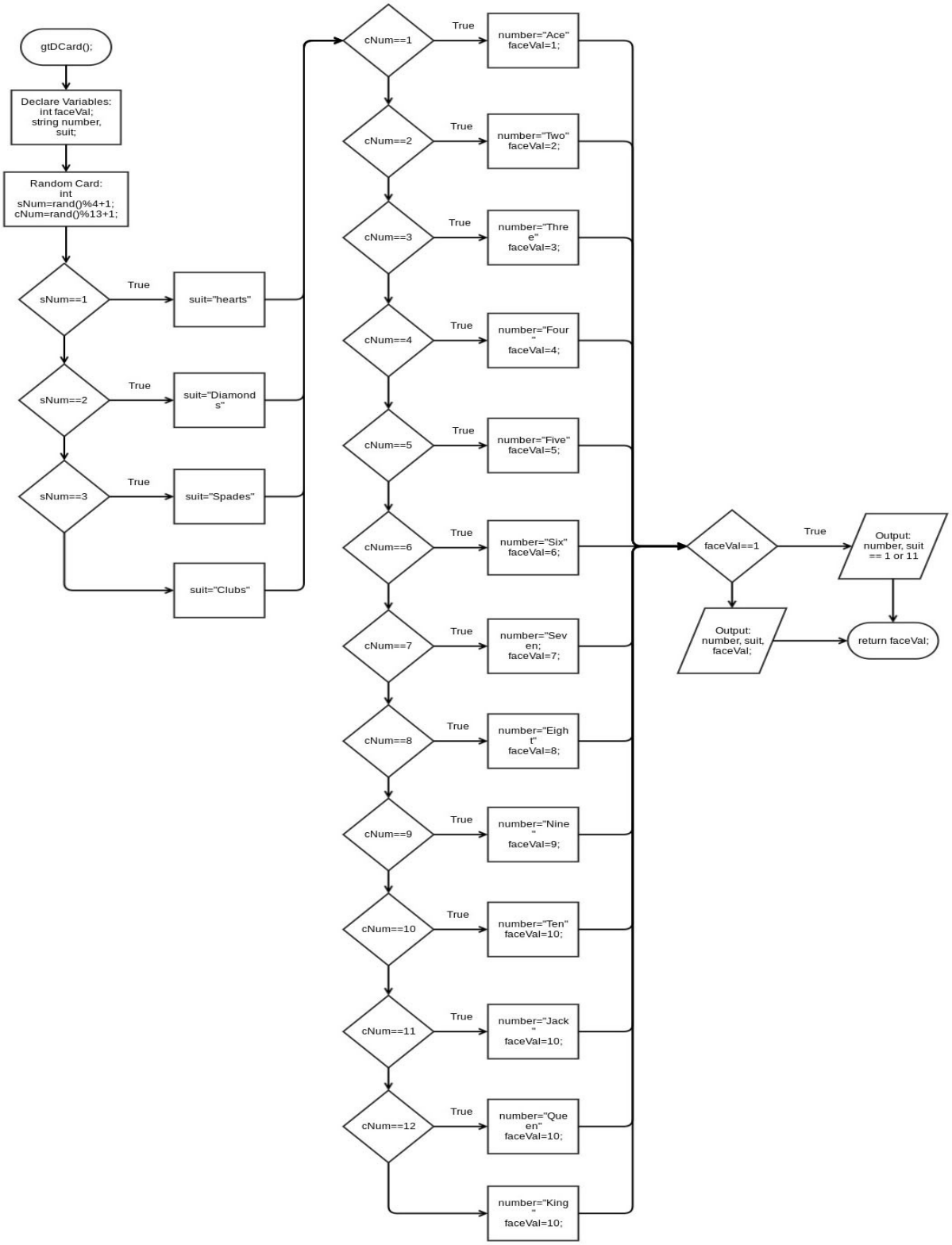
Each card in the deck has a value accustomed to it. Number cards, 2-10, have face value. The “Ace” has a special value of 1 or 11. And the “King,” “Queen,” and “Jack” are all given the value of 10.

