

Exercițiul 1

Fiind date următoarele tipuri de clase spuneți de câte ori este moștenită clasa B în clasa M1. Dar în clasa M2 ?

```
class B
{ /* instructiuni */
};
class D1 : virtual B
{ /* instructiuni */
};
class D2 : virtual B
{ /* instructiuni */
};
class D3 : B
{ /* instructiuni */
};
class D4 : private B
{ /* instructiuni */
};
class D5 : virtual public B
{ /* instructiuni */
};
class M1 : D1, public D2, D3, private D4, virtual D5
{ /* instructiuni */
};
class M2 : D1, D2, virtual D3, virtual D4, virtual D5
{ /* instructiuni */
};
```

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 B
{
protected:
    int x;

public:
    B() { x = 78; }
};

class D1 : virtual public B
```

```
{
public:
    D1() { x = 15; }
};
class D2 : virtual public B
{
public:
    D2() { x = 37; }
};
class C : public D2, public D1
{
public:
    int get_x() { return x; }
};
int main()
{
    C ob;
    cout << ob.get_x();
    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:
    int x;
    A(int i = 0) { x = i; }
    virtual A minus() { return (1 - x); }
};

class B : public A
{
public:
    B(int i = 0) { x = i; }
    void afisare() { cout << x; }
};

int main()
{
    A *p1 = new B(18);
    *p1 = p1->minus();
    p1->afisare();
}
```

```
    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
{
public:
    int x;
    A(int i = 0) { x = i; }
    virtual A minus() { cout << x; }
};

class B : public A
{
public:
    B(int i = 0) { x = i; }
    void afisare() { cout << x; }
};

int main()
{
    A *p1 = new A(18);
    *p1 = p1->minus();
    dynamic_cast<B *>(p1)->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>
using namespace std;

class A
{
public:
    int x;
```

```
    A(int i = 13) { x = i; }
};

class B : virtual public A
{
public:
    B(int i = 15) { x = i; }
};

class C : virtual public A
{
public:
    C(int i = 17) { x = i; }
};

class D : public A
{
public:
    D(int i = 19)
    {
        x = i;
    }
};

class E : public B, public C, public D
{
public:
    int y;
    E(int i, int j) : D(i), B(j)
    {
        y = x + i + j;
    }
    E(E &e) { y = -e.y; }
};

int main()
{
    E e1(-9.3), e2 = e1;
    cout << e2.y;
    return 0;
}
```