/////////////////////// Exercitiul 1 //////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

#include<iostream>

using namespace std;

int x = 10;

void fun()

{

int x = 2;

{

int x = 1;

cout << ::x << endl;

}

}

int main()

{

fun();

return 0;

}

//////////////////////// Exercitiul 2 ////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

#include<iostream>

using namespace std;

class Test {

int value;

public:

Test (int v = 0) {value = v;}

int getValue() { return value; }

};

int main() {

const Test t;

cout << t.getValue();

return 0;

}

////////////////////////// Exercitiul 3 //////////////////////////////

// Ce se va afisa, adica ce ordine vor avea constructorii?

#include<iostream>

using namespace std;

class Point {

private:

int x;

int y;

public:

Point(int i = 0, int j = 0); // Normal Constructor

Point(const Point &t); // Copy Constructor

};

Point::Point(int i, int j) {

x = i;

y = j;

cout << "Normal Constructor called\n";

}

Point::Point(const Point &t) {

y = t.y;

cout << "Copy Constructor called\n";

}

int main()

{

Point \*t1, \*t2;

t1 = new Point(10, 15);

t2 = new Point(\*t1);

Point t3 = \*t1;

Point t4;

t4 = t3;

return 0;

}

///////////////////////////Exercitiul 4/////////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

#include<iostream>

using namespace std;

class base {

int arr[10];

};

class b1: public base { };

class b2: public base { };

class derived: public b1, public b2 {};

int main(void)

{

cout<<sizeof(derived);

getchar();

return 0;

}

///////////////////////// Exercitiul 5 ////////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

class Test1 {

int y;

};

class Test2 {

int x;

Test1 t1;

public:

operator Test1() { return t1; }

operator int() { return x; }

};

void fun ( int x) { };

void fun ( Test1 t ) { };

int main() {

Test2 t;

fun(t);

return 0;

}

///////////////////////// Exercitiul 6 ////////////////////////////////////

// Compileaza? Daca nu, de ce da eroare?

#include<iostream>

using namespace std;

class Test {

    int value;

public:

    Test(int v);

};

Test::Test(int v) {

    value = v;

}

int main() {

    Test t[100];

    return 0;

}

///////////////////////// Exercitiul 7 ////////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

#include<iostream>

using namespace std;

class Test

{

private:

    static int count;

public:

    static Test& fun();

};

int Test::count = 0;

Test& Test::fun()

{

    Test::count++;

    cout<<Test::count<<" ";

    return \*this;

}

int main()

{

    Test t;

    t.fun().fun().fun().fun();

    return 0;

}

///////////////////////// Exercitiul 8 ////////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

#include<iostream>

using namespace std;

class P {

public:

   void print()

   { cout <<" Inside P::"; }

};

class Q : public P {

public:

   void print()

   { cout <<" Inside Q"; }

};

class R: public Q {

};

int main(void)

{

  R r;

  r.print();

  return 0;

}

///////////////////////// Exercitiul 9 ////////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

#include <iostream>

using std::cout;

class Test

{

public:

    Test();

    ~Test();

};

Test::Test()

{

    cout << "Constructor is executed\n";

}

Test::~Test()

{

    cout << "Destructor is executed\n";

}

int main()

{

    delete new Test();

    return 0;

}

///////////////////////// Exercitiul 10 ////////////////////////////////////

//Asta nu e de POO, dar e fun :D. Ce afiseaza?

#include <iostream>

using namespace std;

int main()

{

   int i;

   for (i=0; i<3; i++);

   cout << "hello!" <<i;

   return 0;

}

///////////////////////// Exercitiul 11 ////////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

#include <iostream>

using namespace std;

int main()

{

    const int i = 20;

    const int\* const ptr = &i;

    (\*ptr)++;

    int j = 15;

    ptr = &j;

    cout << i;

    return 0;

}

///////////////////////// Exercitiul 12 ////////////////////////////////////

// Va rog sa nu ma urati pentru asta :D.

// Ce afiseaza sau de ce da eroare, daca da?

#include <iostream>

using namespace std;

int main()

{

    int num[5];

    int\* p;

    p = num;

    \*p = 10;

    p++;

    \*p = 20;

    p = &num[2];

    \*p = 30;

    p = num + 3;

    \*p = 40;

    p = num;

    \*(p + 4) = 50;

    for (int i = 0; i < 5; i++)

        cout << num[i] << ", ";

    return 0;

}

///////////////////////// Exercitiul 13 ////////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

#include <iostream>

using namespace std;

int main()

{

    int arr[] = { 4, 5, 6, 7 };

    int\* p = (arr + 1);

    cout << \*arr + 10;

    return 0;

}

///////////////////////// Exercitiul 14 ////////////////////////////////////

// Ce afiseaza sau de ce da eroare, daca da?

class gfg

{

        ~gfg()

        {

            cout << "I love coding";

        }

    public:    gfg()

        {

            cout << "I am at your service";

        }

};

int main()

{

   gfg mygfg;

}

///////////////////////// Exercitiul 15 ////////////////////////////////////

#include <iostream>

using namespace std;

class A

{

    public:

    A& operator=(const A&a)

    {

        cout << "A's assignment operator called" << endl;

        return \*this;

    }

};

class B

{

    A a[2];

};

int main()

{

    B b1, b2;

    b1 = b2;

    return 0;

}