```cpp
/*
 * File:     GoodDiamond.cpp
 * Purpose:  Demonstrate elegant solution and good coding style
 * Project:  CSIS 2100 Diamond Homework
 * Defines:  void outputSpaces()
 *           int main();
 * Author:   Michael Van Hilst
 * History:  Created 17 October 2012
 * Copyright (c) Michael Van Hilst 2012
 */
#include <iostream>
using namespace std;

/*
 * Function: outputSpaces
 * Purpose:  Helper function that outputs spaces followed by a single character
 * Param:    int iSpaces   - the number of spaces to output
 * Param:    char cEndChar - the character to output at the end of the spaces
 * Param:    bool bEndLine - if true, include a newline at the end
 */
void outputSpaces(int iSpaces, char cEndChar, bool bEndLine) {
    for (int i = 0; i < iSpaces; i++)
        cout << " ";
    if (bEndLine)
        cout << cEndChar << endl;
    else
        cout << cEndChar;
}

int main() {
    static const int MAX_DIAMOND = 21;
    int iSize;    // size of diamond
    int iLeft;    // number of spaces to left of diamond
    int iMiddle;  // number of spaces inside diamond
    char cMarker; // the character that marks the edge of the diamond

    cout << "This program draws a diamond in your console\n";
    // Get input from use
    cout << "Enter an odd size (< " << MAX_DIAMOND << ") and a character: ";
    cin >> iSize >> cMarker;
    // Test for all forms of invalid input
    if (cin.fail() || (iSize <= 0) || (iSize > MAX_DIAMOND) || ((iSize % 2) == 0)) {
        cout << "Invalid input\n";
        return 1;
    }

    // Draw top point of diamond
    iLeft = iSize / 2;
    outputSpaces(iLeft,cMarker, true);
    // Draw top of diamond to just before widest line
    // Notice that "for" allows multiple initialization and next operations using commas
    for (iMiddle = 1, iLeft--; iLeft > 0; iLeft--, iMiddle += 2) { // while left spaces
        outputSpaces(iLeft, cMarker, false);
        outputSpaces(iMiddle, cMarker, true);
    }
    // Draw bottom of diamond from widest line to just before bottom point
    // Notice that "for" allows empty initialization or next operations
    for (; iMiddle > 0; iLeft++, iMiddle -= 2) {                  // while middle spaces
        outputSpaces(iLeft, cMarker, false);
        outputSpaces(iMiddle, cMarker, true);
    }
    // Draw bottom point of diamond
    outputSpaces(iLeft, cMarker, true);
    return 0;
}
```