// User is asked to start the game and 2 dice are randomly rolled, outputting a number from 2-12  
// Depending on dice roll results, the program will determine your fate  
// In-depth instructions below.  
  
/\* Instructions:  
\* A player rolls two six-sided die, which means he can roll a 1, 2, 3, 4, 5 or 6 on either die.  
\* After the dice come to rest they are added together and their sum determines the outcome.  
\* If the sum is 7 or 11 on the first roll, the player wins.  
\* If the sum is 2, 3, or 12 on the first roll, the player loses (this is called “craps”).  
\* If the player rolls 4, 5, 6, 8, 9, or 10 on the first throw, then that becomes the player’s “point”.  
\* To win, the player must “make their point”, that means that they must roll the sum they got on that first throw, ...  
\* ... so they keep rolling the dice. The player loses by rolling a 7 before making the point.  
\*/  
  
#include <iostream>  
#include <cmath>  
#include <fstream>  
using namespace std;  
  
void Instructions(double& games); // Function declaration for instructions function  
void DieStart(double& die\_1, double& die\_2, double& die\_sum); // Function declaration of die randomizer  
void DieSecond(double& die\_3, double& die\_4, double& die\_sum2); // Function declaration of die randomizer 2  
  
int main() {  
 ofstream outfile; // Directs output into separate file  
 ofstream fout("WinLossRecord.txt"); // Allows fout commands to display things in .txt file  
 outfile.open("WinLossRecord.txt"); // Opens .txt file  
  
 double dieone, dietwo, diesum; // Creates variables for the rolling of the first dies and their sum  
 double diethree, diefour, diesum2; // Creates variables for the following of future dies and their sum  
 double GAMES; // Creates a variable for pass by reference of the amount of games played record in instruction funct.  
 int bettingamount, totalbalance; // Creates variables for betting feature of program  
 int seed; // Creates a variable for declaration of seed for different results each time program is ran  
  
  
 Instructions(GAMES); // Inputs text from instructions function and passes by GAMES played  
 cout << "How much money have you brought into the casino with you? " << endl; // Asks user for total balance  
 fout << "How much money have you brought into the casino with you? " << endl;  
 cin >> totalbalance; // Records initial total balance  
 cout << "For every dollar bet, you have the chance to win twice the amount you bet." << endl; // Parameters  
 fout << "For every dollar bet, you have the chance to win twice the amount you bet." << endl;  
 cout << "If you lose, you will lose three the amount of your bet." << endl; // Parameters  
 fout << "If you lose, you will lose three the amount of your bet." << endl;  
 cout << "How much money would you like to bet? " << endl; // Asks user for amount out of total to bed  
 fout << "How much money would you like to bet? " << endl;  
 cin >> bettingamount; // Records value for amount user wants to bet  
 cout << "" << endl;  
  
  
 cout << "Please enter a random number seed: " << endl; // Asks user for random number seed  
 fout << "Please enter a random number seed: " << endl;  
 cin >> seed; // Records seed number inputted  
 srand(seed); // Inputs seed number into srand to help with the randomizing of rand()  
 cout << "" << endl; // For spacing  
  
  
 for (int x = 0; x < GAMES; x++) { // Counts games played and stops when x = GAMES inputted by user  
 DieStart(dieone, dietwo, diesum); // Passes by reference the randomized dice values  
 cout << "Your first roll is a " << dieone << "." << endl; // Displays first roll to user  
 fout << "Your first roll is a " << dieone << "." << endl;  
 cout << "Your second roll is a " << dietwo << "." << endl; // Displays second roll to user  
 fout << "Your second roll is a " << dietwo << "." << endl;  
 cout << "Player rolled: " << dieone << " + " << dietwo << " = " << diesum << endl; // Sums rolls and displays  
 fout << "Player rolled: " << dieone << " + " << dietwo << " = " << diesum << endl;  
  
 if (diesum == 7 || diesum == 11) { // Conditions for winning during the first round  
 cout << "YOU HAVE WON!! CONGRATS!! " << endl; // Displays to user that he/she has won  
 fout << "YOU HAVE WON!! CONGRATS!! " << endl;  
 totalbalance = totalbalance + bettingamount \* 2; // Calculates earnings  
 cout << "Your current balance is: " << totalbalance << endl; // Displays new balance of user  
 fout << "Your current balance is: " << totalbalance << endl;  
 cout << "\n" << endl;  
 fout << "\n" << endl;  
 }  
 if (diesum == 2 || diesum == 3 || diesum == 12) { // Condition for losing during the first round  
 cout << "Better luck next time. " << endl; // Display of when user loses during the first round  
 fout << "Better luck next time. " << endl;  
 totalbalance = totalbalance - bettingamount \* 3; // Calculates losses  
 cout << "Your current balance is: " << totalbalance << endl; // Displays new balance to user  
 fout << "Your current balance is: " << totalbalance << endl;  
 cout << "\n" << endl;  
 fout << "\n" << endl;  
 if (totalbalance < 0) { // Condition for if user runs out of money in his/her total balance  
 cout << "You are out of money! YOU'RE BUSTED!" << endl; // Displays to user that he/she is bankrupt  
 fout << "You are out of money! YOU'RE BUSTED!" << endl;  
 return 0; // Ends program when user hits -$  
 }  
 }  
 if (diesum == 4 || diesum == 5 || diesum == 6 || diesum == 8 || diesum == 9 || diesum == 10) {  
 // Condition for when user enters round two due to no win/loss condition  
 cout << "Round two incoming" << endl; // Displays to user that round two is incoming  
 fout << "Round two incoming" << endl;  
  