3.3 Playing Each Hand

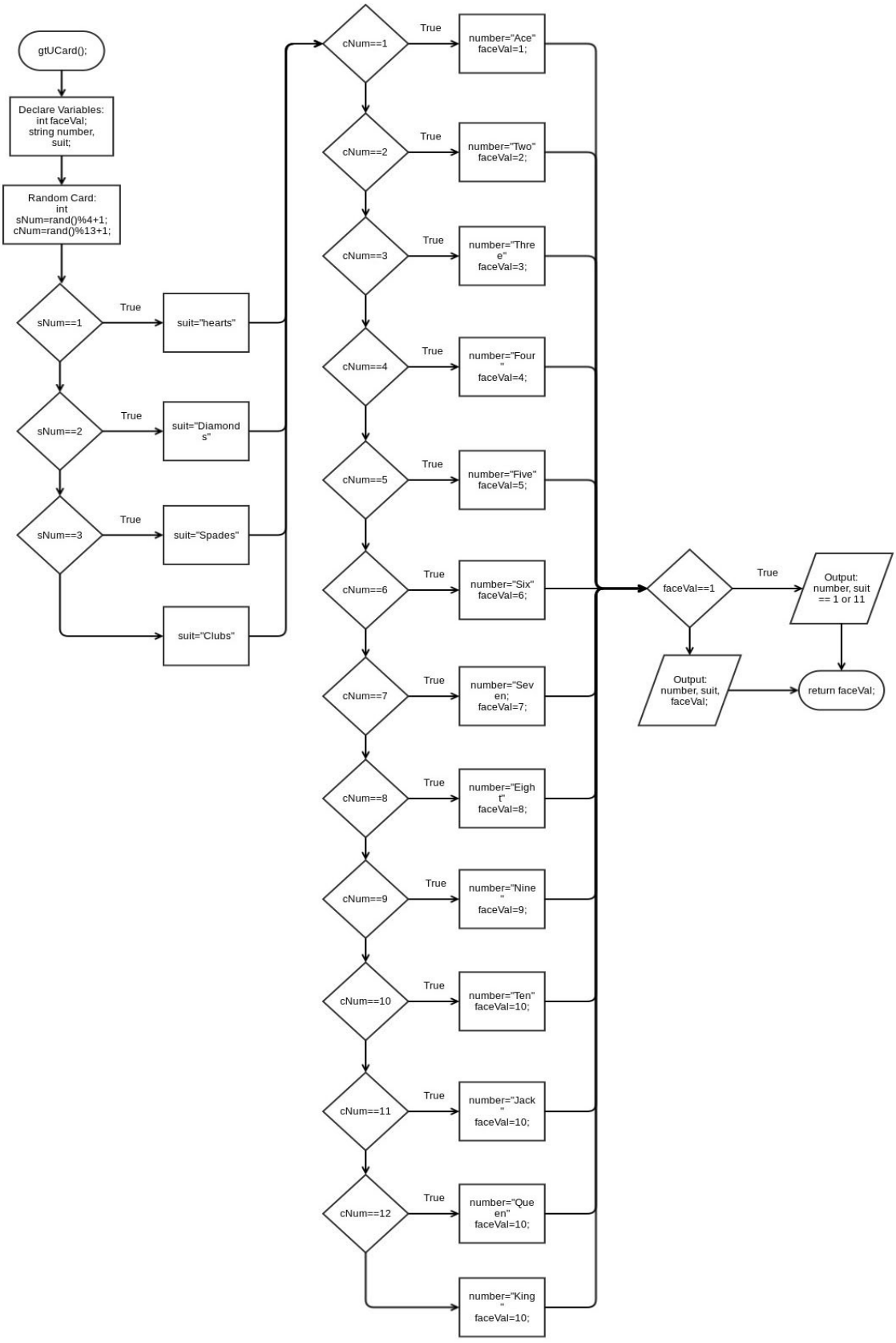
Each hand is initialize with two cards faced up. As stated before, the goal is to get a higher sum than the dealer without going over the sum of 21. If the two initialized cards are an “Ace” and a card valued at 10, then the player gets a “black jack” and wins the hand automatically only if the dealer doesn't get the same initialized sum. If the player's sum does not initially occur as 21, then the player can decide to Hit (take another card) or Stand (stay with the current sum of cards). The player can decide when to Hit and Stand on his or her own. Unlike the player's decisions, the dealer must keep Hitting until the sum of the dealer's cards is 17 or greater. Once the sum for the dealer is 17 or greater, then the dealer must automatically Stand. Once the hands of the players and dealers is complete, the sums of the cards get compared. If the dealer's sum is greater than a player's sum, then the dealers beats that player. Visa versa for a player victory. A “push” will occur if the sum of the dealer's cards matches the sum of the player's cards. If a “push” occurs, then the game is a draw.

