~\Documents\GitHub\Projects\evan_baesler_assignment3.cpp

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <string>

using namespace std;

// Our function for the player to input their guess.
int playerTurn(int low, int high){

    int output = 0;

    // We use a true loop since return will exit this function entirely.
    while (true)
    {

        cout << endl << "Your turn! Enter your guess: ";
        cin >> output;

        // Checks for letters or other non-numeric inputs.
        if(cin.fail()){
            cout << endl << "Invalid input! Please enter a number between 10 and 99." << endl;
            cin.clear();
            cin.ignore(256,'\n');
        }

        // Checks to make sure input is within bound.
        else if(output >= high || output <= low){

            cout << endl << "Invalid input! Please enter a number between " << low << " and " << high << "." << endl;

        }

        // If our input meets our bounds, we record it as the guess.
        else if(output < high && output > low){

            return(output);

        }

    }

}

int computerTurn(int low, int high){

    int output = 0;

    // Generates a randomized number with an offset starting at (low + 1) or 11 at the beginning to a high of (high - low - 1) or (99 - 10 - 1) = 88.
    // Basically at initialization would be rand() % 88 + 11, or [0,87] + 11 for [11 , 98].
    output = rand() % (high - low - 1) + low + 1;
    cout << endl << "Computer guesses: ";
    cout << output << endl;

    return(output);

}

```

```
int main() {

    // Initializes our randomization key at the beginning of the program execution.
    srand(time(0));
    bool playing = true;
    int answer = 0;
    int guessLog = 0;
    bool closingStatement = true;
    string closeCheck;

    // Uses a true loop because the function will either be infinitely repeated, or broken by a return when program is exited.
    while(true){

        answer = rand() % 90 + 10;
        int low = 10;
        int high = 99;
        playing = true;

        cout << endl << "Welcome to the Guessing Game!" << endl;
        cout << "A secret 2-digit code has been set between 10 and 99." << endl << endl;

        // This loop represents our gameplay, and the messages above are only repeated when this loop is broken (another game is started.)
        while(playing){

            // We give the player the first turn, and then check their input. If the answer is correct in any case, there is a break and the loop ends.
            guessLog = playerTurn(low,high);

            if(guessLog == answer){

                playing = false;
                cout << endl << "You cracked the code!" << endl;
                break;

            }
            else if(guessLog < answer){

                cout << endl << "Too low!" << endl;
                low = guessLog;

            }
            else{

                cout << endl << "Too high!" << endl;
                high = guessLog;

            }


            // Follows the same check logic as above.
            guessLog = computerTurn(low,high);

            if(guessLog == answer){

                playing = false;
                cout << endl << "Computer cracked the code!" << endl;
                break;

            }
            else if(guessLog < answer){

                cout << endl << "Too low!" << endl;
```

```
            low = guessLog;

        }
        else{

            cout << endl << "Too high!" << endl;
            high = guessLog;

        }


    }

    // After a game is over, turns the boolean to true, iterates through a loop until a sufficient response is given, then sets it to false to start a new game or returns to exit the program.
    closingStatement = true;

    while(closingStatement){

        cout << endl << "Would you like to play again? (y/n): ";
        cin >> closeCheck;

        if(closeCheck == "y" || closeCheck == "Y"){

            closingStatement = false;
            playing = false;

        }
        else if (closeCheck == "n" || closeCheck == "N"){

            return(0);

        }
        else{

            cout << endl << "Error: Invalid input!" << endl;

        }


    }

}
}
```

```
import random


# Our function for the player to input their guess.
def playerTurn(low, high):
    output = 0

    # We use a True loop since return will exit this function entirely.
    while True:

        # Checks for letters or other non-numeric inputs.
        try:

            output = int(input("Your turn! Enter your guess: "))

        except ValueError:

            print("\nInvalid input! Please enter a number between 10 and 99.\n")
            continue

        # Checks to make sure input is within bound.
        if output >= high or output <= low:

            print("\nInvalid input! Please enter a number between ", low, " and ", high, ".\n")


        # If our input meets our bounds, we record it as the guess.
        elif high > output > low:

            return output


def computerTurn(low, high):
    output = 0

    # Generates a number between low + 1 and high - 1, to stay within bounds. Will only generate integers so no worries for error cases.
    output = random.randint(low + 1, high - 1)
    print("Computer guesses: ", output)

    return output


def main():
    playing = True
    answer = 0
    guessLog = 0
    closingStatement = True
    closeCheck = ""

    # Uses a True loop because the function will either be infinitely repeated, or broken by a return when program is exited.
    while True:

        # We use a random number between 11 and 98 as these values are included in our random generation,
        # and we do not want 10 or 99 as an option (unguessable.)
        answer = random.randint(11, 98)
        low = 10
        high = 99
        playing = True

        print("\nWelcome to the Guessing Game!")
        print("A secret 2-digit code has been set between 10 and 99.\n\n")

        # This loop represents our gameplay, and the messages above are only repeated when this loop is broken (another game is started.)
        while playing:

            # We give the player the first turn, and then check their input.If the answer is correct in any case, there is a
            # break and the loop ends.
            guessLog = playerTurn(low, high)

            if guessLog == answer:

                playing = False
                print("\nYou cracked the code!\n")
                break

            elif guessLog < answer:

                print("\nToo low!\n")
                low = guessLog

            else:

                print("\nToo high!\n")
                high = guessLog

            # Follows the same check logic as above.
            guessLog = computerTurn(low, high)

            if guessLog == answer:

                playing = False
                print("\nComputer cracked the code!\n")
                break

            elif guessLog < answer:

```

```
                print("\nToo low!\n")
                low = guessLog

            else:

                print("\nToo high!\n")
                high = guessLog

        # After a game is over, turns the boolean to True, iterates through a loop until a sufficient response is given,
        # then sets it to false to start a new game or returns to exit the program.
        closingStatement = True

        while closingStatement:

            closeCheck = input("Would you like to play again? (y/n): ")

            if closeCheck == "y" or closeCheck == "Y":

                closingStatement = False
                playing = False

            elif closeCheck == "n" or closeCheck == "N":

                exit()

            else:

                print("Error: Invalid input!\n")


main()
```