

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 Test1
{
    int x;

public:
    void show() {}
};
class Test2
{
    int x;

public:
    virtual void show() {}
};
int main(void)
{
    cout << sizeof(Test1) << endl;
    cout << sizeof(Test2) << endl;
    return 0;
}
```

Exercițiul 2

Spuneți dacă programul de mai jos este corect. 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 P
{
public:
    virtual void show() = 0;
};
class Q : public P
{
    int x;
};
int main(void)
{
    Q q;
```

```
    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 A
{
public:
    virtual void fun() { cout << "A" << endl; }
};
class B : public A
{
public:
    virtual void fun() { cout << "B" << endl; }
};
class C : public B
{
public:
    virtual void fun() { cout << "C" << endl; }
};
int main()
{
    A *a = new C;
    A *b = new B;
    a->fun();
    b->fun();
    return 0;
}
```

Exercițiul 4

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
{
protected:
    int x;
}
```

```

public:
    A(int i = -16) { x = i; }
    virtual A f(A a) { return x + a.x; }
    void afisare() { cout << x; }
};

class B : public A
{
public:
    B(int i = 3) : A(i) {}
    A f(A a) { return x + a.x + 1; }
};

int main()
{
    A *p1 = new B, *p2 = new A, *p3 = new A(p1->f(*p2));
    p3->afisare();
    return 0;
}

```

Exercițiul 5

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>
#include <stdio.h>
using namespace std;
class Base
{
public:
    Base()
    {
        fun();
    }
    virtual void fun()
    {
        cout << "\nBase Function";
    }
};

class Derived : public Base
{
public:
    Derived() {}
    virtual void fun()
    {
        cout << "\nDerived Function";
    }
}

```

```
};

int main()
{
    Base *pBase = new Derived();
    delete pBase;
    return 0;
}
```

Exercițiul 6

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 base
{
public:
    virtual void show() { cout << " In Base \n"; }
};

class derived : public base
{
    int x;

public:
    void show() { cout << "In derived \n"; }
    derived() { x = 10; }
    int getX() const { return x; }
};

int main()
{
    derived d;
    base *bp = &d;
    bp->show();
    cout << bp->getX();
    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 A
{
protected:
    int x;

public:
    A(int i = -16)
    {
        x = i;
    }
    virtual A f(A a) { return x + a.x; }
    void afisare() { cout << x; }
};

class B : public A
{
public:
    B(int i = 3) : A(i) {}

    B f(B b)
    {
        return x + b.x + 1;
    }
};

int main()
{
    A *p1 = new B, *p2 = new A, *p3 = new A(p1->f(*p2));
    p3->afisare();
    return 0;
}

```

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 A
{
protected:
    int x;

public:
    A(int i = -31) { x = i; }
    virtual A operator+(A a) { return x + a.x; }
}

```

```
};

class B : public A
{
public:
    B(int i = 12) { x = i; }
    B operator+(B b) { return x + b.x + 1; }
    void afisare() { cout << x; }
};

int main()
{
    A *p1 = new B, *p2 = new A;
    B *p3 = new A(p2->operator+(*p1));
    p3->afisare();
    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 B
{
    int i;

public:
    B() { i = 1; }
    virtual int get_i() { return i; }
};
class D : public B
{
    int j;

public:
    D() { j = 2; }
    int get_i() { return B::get_i() + j; }
};
int main()
{
    const int i = cin.get();
    if (i % 3)
    {
        D o;
    }
    else
    {

```

```
        B o;  
    }  
    cout << o.get_i();  
    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 B  
{  
    int i;  
  
public:  
    B() { i = 1; }  
    virtual int get_i() { return i; }  
};  
class D : virtual public B  
{  
    int j;  
  
public:  
    D() { j = 2; }  
    int get_i() { return B::get_i() + j; }  
};  
class D2 : virtual public B  
{  
    int j2;  
  
public:  
    D2() { j2 = 3; }  
    int get_i() { return B::get_i() + j2; }  
};  
class MM : public D, public D2  
{  
    int x;  
  
public:  
    MM() { x = D::get_i() + D2::get_i(); }  
    int get_i() { return x; }  
};  
int main()  
{  
    B *o = new MM();  
    cout << o->get_i() << "\n";  
    MM *p = dynamic_cast<MM *>(o);
```

```

    if (p)
        cout << p->get_i() << "\n";
    D *p2 = dynamic_cast<D *>(o);
    if (p2)
        cout << p2->get_i() << "\n";
    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 B
{
protected:
    int x;

public:
    B(int i = 28) { x = i; }
    virtual B f(B ob) { return x + ob.x + 1; }
    void afisare() { cout << x; }
};
class D : public B
{
public:
    D(int i = -32) : B(i) {}
    B f(B ob) { return x + ob.x - 1; }
};
int main()
{
    B *p1 = new D, *p2 = new B, *p3 = new B(p1->f(*p2));
    p3->afisare();
    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 B

```



```

{
protected:
    int x;

public:
    B(int i = 12) { x = i; }
    virtual B f(B ob) { return x + ob.x + 1; }
    void afisare() { cout << x; }
};
class D : public B
{
public:
    D(int i = -15) : B(i - 1) { x++; }
    B f(B ob) { return x - 2; }
};
int main()
{
    B *p1 = new D, *p2 = new B, *p3 = new B(p1->f(*p2));
    p3->afisare();
    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 A
{
protected:
    static int x;

public:
    A(int i = 0) { x = i; }
    virtual A schimb() { return (7 - x); }
};
class B : public A
{
public:
    B(int i = 0) { x = i; }
    void afisare() { cout << x; }
};
int A::x = 5;
int main()
{
    A *p1 = new B(18);
    *p1 = p1->schimb();
    ((B *)p1)->afisare();
}

```

```
    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 B
{
protected:
    int x;
    B(int i = 10) { x = i; }

public:
    virtual B operator+(B ob)
    {
        B y(x + ob.x);
        return y;
    }
};
class D : public B
{
public:
    D(int i = 10) { x = i; }
    void operator=(B p) { x = p.x; }
    B operator+(B ob)
    {
        B y(x + ob.x + 1);
        return y;
    }
    void afisare() { cout << x; }
};

int main()
{
    D p1(-59), p2(32), *p3 = new D;
    *p3 = p1 + p2;
    p3->afisare();
    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;
class B
{
public:
    int x;
    B(int i = 0) { x = i; }

    virtual B aduna(B ob) { return (x + ob.x); }
    B minus() { return (1 - x); }
    void afisare() { cout << x; }
};

class D : public B
{
public:
    D(int i = 0) { x = i; }

    B aduna(B ob) { return (x + ob.x + 1); }
};

int main()
{
    B *p1, *p2;
    p1 = new D(138);
    p2 = new B(-37);
    *p2 = p2->aduna(*p1);
    *p1 = p2->minus();
    p1->afisare();
    return 0;
}
```