```cpp
// W9.cpp : Defines the entry point for the console application.
//          Dziedziczenie: wywolanie konstruktorow, destruktorow

#include "stdafx.h"
#include <iostream>
using namespace std;

//declaration of class base
class base
{
      int ib;
      char obj_name[256];
public:
      int jb;
      base() {
            ib = 0; jb = 0;
            strcpy_s(obj_name, sizeof(obj_name), "......");
            cout << "konstruktor base " << obj_name << "\n";
      }                    //konstruktor bez parametrow
      base(int i, int j, char *str) {
            ib = i; jb = j;
            strcpy_s(obj_name, sizeof(obj_name), str);
          cout << "konstruktor base " << obj_name << "\n";
      }                    //konstruktor sparametryzowany
      ~base() { cout << "destruktor base " << obj_name << "\n"; }
      int get_ib() { return ib; }
      void show() { cout << "base: " << obj_name << " " << " ib = " << ib << " jb = " << jb << endl; }
};


//declaration of class derived_1
class derived_1 : public base
{
      int id;
public:
      int jd;
      derived_1(int i, int j);
      derived_1(int i, int j, int i_bas, int j_bas, char *str);
      ~derived_1() { cout << "destruktor derived_1\n"; }
      int get_id() { return id; }
      void show() { cout << "derived_1: id = " << id << " jd = " << jd << " "; base::show(); }
};
```

```
//implementation of class derived_1

derived_1::derived_1(int i, int j)
/*=============================================================================
Konstruktor klasy derived_1
=============================================================================*/
{
      //Tu bedzie wywolany konstruktor niesparametryzowany klasy base
      id = i;
      jd = j;

      cout << "konstruktor derived_1\n";
}

derived_1::derived_1(int i, int j, int i_bas, int j_bas, char *str) :
                                                              base(i_bas, j_bas, str)
/*=============================================================================
Konstruktor klasy derived_1
=============================================================================*/
{
      id = i;
      jd = j;
      cout << "konstruktor derived_1\n";
}

int _tmain(int argc, _TCHAR* argv[])
{
      base b1(1,2, "b1");
      b1.show();

      derived_1 d1(11,12);     //dla klasy bazowej bedzie wywolany konstruktor bez parametrow
      d1.show();

      derived_1 d2(11, 12, 3, 4, "d2"); //ktory konstruktor bedzie wywolany dla klasy bazowej?
      d2.show();

      system("pause");

      return 0;
}
```