```
// ************************************************************************
// * James Small                                                          *
// * Cis-17a                                                              *
// * 10-21-06                                                             *
// * Project 1                                                            *
// *                                                                      *
// *                                                                      *
// *                     The Game of Nim                                  *
// *                                                                      *
// * This is basically are marble game where each player, you and the     *
// * computer selects marbles from a pile.  Whoever selects the last      *
// * marble losses.  That's pretty much it.  The program will keep track *
// * of high scores while doing this.  You can view the rules once you    *
// * start the game for a better idea of the game.                        *
// ************************************************************************

#include <iostream>
#include <iomanip>
#include <string>
#include <cmath> // Used for Pow function
#include <fstream> // Used for file writing and reading
#include "Nim.h"

using namespace std;

const int lowinitial = 10; // Low value for the random number generator for the initial amount of marbles
const int highinitial = 100; // High value for the random number generator for the initial amount of marbles

const int lowpair = 0; // Low value for the random number generator for who goes first and how smart the com
puter is
const int highpair = 10; // High value for the random number generator for who goes first and how smart the
computer is
const int stringsize = 10; // Used for the size of your name
const int savesize = 10; // Used for the amount of high scores in the save file

void menu1(Nim &Game, int smart);
void menu2(Nim Game);
void menu3();
void menu4(Nim Game, int &smart, bool &random);
void menu5(Nim Game);

int main(int argv,char *argc[])
{
    int smart;
    int menuchoice;
    char name[stringsize];
    bool random = false;

    cout << "Please enter your first name: ";
    cin.getline(name, stringsize);
    cout << endl;

    while (strlen(name) == 0) // Loops until you enter something for your name
    {
        cout << "\nPlease enter a valid first name: ";
        cin.getline(name, stringsize);
        cout << endl;
    }

    Nim game(name); // Defines Nim game object and passes the players name into it

    do // Loop used for the game menu
    {
        system("cls");

        if (!random) // Used if you haven't selected how smart the pc is.  It will randomly select how smart
 it will be
        {
            game.random(smart, lowpair, highpair); // Selects random number between 0 and 10

            if (smart <= 5) // Used if random number is less than or equal to five a makes the computer stup
id
            {
                smart = 0;
            }
            else // Used if random number is greater than five and makes the computer smart.
```

```
            {
                smart = 1;
            }
        }

        game.GameMenu();
        cin >> menuchoice;

        switch (menuchoice) // Switch used for menuchoice
        {
        case 1: menu1(game, smart);
            break;
        case 2: menu2(game);
            break;
        case 3: menu3();
            break;
        case 4: menu4(game, smart, random);
            break;
        case 5: menu5(game);
            break;
        default: cout << "\nInvalid Menuchoice!!\n\n";
        };

    }while (menuchoice != 5);

return 1;
}

// ********************************************************************
// * Definition of the function: menu1                                *
// * This function takes as its arguements a reference to the Nim Game  *
// * object along with an int for how smart the computer will be.  It   *
// * runs the main loop for the entire game going back and forth between*
// * the computer and user selecting marbles.  Once a winner is         *
// * determined, it ends the loop and brings you back to the main menu. *
// * It is used if menu choice 1 is selected.                         *
// ********************************************************************