4 Flow Chart

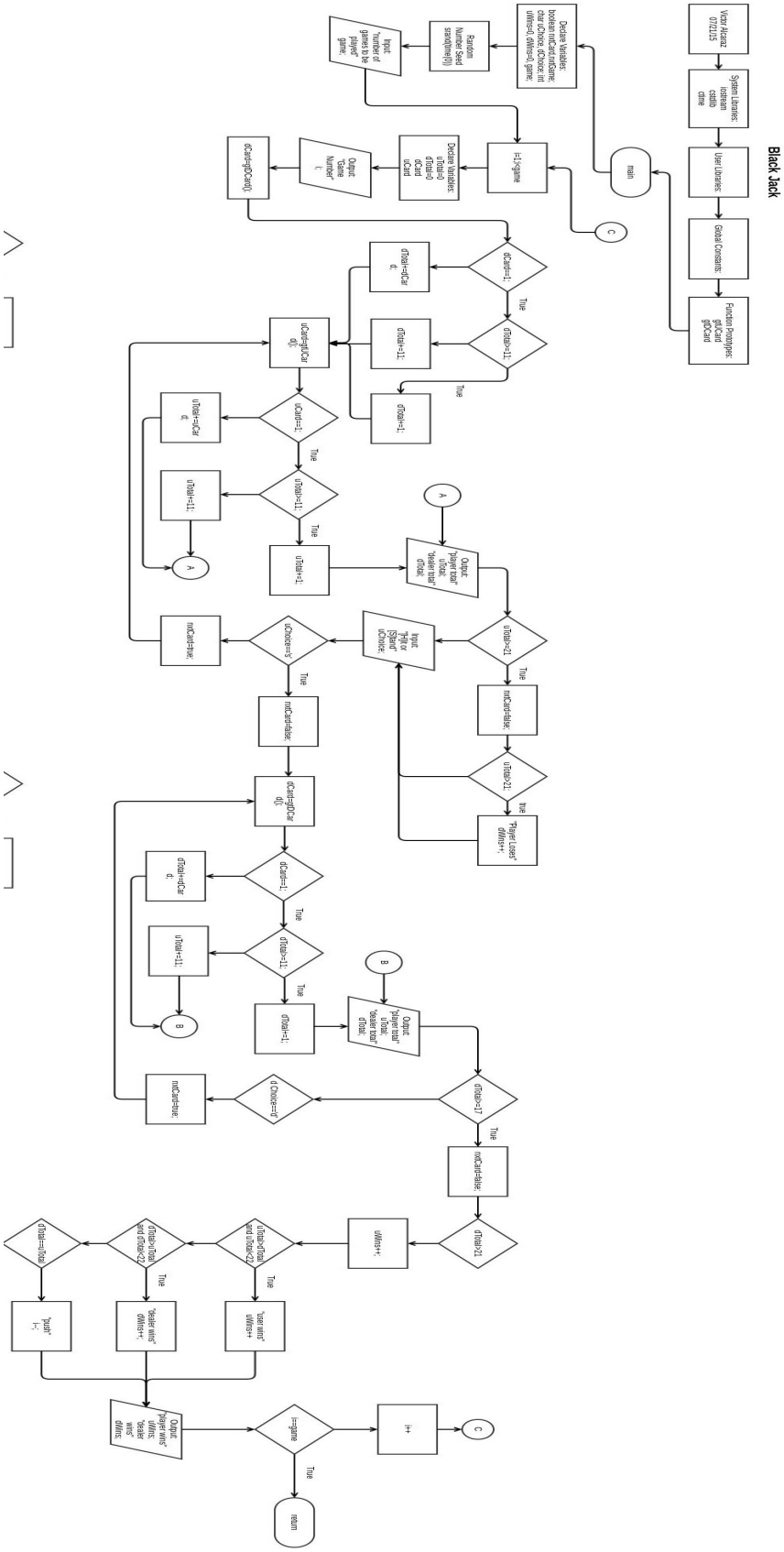
Dealer Card Function



Player Card Function



Main Function



5 Code

```
/*
 * File:   main.cpp
 * Author: Victor Alcaraz
 * Created on July 20, 2015, 6:49 PM
 * Purpose: Play Black Jack
 */

//System Libraries
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

//User Libraries

//Global Constants

//Function Prototypes
int gtUCard();
int gtDCard();

//Execution Begins Here!
