```cpp
// inheritance_simple.cpp
//
#ifdef _WIN32
    #pragma warning(disable:4996)
    #include <tchar.h>
    #include <windows.h>
    #include <conio.h>
    #include <direct.h>
#elif (defined __linux__) || (defined _AIX) || (defined __APPLE__)
    #include <stdlib.h>
    #include <sys/types.h>
    #include <sys/stat.h>
    #include <unistd.h>
    typedef char _TCHAR;
    #define _tmain main
#endif

#include <stdio.h>
#include <string.h>
#include <fstream>
#include <iostream>
using namespace std;

#include "MaticaObdlznikova.h"
#include "MaticaStvorcova.h"

#define MAXLINE 255
#define MAXLEN 20

void GetNamesOfIOFiles(const char[], char[]);
//---------------------------------------------------------------------------
int _tmain(int argc, _TCHAR* argv[])
{
    TMaticaStvorcova A, B, C;
    my_class xx;
    ifstream in;
    char nameoffile[MAXLINE];
    int Exponent;

    cout << "\n  Tento program nacita z textoveho suboru stvorcovu maticu A,\n"
        << "z klavesnice prirodzene cislo k a na obrazovku a do textoveho \n"
        << "suboru vypise k-tu mocninu matice A.\n";
    GetNamesOfIOFiles("MATICA.TXT", nameoffile);
    in.open(nameoffile, ios::in);
    in >> A;
    in.close();

    cout << "\nZ textoveho suboru bola nacitana matica A:";
    cout << A;
    cout << "\nZadajte prirodzene cislo ako exponent matice A!\n  Exponent=";
    do
        cin >> Exponent;
    while (Exponent < 0);

    B = A.UmocniMaticu(Exponent);  // Overload the function pow(A, Exponent)
                            // according to our function UmocniMaticu(Exponent)!!!
    cout << endl << Exponent;
    switch (Exponent % 10) {
    case 1: cout << "-va mocnina matice A je:"; break;
    case 2: cout << "-ha mocnina matice A je:"; break;
    case 3: cout << "-tia mocnina matice A je:"; break;
    default: cout << "-ta mocnina matice A je:"; break;
    }
    cout << B;

    C = A * A;
    cout << "\nMatica C = A * A je:" << C;

    //  A += A;
    //  cout << "\nDvojnasobok matice A je:" << A;

    xx.my_getch();
    return 0;
```

```
}
//-----------------------------------------------------------------------------
void GetNamesOfIOFiles(const char name_of_input_file[], char path_to_input_file[])
{
    char current_path[MAXLINE];
    current_path[0] = '\0';

#ifdef _WIN32
    TCHAR exePath[MAXLINE];

    HMODULE hModule = GetModuleHandle(NULL);
    if (hModule != NULL) {
        if (!GetModuleFileName(hModule, exePath, MAXLINE)) {
            cout << "Nepodarila sa zistit cesta k exe-suboru.\n";
            exit(1);
        }
    }
    else {
        cout << "Module handle is NULL.\n" << endl;
        exit(1);
    }

    int iii;
    bool flag = false;
    for (iii = (int)wcslen(exePath); iii >= 0; iii--) {
        if (!flag && exePath[iii] == '\\') {
            current_path[iii + 1] = '\0';
            flag = true;
        }
        if (flag)
            current_path[iii] = (char)exePath[iii];
    }
#elif (defined __linux__) || (defined __APPLE__)
    unsigned iii;
    char line[MAXLINE];
    FILE* fp;
    if ((fp = popen("/bin/pwd", "r")) == NULL) {
        perror("popen error");
        exit(1);
    }
    if (fgets(line, MAXLINE, fp) == NULL) {
        perror("fgets error");
        exit(1);
    }
    pclose(fp);

    iii = 0;
    while (line[iii] != '\r' && line[iii] != '\n') {
        current_path[iii] = line[iii];
        iii++;
    }
    current_path[iii] = '\0';
#elif (defined _AIX)
    unsigned iii;
    char line[MAXLINE];
    FILE* fp;
    if ((fp = popen("user/bin/pwd", "r")) == NULL) {
        perror("popen error");
        exit(1);
    }
    if (fgets(line, MAXLINE, fp) == NULL) {
        perror("fgets error");
        exit(1);
    }
    pclose(fp);

    iii = 0;
    while (line[iii] != '\r' && line[iii] != '\n') {
        current_path[iii] = line[iii];
        iii++;
    }
    current_path[iii] = '\0';
#endif
```

```

    path_to_input_file[0] = '\0';
    strcat(path_to_input_file, current_path);
#if (defined __linux__) || (defined _AIX) || (defined __APPLE__)
    strcat(path_to_input_file, "/inputs/");
#elif (defined _WIN32)
    strcat(path_to_input_file, "inputs\\");
#endif
    strcat(path_to_input_file, name_of_input_file);
}
//---------------------------------------------------------------------------

```