void menu1(Nim &game, int smart)
{
    int initialvalue;
    int currentvalue;
    int turn; // 0 is user, 1 is computer
    int currentnumber;
    int computercurrentnumber;
    int smartvalue = 11; // Used when computer is smart and needs to make an intelligent guess
    int temppcnum = 101; // Used when computer is smart and needs to make an intelligent guess

    game.random(initialvalue, lowinitial, highinitial); // Random initial marble value
    system("cls");
    cout << "\nAmount of marbles in the pile = " << initialvalue << endl;
    currentvalue = initialvalue; // Sets current value to initial value
    game.random(turn, lowpair, highpair); // Random who goes first.

    if (turn <= 5) // Used if random is less than or equal to five and sets turn to 0
        {
            turn = 0;
        }
        else // Used if random is more than five and sets turn to 1
        {
            turn = 1;
        }

    while (turn != 3) // Loops while turn does not equal 3
    {
        if (turn == 0) // Used if it's your turn
        {
            cout << "\nPlease enter in your guess: ";
            cin >> currentnumber;

            while (currentnumber > currentvalue / 2 || currentnumber < 1) // Loops until you enter a valid n
umber
            {
                cout << "\nYour number was not valid, try again\n";
                cout << "\nPlease enter in your guess: ";
                cin >> currentnumber;
```

```
            }

            currentvalue = currentvalue - currentnumber; // Subtracts your value from the current value

            if (!game.CheckWinner(currentvalue, turn)) // Checks if your the winner and tells you if you are
            {
                system("cls");
                cout << "\n\nYou have won the Game of Nim\n\n";

                turn = 2; // Sets turn to 2 so it will end up being 3 and exits the while loop above

                system("pause");
            }

            turn++; // Increases the turn value by 1
        }
        else if (turn == 1) // Used if it's the computers turn
        {
            if (smart == 0) // Used if computer is dumb
            {
                game.random(computercurrentnumber, 1, currentvalue / 2); // Randomly guesses a valid number
            }
            else if (smart == 1) // Used if computer is smart
            {
                computercurrentnumber = temppcnum; // Sets the computers guess to 101 for looping purposes

                // Loops below until the computers smart guess is valid

                while (computercurrentnumber > currentvalue / 2 || computercurrentnumber < 1)
                {
                    // Used if current value is one of the below smart numbers

                    if (currentvalue == 3 || currentvalue == 7 || currentvalue == 15 || currentvalue == 31 |
| currentvalue == 63)
                    {
                        game.random(computercurrentnumber, 1, currentvalue / 2); // Randomly guesses a valid
 number
                    }
                    else // Used if currentvalue is not one of the above
                    {
                        temppcnum = pow(2, smartvalue) - 1; // Sets temppcnum to a power of 2 minus 1 using
the smart value
                        computercurrentnumber = currentvalue - temppcnum; // Sets computecurrentnumber to cu
rrent value minus temppcnum
                    }

                    smartvalue--; // Decreases smart value and checks again until you have a valid smart gue
ss
                }
            }

            system("cls");

            cout << "\nComputers guess = " << computercurrentnumber;

            currentvalue = currentvalue - computercurrentnumber; // Subtracts computers number from current
value

            cout << "\n\nThe current amount of marbles left = " << currentvalue << "\n";

            if (!game.CheckWinner(currentvalue, turn)) // Checks if the computer is the winner.
            {
                system("cls");

                cout << "\n\nYou have lost the Game of Nim\n\n"; // Says you lost if there is 1 marble left
                turn = 4; // Sets turn to 4 so it will be 3 on exit of loop above

                system("pause");
            }

            turn--; // Decreases turn value by 1
        }
    }
}

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