int main(int argc, char** argv) {
    //Declare Variables
    bool nxtCard,  //next card boolean
         nxtGame;  //next game boolean
    char uChoice,  //user choice
         dChoice;  //dealer hits
    int  uWins=0,   //Player wins total
         dWins=0,   //Dealer wins total
         game;      //number of games

    //set random number seed
    srand(time(0));

    //user input number of games
    cout<<"How many games do you want to play?"<<endl;
    cin>>game;
    cout<<endl;

    //complete games play again?
    for(int i=1;i<=game;i++){
        int uTotal=0,  //card totals for user
            dTotal=0,  //card totals for dealer
            dCard,     //dealer card
            uCard;     //user card

        //number game
        cout<<endl<<"Game Number "<<i<<endl;

        //dealer card
        cout<<endl;
        dCard=gtDCard();
        if(dCard==1){
            if(dTotal>=11){
```

```
        dTotal+=1;
    }else{
        dTotal+=11;
    }
}else{
        dTotal+=dCard;
}

//loop to get card totals
do{
    uCard=gtUCard();
    if(uCard==1){
        if(uTotal>=11){
            uTotal+=1;
        }else{
            uTotal+=11;
        }
    }else{
        uTotal+=uCard;
    }

    cout<<"Your Total   = "<<uTotal<<endl;
    cout<<"Dealer Total = "<<dTotal<<endl<<endl;

    if(uTotal>=21){
        nxtCard=false;
        if(uTotal>21){
            cout<<"You busted over 21. You lose."<<endl<<endl;
            dWins++;
        }
    }else{

        //validation and hit or stand call for player
        do{
            cout<<"Hit [H] or Stand [S]"<<endl;
            cin>>uChoice;
            cout<<endl;
            if(uChoice!='s' && uChoice!='S' && uChoice!='H' && uChoice!='h'){
            cout<<"You did not put an 'H' or an 'S'"<<endl;
            }
        }while(uChoice!='s' && uChoice!='S' && uChoice!='H' && uChoice!='h');

        if(uChoice=='S' || uChoice=='s'){
            nxtCard=false;
        }else{
            nxtCard=true;
        }
    }
}while(nxtCard==true);

//dealer totals
do{
    //validation if user busted, skip dealer
    if(uTotal>21){
        nxtCard=false;
    }else{
        //dealer card totals
        dCard=gtDCard();
        if(dCard==1){
```

```
                if(dTotal>=11){
                    dTotal+=1;
                }else{
                    dTotal+=11;
                }
            }else{
                dTotal+=dCard;
            }
            cout<<"Your Total   = "<<uTotal<<endl;
            cout<<"Dealer Total = "<<dTotal<<endl<<endl;
            //dealer stays on 17-21
            if(dTotal>=17){
               nxtCard=false;
               //dealer busts over 21
               if(dTotal>21){
                  cout<<"The Dealer has busted! You win!"<<endl;
                  uWins++;
               }
            }else{
               //validation to continue
               do{
                  cout<<"Press [D] to continue"<<endl;
                  cin>>dChoice;
                  if(dChoice!='D' && dChoice!='d'){
                     cout<<"You did not hit [D]"<<endl<<endl;
                  }
               }while(dChoice!='D' && dChoice!='d');
               nxtCard=true;
            }
         }
      }while(nxtCard==true);
      //calculating win, loss, or push
      if(uTotal>dTotal && uTotal<22){
         cout<<"You beat the dealer!"<<endl<<endl;
         uWins++;
      }else if(dTotal>uTotal && dTotal<22){
         cout<<"Dealer wins!"<<endl<<endl;
         dWins++;
      }else if(dTotal==uTotal){
         cout<<"Push"<<endl<<endl;
         i--;
      }

      //display wins and losses
      cout<<"Your wins   = "<<uWins<<endl;
      cout<<"Dealer wins = "<<dWins<<endl<<endl;

      if(i==game){
         cout<<"Thanks for playing!"<<endl;
      }
      cout<<"Press any letter to continue"<<endl;
      cin>>uChoice;
   }

   //Exit Stage Right!
