Exercițiul 1

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class A
{
private:
    const int x = 5;

public:
    A() : x(2)
    {
    }
    const int getX() const
    {
        return x;
    }
};

int main()
{
    A *obj = new A();
    cout << obj->getX();
    return 0;
}
```

Exercițiul 2

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class A
{
private:
    int x = 5;

public:
    A() : x(2)
    {
        x = 6;
    }
    const int getX() const
```

```
    {
        return x;
    }
};

int main()
{
    A *obj = new A();
    cout << obj->getX();
    return 0;
}
```

Exercițiul 3

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Test
{
public:
    Test(Test &) {}
    Test() {}
};
Test fun()
{
    cout << "fun() Called\n";
    Test t;
    return t;
}
int main()
{
    Test t1;
    Test t2 = fun();
    return 0;
}
```

Exercițiul 4

Spuneți dacă programul de mai jos este corect. În caz negativ spuneți de ce nu este corect.

```
#include <iostream>
using namespace std;
class Test
{
    Test self;
};
```

```
int main()
{
    Test t;
    getchar();
    return 0;
}
```

Exercițiul 5

Spuneți dacă programul de mai jos este corect. În caz negativ spuneți de ce nu este corect.

```
#include <iostream>
using namespace std;
class Test
{
    static Test self;
};
int main()
{
    Test t;
    getchar();
    return 0;
}
```

Exercițiul 6

Spuneți dacă programul de mai jos este corect. În caz negativ spuneți de ce nu este corect.

```
#include <iostream>
using namespace std;
class Test
{
    Test *self;
};
int main()
{
    Test t;
    getchar();
    return 0;
}
```

Exercițiul 7

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Test
{
    static int x;

public:
    Test() { x++; }
    static int getX() { return x; }
};
int Test::x = 0;
int main()
{
    cout << Test::getX() << " ";
    Test t[5];
    cout << Test::getX();
}
```

Exercițiul 8

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Test
{
private:
    static int count;

public:
    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;
}
```

Exercițiul 9

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Cls
{
    int x;

public:
    Cls(int i) : x(i) {}
    const int &f()
    {
        return x;
    }
};
int main()
{
    Cls a(14);
    int b = a.f()++;
    cout << b << '\n';
    return 0;
}
```

Exercițiul 10

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class A
{
    int x;
    static int y;

public:
    A(int i, int j) : x(i), y(j)
    {
    }
    int f() const;
};
int A::y;
int A::f() const
{
    return y;
```

```
}
int main()
{
    const A a(21, 2);
    cout << a.f();
    return 0;
}
```

Exercițiul 11

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class A
{
    int x, *y;

public:
    A(int i)
    {
        x = i;
        y = new int[x];
    }
    A(A &a)
    {
        x = a.x;
        y = new int[x];
    }
    int getX() const
    {
        return x;
    }
};
int f(A a)
{
    return a.getX();
}
int main()
{
    const A a(5);
    cout << (a.getX() == f(a));
    return 0;
}
```

Exercițiul 12

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Cls
{
    int x;

public:
    Cls(int i = 0)
    {
        cout << 1;
        x = i;
    }
    Cls(Cls &ob)
    {
        cout << 2;
        x = ob.x;
    }
    int getX() const
    {
        return x;
    }
};
Cls &f(Cls &c)
{
    return c;
}
int main()
{
    Cls r;
    Cls s = f(f(f(r)));
    cout << s.getX();
    return 0;
}
```

Exercițiul 13

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Cls
{
    static int i;
    int j;
```

```
public:
    Cls(int x = 7)
    {
        j = x;
    }
    static int imp(int k)
    {
        Cls a;
        return i + k + a.j;
    }
    int getJ() const
    {
        return j;
    }
};
int Cls::i;
int main()
{
    int k = 5;
    cout << Cls::imp(k);
    return 0;
}
```

Exercițiul 14

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Cls
{
    int x;

public:
    Cls(int i = 32)
    {
        x = i;
    }
    int f() const
    {
        return x++;
    }
};
int main()
{
    const Cls d(-15);
    cout << d.f();
```

```
    return 0;
}
```

Exercițiul 15

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
int &fun()
{
    static int a = 10;
    return a;
}
int main()
{
    int &y = fun();
    y = y + 30;
    cout << fun();
    return 0;
}
```

Exercițiul 16

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#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;
}
```

Exercițiul 17

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class X
{
private:
    static const int a = 76;

public:
    static int getA() { return a; }
};
int main()
{
    cout << X::getA() << endl;
    return 0;
}
```

Exercițiul 18

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class A
{
    int aid;

public:
    A(int x)
    {
        aid = x;
    }
    void print()
    {
        cout << "A::aid = " << aid;
    }
};
class B
{
    int bid;

public:
    static A a;
```

```
    B(int i) { bid = i; }
};
int main()
{
    B b(10);
    b.a.print();
    return 0;
}
```

Exercițiul 19

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class A
{
    int id;
    static int count;

public:
    A()
    {
        count++;
        id = count;
        cout << "constructor called " << id << endl;
    }
    ~A()
    {
        cout << "destructor called " << id << endl;
    }
};
int A::count = 0;
int main()
{
    A a[2];
    return 0;
}
```

Exercițiul 20

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Test
{
private:
    int x;
    static int count;

public:
    Test(int i = 0) : x(i) {}
    Test(const Test &rhs) : x(rhs.x) { ++count; }
    static int getCount() { return count; }
};
int Test::count = 0;
Test fun()
{
    return Test();
}
int main()
{
    Test a = fun();
    cout << Test::getCount();
    return 0;
}
```

Exercițiul 21

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Cls
{
    int x;

public:
    int setX(int i)
    {
        int y = x;
        x = i;
        return x;
    }
    int getX() { return x; }
};
int main()
{
    Cls *p = new Cls[100];
    int i = 0;
```

```
    for (; i < 50; i++)
        p[i].setX(i);
    for (i = 5; i < 20; ++i)
    {
        cout << p[i].getX() << " ";
    }
    return 0;
}
```

Exercițiul 22

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Cls
{
    int x;

public:
    Cls(int i = 1)
    {
        x = i;
    }
    int setX(int i)
    {
        int y = x;

        x = i;
        return y;
    }
    int getX()
    {
        return x;
    }
};
int main()
{
    Cls *p = new Cls[10];
    int i = 0;
    for (; i < 10; ++i)
        p[i].setX(i);
    for (i = 0; i < 10; i++)
        cout << p[i].getX(i);
    return 0;
}
```

Exercițiul 23

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
int a = 2;
class Test
{
    int &t = a;

public:
    Test(int &t) : t(t) {}
    int getT() { return t; }
};
int main()
{
    int x = 20;
    Test t1(x);
    cout << t1.getT() << endl;
    x = 30;

    cout << t1.getT() << endl;
    return 0;
}
```

Exercițiul 24

Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect și realizați o modificare astfel încât acesta să compileze fără a-i schimba funcționalitatea.

```
#include <iostream>
using namespace std;
class Point
{
private:
    int x;
    int y;

public:
    Point(int i = 0, int j = 0);
    Point(const Point &t);
};
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;
    return 0;
}
```