

Nume :

DL Informatică

03.09.2011

Grupa :

Tutore laborator :

IV. Cum se poate face supraîncărcarea operatorilor ca funcții independente în C++. Particularități.
(cum se face, sintaxa, de ce așa, alternativa, ce nu se poate supraîncărca, exemplu, etc.)

V. Descrieți pe scurt transmiterea parametrilor unei funcții prin referință.
(cum se face, sintaxa, de ce așa, alternativa, ce nu se poate transmite, exemplu, etc.)

VI. Spuneți pe scurt ce este lista de inițializare a unui constructor și subliniați importanța ei.

- VII. Scrieți un program cu două clase care: în prima clasă să citească de la tastatură un întreg și să calculeze suma \underline{s} a digiților pari ai numărului respectiv. Apoi în a doua clasă să se calculeze produsul tuturor numerelor întregi de la suma \underline{s} (calculata de către prima clasă) la valoarea 30 și să afișeze rezultatul pe ecran. (ex. nr=23856378, $s=2+8+6+8=24$, $p=24*25*26*27*28*29*30$)

- VIII. 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 CL
{ static int x;
  public: CL(int i=3) {x=i; }
  int get_x() { return x; }
  int& set_x(int i) { x=i;return x;}
  CL operator=(CL a1) { set_x(a1.get_x()); return a1;}
} a(19);
int CL::x;
int main()
{ CL b;
  cout<<(b=a).get_x();
  return 0;
}
```

- 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 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 plus() { cout<<x++; } };
int main()
{
    A *p1=new B(23);
    *p1=p1->minus();
    p1->plus();
    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>
class B
{
    protected: int x;
    public: B(int i=1) { x=i; } };
class D: B
{
    public: D(D& o) { x=o.x+17; }
    void afisare() { cout<<x<<" "; } };
int main()
{
    D o1, o2(o1);
    o2.afisare();
    return 0;
}
```

- XI. 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=5) { x=i; }
        virtual B f(B ob) { return x+ob.x+10; }
        void afisare(){ cout<<x; } };
class D: public B
{
    public: D(int i=-7):B(i) {}
        B f(B ob) { return x+ob.x-3; } };
int main()
{
    B *p1=new D, *p2=new B, *p3=new B(p1->f(*p2));
    p3->afisare();
    return 0;
}
```

XII. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce constructori și destructori sunt executați în programul de mai jos și în ce ordine. În caz negativ spuneți de ce nu este corect.

```
#include <iostream>
using namespace std;

struct B{    int i;
    public: B(int x=1){i=x; }
} ob0;
class A {    int who;
            B b;
public: A(int id):b(id-1){who=id;};
        ~A();
} ob1(1), ob2(2);
A::~~A() { cout << "Aha! " << who << "\n";
}
class D: public A, B{
    public: D(int i):A(i),B(i){}
};
void f(D ob) {};
int main()
{    A a(1);
    B b(2);
    D d(3);
    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 A
{    public: int x;
    A(int i=-15) { x=i; } };
class B: virtual public A
{    public: B(int i=-11) { x=i; } };
class C: virtual public A
{    public: C(int i=-19) { x=i; } };
class D: virtual public A
{    public: D(int i=-59) { 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& ob) { y=ob.x+ob.y; } };
int main()
{    E e1(-33,14), e2=e1;
    cout<<e2.y;
    return 0;
}
```

XIV. 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>
using namespace std;

int f(int y) { if (y<0) throw y;
return y/2; }

int f(int y, int z) { if (y<z) throw z-y;
return y/2; }

float f(int &yy, int z, int y=9)
{ cout<<" y este referinta";
return (float) yy/2 ; }

int main()
{    int x;
    try
    {
        cout<<"Da-mi un numar par: ";
        cin>>x;
        if (x%2) x=f(x);
        else x=f(x-1);
        cout<<"Numarul "<<x<<" e bun!"<<endl;
    }
    catch (int i)
    { cout<<"Numarul "<<i<<" nu e bun!"<<endl;}
    return 0;
}
```

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>
template<class T, class U>
T f(T x, U y)
{ return x+y;
}

int *f(int *x, int y)
{ return x-y;
}

int main()
{ int *a=new int(100), b(6);
  cout<<*f(a,b);
  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>
#include <typeinfo>
struct B
{ int i;
  public: B() { i=1; }
          int get_i() { return i; }
};
class D: public B
{ int j;
  public: D() { j=2; }
          int get_j() {return j; }
};
int main()
{ B *p=new D;
  cout<<p->get_i();
  if (typeid((B*)p).name()=="D*") cout<<((D*)p)->get_j();
  return 0;
}
```

XVII. 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>

int mister(int a){
    int m,n;
    if (a<=3) return a;
    else {cout<<a+5;
          m=a+2;
          n=mister(a-1)+mister(a-3);
          cout<<m+n;
          return n;}
}

int main(){
    int i;
    cout<<mister(6);
    return 0;
}
```

XVIII. 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: static int x;
  int i;
  public: B() { x++; i=1; }
         ~B() { x--; i--;}
         static int get_x() { return x; }
         int get_i() { return i; } };

int B::x;
class D: public B
{ public: D() { x++;}
         ~D() { x--;}
         int f1(B o){return 5+o.get_i();}   };

int f(B *q)
{ return (q->get_x())+1; }

int main()
{ B *p=new B[10];
  cout<<f(p);
  delete[] p;
  p=new D;
  cout<<p->f1(p);
  delete p;
  cout<<D::get_x();
  return 0;
}
```