   return 0;
}

/**************************************************
```

```
 *                get Dealer card                *
 ************************************************
 * Purpose: To get cards from random numbers for
 *          the dealer
 * Output:
 *      random card(random suit and random
 *                  face care/number)
 ************************************************/
int gtDCard(){
    //Declare Variables
    int faceVal;
    string number,
           suit;

    //Set random card
    int sNum=rand()%4+1;
    int cNum=rand()%13+1;

    if(sNum==1){
       suit = "Hearts";
    }else if(sNum==2){
       suit = "Diamonds";
    }else if(sNum==3){
       suit = "Spades";
    }else {
       suit = "Clubs";
    }

    if(cNum==1){
       number = "Ace";
       cout<<"Dealer got an ace! That is a 1 or 11"<<endl;
       faceVal=1;
    }else if(cNum==2){
       number = "Two";
       faceVal=2;
    }else if(cNum==3){
       number = "Three";
       faceVal=3;
    }else if(cNum==4){
       number = "Four";
       faceVal=4;
    }else if(cNum==5){
       number = "Five";
       faceVal=5;
    }else if(cNum==6){
       number = "Six";
       faceVal=6;
    }else if(cNum==7){
       number = "Seven";
       faceVal=7;
    }else if(cNum==8){
       number = "Eight";
       faceVal=8;
    }else if(cNum==9){
       number = "Nine";
       faceVal=9;
    }else if(cNum==10){
       number = "Ten";
       faceVal=10;
```

```
    }else if(cNum==11){
       number = "Jack";
       faceVal=10;
    }else if(cNum==12){
       number = "Queen";
       faceVal=10;
    }else {
       number = "King";
       faceVal=10;
    }
    cout<<"Dealer card:"<<endl;
    if(faceVal==1){
       cout<<number<<" of "<<suit<<" = 1 or 11"<<endl<<endl;
    }else{
       cout<<number<<" of "<<suit<<" = "<<faceVal<<endl<<endl;
    }
    return faceVal;
}

/**************************************************
 *                get User card                   *
 **************************************************
 * Purpose: To get cards from random numbers for
 *          the user
 * Output:
 *      random card(random suit and random
 *                  face care/number)
 **************************************************/
int gtUCard(){
    //Declare Variables
    int faceVal;
    string number,
           suit;

    //Set random card
    int sNum=rand()%4+1;
    int cNum=rand()%13+1;

    if(sNum==1){
       suit = "Hearts";
    }else if(sNum==2){
       suit = "Diamonds";
    }else if(sNum==3){
       suit = "Spades";
    }else {
       suit = "Clubs";
    }

    if(cNum==1){
       number = "Ace";
       cout<<"You got an ace! That is a 1 or 11"<<endl;
       faceVal=1;
    }else if(cNum==2){
       number = "Two";
       faceVal=2;
    }else if(cNum==3){
       number = "Three";
       faceVal=3;
    }else if(cNum==4){
```

```
        number = "Four";
        faceVal=4;
    }else if(cNum==5){
        number = "Five";
        faceVal=5;
    }else if(cNum==6){
        number = "Six";
        faceVal=6;
    }else if(cNum==7){
        number = "Seven";
        faceVal=7;
    }else if(cNum==8){
        number = "Eight";
        faceVal=8;
    }else if(cNum==9){
        number = "Nine";
        faceVal=9;
    }else if(cNum==10){
        number = "Ten";
        faceVal=10;
    }else if(cNum==11){
        number = "Jack";
        faceVal=10;
    }else if(cNum==12){
        number = "Queen";
        faceVal=10;
    }else {
        number = "King";
        faceVal=10;
    }

    cout<<"Your Card: "<<endl;
    if(faceVal==1){
        cout<<number<<" of "<<suit<<" = 1 or 11"<<endl<<endl;
    }else{
        cout<<number<<" of "<<suit<<" = "<<faceVal<<endl<<endl;
    }
    return faceVal;
}
```