

Examen scris

- I. Spuneți dacă programul de mai jos este corect. În caz afirmativ, precizați exact constructorii care se execută și în ce ordine. În caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
class cls1
{ public: int x;
      cls1(int i=13) { x=i; } };
class cls2: virtual public cls1
{ public: cls2(int i=15) { x=i; } };
class cls3: virtual public cls1
{ public: cls3(int i=17) { x=i; } };
class cls4: public cls1
{ public: cls4(int i=19) { x=i; } };
class cls5: public cls2, public cls3, public cls4
{ public: int y;
      cls5(int i,int j):cls4(i),cls2(j){ y=i+j; }
      cls5(cls5& ob) ){ y=-ob.y; }};
int main()
{ cls5 ob1(-9,3), ob2=ob1;
  cout<<ob2.y;
  return 0;
}
```

- II. Descrieți pe scurt constructorul de copiere și situațiile în care acesta este folosit..

- III. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
class cls
{ int x;
  const int y;
  public: cls(int i, int j):x(i), y(j) { }
        int imp(int, int) const; };
int imp(int i, int j) const
{ x=i; y=j; return x+y; }
int main()
{ cls ob(5);
  cout<<ob.imp(-9,8);
  return 0;
}
```

- IV. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include <iostream.h>
class B
{
    int a;
    public: B(int i=0) { a=i; }
           int get_a(){ return a; } };
class D: private B
{
    public: D(int x=0): B(x) {}
           int get_a() { return B::get_a(); } };
int main()
{
    D d(-89);
    cout<<d.get_a();
    return 0;
}
```

- V. Descrieți pe scurt ce reprezintă obiectul implicit al unei funcții.

- VI. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
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;
}
```

VII. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
class cls
{
    public: int sa;
           cls(int s=0) { sa=s; }
           operator int() { return sa; }
           int f(int c) { return (sa*(1+c/100)); } };

int main()
{
    cls p(37);
    cout<<p.f(p);
    return 0;
}
```

VIII. Descrieți diferența dintre un pointer și o referință.

IX. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
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;
}
```

- X. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include <iostream.h>
template <class tip>
class cls
{ tip z;
  public: cls(tip i) { z=i; }
         tip operator-(cls); };
template <class tip>
tip cls<tip>::operator-(cls<tip> a)
{ return z-a.z;
}
template <class tip>
tip dif(tip x, tip y)
{ return x-y;
}
int dif(int x, int y)
{ return x>=y?x-y:y-x;
}
int main()
{ cls<int> i=3; cls<float> j=4;
  cout<<dif(i,j);
  return 0;
}
```

- XI. Descrieți pe scurt cum puteți prelua o dată prin incluziune și a doua oară prin moștenire o clasă `numar` într-o clasă `lista` care descrie liste nevide de dimensiune variabilă de elemente de tip `numar`.

- XII. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
class cls
{ static int x;
  public: cls(int i=25) { x=i; }
         friend int& f(cls); };
int cls::x=-13;
int& f(cls c) { return c.x; }
int main()
{ cls d(15);
  cout<<f(d);
  return 0;
}
```

XIII. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
class cls
{
    int v,nr;
    public: cls(int i) { nr=i; v=new int[i]; }
           friend int& operator[](int);
           friend ostream& operator<<(ostream&,cls); };
int& operator[](cls& x, int i)
{ return x.v[i]; }
ostream& operator<<(ostream& o, cls x)
{ for(int i=0;i<x.nr;i++) cout<<x.v[i]<<" "; return o; }
int main()
{ cls x(10);
  x[5]=7;
  cout<<x;
  return 0;
}
```

XIV. Descrieți pe scurt metoda de identificare a tipului în timpul rulării (RTTI).

XV. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
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 cls::i;
int main()
{ int k=5;
  cout<<cls::imp(k);
  return 0;
}
```

XVI. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează, în caz negativ spuneți de ce nu este corect.

```
#include<iostream.h>
class cls
{
    int x;
    public: cls(int i=32) { x=i; }
           int f() const; };
int cls::f() const { return x++; }
void main()
{
    const cls d(-15);
    cout<<d.f();
}
```

XVII. Spuneți dacă o variabilă constantă poate fi transmisă ca parametru al unei funcții și dacă da, în ce situații. Justificați.

XVIII. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afișează pentru o valoare întreagă citită egală cu 7, în caz negativ spuneți de ce nu este corect.

```
#include <iostream.h>
float f(float f)
{ if (f) throw f;
  return f/2;
}
int main()
{ int x;
  try
  {
    cout<<"Da-mi un numar intreg: ";
    cin>>x;
    if (x) f(x);
    else throw x;
    cout<<"Numarul "<<x<<" e bun!"<<endl;
  }
  catch (int i)
  { cout<<"Numarul "<<i<<" nu e bun!"<<endl;
  }
  return 0;
}
```