

## PROGRAMARE ORIENTATĂ PE OBIECTE – EXAMEN SCRIS

- I. Descrieți pe scurt cele două feluri de folosire a cuvântului cheie “virtual” la moștenire și în ce cazuri se folosesc.
- II. Cum se poate supraîncărca operatorul [] și care este folosirea uzuală a acestei supraîncărcări?
- III. Spuneți pe scurt ce este lista de inițializare a unui constructor și subliniați importanța ei.
- IV. Descrieți pe scurt mecanismul de tratare a excepțiilor.
- V. Descrieți pe scurt întoarcerea rezultatului unei funcții prin referință.
- VI. Spuneți pe scurt la ce se referă supraîncărcarea funcțiilor șablon (template).
- VII. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afisează, în caz negativ spuneți de ce nu este corect.

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

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

```
#include <iostream.h>
class A
{ static int x;
  public: A(int i=0) {x=i; }
  int get_x() { return x; }
  int& set_x(int i) { x=i;}
  A operator=(A a1) { set_x(a1.get_x()); return a1;}
};
int main()
{ A a(212), b;
  cout<<(b=a).get_x();
  return 0;
}
```

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

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

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

```
#include <iostream.h>
class problema
{ int i;
  public: problema(int j=5){i=j;}
  void schimba(){i++;}
```

Nume :.....

An I Informatică

31.05.2010

Grupa :.....

```
        void afiseaza(){cout<<"starea curenta "<<i<<"\n";}
};
problema mister1() { return problema(6);}
void mister2(problema &o)
{   o.afiseaza();
    o.schimba();
    o.afiseaza();
}
int main()
{   mister2(mister1());
    return 0;
}
```

Notă : Fiecare subiect are 0,5 pcte. Se acordă 1 pct din oficiu. Timp de lucru 2 ore

Grupa :.....

XI. Spuneți câți constructori sunt executați în programul de mai jos și în ce ordine.

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

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

```
#include<iostream.h>
class B
{ int x;
  public: B(int i=7) { x=i; }
          int get_x() { return x; }
          operator int() { return x; } };
class D: public B
{ public: D(int i=-12):B(i) {}
      D operator+(D a) { return get_x()+a.get_x()+1; } };
int main()
{ D a; int b=18;
  b+=a;
  cout<<b;
  return 0;
}
```

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

```
#include <iostream.h>
#include <typeinfo>
class B
{ int i;
  public: B() { i=1; }
          int get_i() { return i; }
};
class D: 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;
}
```

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

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

Grupa :.....

XV. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afisează, 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--; }
    static int get_x() { return x; }
    int get_i() { return i; } };
int B::x;
class D: public B
{ public: D() { x++; }
    ~D() { x--; } };
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<<f(p);
  delete p;
  cout<<D::get_x();
  return 0;
}
```

XVI. Spuneți dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afisează, 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(3), b(23);
  cout<<*f(a,b);
  return 0;
}
```

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

```
#include<iostream.h>
class A
{ int x;
  public: A(int i=0) { x=i; }
         A operator+(const A& a) { return x+a.x; }
         template <class T> ostream& operator<<(ostream&); };
template <class T>
ostream& A::operator<<(ostream& o) { o<<x; return o; }
int main()
{ A a1(33), a2(-21);
  cout<<a1+a2;
  return 0;
}
```

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

```
#include <iostream.h>
class cls
{ int x;
  public: cls(int i) { x=i; }
         int set_x(int i) { int y=x; x=i; return y; }
         int get_x(){ return x; } };
int main()
{ cls *p=new cls[10];
  int i=0;
  for(;i<10;i++) p[i].set_x(i);
  for(i=0;i<10;i++) cout<<p[i].get_x();
  return 0;
}
```