```
// * Definition of the function: menu2                                  *
// * This function takes as its arguements the Nim Game object.  It then*
// * displays the high score menu along with their wins, losses, and    *
// * their score.                                                        *
// **********************************************************************

void menu2(Nim Game)
{
    system("cls");

    // Temp varibales used to display high score

    string tempname;
    int tempwins, templosses;
    int tempscore;

    cout << fixed << setprecision(2);
    cout << "\n    *** High Scores List ***\n\n";
    cout << left << setw(11) << "Name" << right << setw(5) << "Wins" << right << setw(8) << "Losses" << righ
t << setw(10) << "Score\n\n";

    for (int index = 0; index < savesize; index++) // Loops used to display high scores
    {

    Game.HighScores(index, tempname, tempwins, templosses, tempscore); // Gets name and scores for the index
 position of the high scores array

    cout << left << setw(11) << tempname << right << setw(5) << tempwins << right << setw(8) << templosses <
< right << setw(8) << tempscore << endl;
    }
    cout << endl;

    system("pause");
    system("cls");
}

// **********************************************************************
// * Definition of the function: menu3                                  *
// * This function displays the rules for the Game of Nim.              *
// **********************************************************************

void menu3()
{
    system("cls");
    cout << "\n*** Here are the Rules to the Game of Nim ***\n\n";
    cout << "The Game of Nim:  This is a well-known game with a number of variants.\n";
    cout << "We will consider the following variant, which has an interesting winning\n";
    cout << "strategy.  Two players, you and the computer, will atlternately take\n";
    cout << "marbles from a pile.  In each move, the player chooses how many marbles\n";
    cout << "to take from the pile.  The player must take at least 1, but at most half\n";
    cout << "of the remaining marbles left.  Then the other player takes a turn.  The\n";
    cout << "player who takes the last marble looses.\n\n";
    cout << "The program will random select a starting number between 10 and 100.  It\n";
    cout << "will randomly determine who will go first, you or the computer.  It will\n";
    cout << "also determine if the computer will be smart or stupid.  You can also\n";
    cout << "manually select how smart the computer is from the main menu.\n\n";
    cout << "The program will keep track of the number of wins vs losses and your\n";
    cout << "total score.  You earn 100 points for a win, and lose 50 for a lose.\n";
    cout << "When you end the game, if you have a high score, it will let you know\n";
    cout << "where you stand in the 10 highest scores.  The high scores will be saved\n";
    cout << "so you can try and beat your high score nex time you launch the program.\n\n";
    cout << "Good Luck!!!!!\n\n";
    system("pause");
    system("cls");
}

// **********************************************************************
// * Definition of the function: menu4                                  *
// * This function takes as its arguements a reference to the Nim Game  *
// * object, an int reference parameter for how smart the computer is,  *
// * and a bool reference for if the user selected how smart the pc is  *
// * manually.  It displays a menu allowing the user to select how smart*
// * the computer should be.  It will then stay at this level until you *
// * decide to change it.                                               *
// **********************************************************************
```

```
void menu4(Nim Game, int &smart, bool &random)
{
    int smartchoice;

    random = true;

    system("cls");

        do // Loops until user selects a valid choice below
        {
            cout << "\nHow smart do you want the computer to be?\n\n";
            cout << "1. Stupid\n";
            cout << "2. Smart\n";
            cout << "3. Random\n\n";
            cout << "Your Choice Is: ";
            cin >> smartchoice;

            switch (smartchoice) // Used to determine what the users choice was
            {
                case 1: smart = 0; // If 1, sets the computer to stupid
                    break;
                case 2: smart = 1; // If 2, sets the computer to smart
                    break;
                case 3: Game.random(smart, lowpair, highpair); // If 3, randomly sets how smart the computer
 is

                        if (smart <= 5)
                        {
                            smart = 0;
                        }
                        else
                        {
                            smart = 1;
                        }

                        random = false;

                    break;
                default: system("cls");
                        cout << "\nInvalid Choice\n";
            };

        }while (smartchoice < 1 || smartchoice > 3); // Loops while computer choice is not between 1 and 3
}

// ********************************************************************
// * Definition of the function: menu5                                *
// * This function takes as its arguements a reference for the Nim game *
// * object.  It then uses a member function to see if you have a high  *
// * score and at what position you are in in the list.  It then tells  *
// * you if you have a high score or not.                             *
// ********************************************************************

void menu5(Nim Game)
{
    int position;

    if (Game.CheckHighScore(position)) // Checks if your a winner and then displays the results
    {
        system("cls");
        cout << "\n\nYou have a high score!!\n\n";
        cout << "You have the number " << position + 1 << " score out of 10\n\n";
        cout << "Thank You for playing the Game of Nim\n\n";
    }
    else
    {
        system("cls");
        cout << "\nYou have not gotten a high score, sorry.\n\n";
        cout << "Thank You for playing the Game of Nim\n\n";
    }
}
```