 do { // Complimentery to while loop  
 DieSecond(diethree, diefour, diesum2); //Imports second dice function with PBR  
 cout << "Roll 1: " << diethree << endl; // Displays result of the first roll  
 fout << "Roll 1: " << diethree << endl;  
 cout << "Roll 2: " << diefour << endl; // Displays result of the second roll  
 fout << "Roll 2: " << diefour << endl;  
 cout << "Player rolled: " << diethree << " + " << diefour << " = " << diesum2 << endl; // Displays sum  
 fout << "Player rolled: " << diethree << " + " << diefour << " = " << diesum2 << endl;  
 } while (diesum2 != diesum && diesum2 != 7); // Condition for when loop is entered to avoid infinite loop  
 if (diesum2 == diesum) { // When old die sum equals new die sum, user wins  
 cout << "YOU WON!!! " << endl; // Displays to user that he/she has won  
 fout << "YOU WON!!! " << endl;  
 totalbalance = totalbalance + bettingamount \* 2; // Calculates and adds to balance  
 cout << "Your current balance is: " << totalbalance << endl;  
 fout << "Your current balance is: " << totalbalance << endl;  
 cout << "\n" << endl;  
 fout << "\n" << endl;  
 } else if (diesum2 == 7) { // Condition for when user losses in the second round  
 cout << "Better luck next time. " << endl; // Tells user that they lost  
 fout << "Better luck next time. " << endl;  
 totalbalance = totalbalance - bettingamount \* 3; // Calculates and subtracts from the balance  
 cout << "Your current balance is: " << totalbalance << endl;  
 fout << "Your current balance is: " << totalbalance << endl;  
 cout << "\n" << endl;  
 fout << "\n" << endl;  
 if (totalbalance < 0) {  
 cout << "You are out of money! YOU'RE BUSTED!" << endl;  
 fout << "You are out of money! YOU'RE BUSTED!" << endl;  
 return 0;  
 }  
 } else { // If user neither wins or losses again, while loop repeats  
 cout << "Roll again." << endl; // Tells user that they must roll again  
 fout << "Roll again." << endl;  
 }  
 }  
 }  
}  
  
  
  
  
void Instructions(double& games) // Function for instructions  
{  
 cout << "WELCOME TO THE CASINO!" << endl;  
 cout << "You have decided to play the game of craps." << endl;  
 cout << "To play, one must roll two dice which will have their sums added up shortly after the roll" << endl;  
 cout << "If the sum values 7 or 11 on the first roll, you win." << endl;  
 cout << "If the sum values 2, 3, or 12 on the first roll, you lose" << endl;  
 cout << "If the sum values 4, 5, 6, 7, 8, 9, or 10 on the first roll, you enter the second stage of craps" << endl;  
 cout << "At this point, you must continue to roll until you roll either your original number or a 7" << endl;  
 cout << "If you roll your number, you win. If you roll a 7, you lose." << endl;  
 cout << "That having been said, let the game begin!" << endl;  
 cout << "" <<endl;  
  
 cout << "How many games would you like to play? (Please enter a value from 5-20):" << endl;  
 // Asks user for the number of games he/she wants to play  
 cin >> games; // Stores value for the number of games  
  
  
 while (games < 5 or games > 20) { // Parameter for number of games  
 if (games < 5) { // When games is under 5, user is asked to enter a higher number of games  
 cout << "You must play atleast 5 games to reserve your spot on the table." << endl;  
 cout << "Please re-enter the number of games you would like to play" << endl;  
 cin >> games; // Records new input for games  
 }  
 if (games > 20) { // When games is over 20, user is asked to enter a lower number of games  
 cout << "You are only allowed to play a maximum of 20 games. OTHER PEOPLE WANT TO PLAY!" << endl;  
 cout << "Please re-enter the number of games you would like to play" << endl;  
 cin >> games; // Records new input for games  
 }  
 }  
 if ( 5 <= games && games <= 20) { // When correct amount of games is entered, user is allowed to continue  
 cout << "Thank you for your input" << endl; // Thanks user for input  
 cout << "You will be playing " << games << " games. " << endl; // Displays number of games  
 cout << "" << endl;  
 }  
}  
  
void DieStart(double& die\_1, double& die\_2, double& die\_sum) { // Function for random dice roll 1  
 die\_1 = rand() % 6 + 1; // Randomizes number from 1-6 for die 1  
 die\_2 = rand() % 6 + 1; // Randomizes number from 1-6 for die 2  
 die\_sum = die\_1 + die\_2; // Sums dice  
}  
  
void DieSecond(double& die\_3, double& die\_4, double& die\_sum2){ // Function for random dice roll 2  
 die\_3 = rand() % 6 + 1; // Randomizes number from 1-6 for die 1  
 die\_4 = rand() % 6 + 1; // Randomizes number from 1-6 for die 2  
 die\_sum2 = die\_3 + die\_4; // Sums dice  
}