Nume …….. Nr. 2 13 iunie 2016

Grupa ……. Programare orientată pe obiecte

Examen scris – anul I și grupa 211

I. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include<iostream>
using namespace std;
class A
{ int i;
  public: A(int x=2):i(x+1) {}
  virtual int get_i() { return i; }};
class B: public A
{ int j;
  public: B(int x=20):j(x-2) {}
  virtual int get_j() {return A::get_i()+j; }};
int main()
{ A o1(5);
  B o2;
  cout<<o1.get_i()<<" ";
  cout<<o2.get_j()<<" ";
  cout<<o2.get_i();
  return 0;
}
6 21 13 0.5p
Spune ca ar compila ... 0.1p
```

II. Descrieți pe scurt destructorii (sintaxă, particularități, utilitate, exemplu).

III. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneți ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
class A
{   protected: static int x;
    private: int y;
    public: A(int i) { x=i; y=-i+4; }
        int put_x(A a) { return a.x+a.y; } };
int A::x=7;
int main()
{   A a(10);
```

```
    cout<<a.put_x(20);
    return 0;
}
```

4 0.5p

Alta valoare dar afira ca ar compila 0.1 p

IV. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include<iostream>
using namespace std;
template<class X>
int functie(X x, X y)
{ return x+y;
}
int functie(int & x, int *y)
{ return x-*y;
}
int main()
{ int a=7, *b=new int(4);
  cout<<functie(a,b);
  return 0;
}
```

3 0.5p

Alta valoare 0.1p

V. Descrieţi pe scurt caracteristicile pointerului this (sintaxă, tip, utilitate, unde apare, unde nu apare).

VI. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
class B
{ int i;
  public: B() { i=80; }
          virtual int get_i() { return i; }
};
class D: public B
{ int j;
  public: D() { j=27; }
          int get_j() {return j; }
```

```
};
int main()
{ D *p=new B;
  cout<<p->get_i();
  cout<<((D*)p)->get_j();
  return 0;
}
```

Sunt 2 variante: D *p=new B; Nu compileaza ; 0.5p
Probleme la rulare, deoarece clasa B nu contine functia get_j. 0.3p

VII. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
class B
{ protected: int i;
  public: B(int j=5) {cout << " cb "; i=j-2; }
          ~B(){ cout << " db ";}
          int get_i() { return i; } };

class D1: public B
{ int j;
  public: D1(int k=10) {cout << " cd1 "; j=i-k+3; }
          ~D1(){ cout << " dd1 ";} };

 class D2: public D1
{ int k;
  public: D2(int l=15) {cout << " cd2 "; k=i-l+3; }
           ~D2(){ cout << " dd2 ";}  };
D1 f(D1 d1, D2 d2) {return d1.get_i()+d2.get_i(); }
int main()
{ D2 ob2; D1 ob1(3);
  cout<<f(ob1,ob2).get_i()+ob2.get_i();
  return 0;
}
```

Compileaza si afiseaza
cb cd1 cd2 cb cd1 cb cd1 6 dd1 db dd1 db dd2 dd1 db dd1 db dd2 dd1 db
Micile neatentii le trecem cu vederea. 0.5p

VIII. Asemănări şi diferențe între pointeri şi referințe.

IX. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
class A
{   static int *x;
    public: A() {}
            int get_x() { return (++(*x))++; } };
int *A::x(new int(19));
int main()
{   A *p=new A,b;
    cout<<b.get_x()<<" ";
    cout<<p->get_x();
    return 0;
}
```

Afiseaza 20 22 0.5p
Alte valori 0.1p

X. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
class X {   int i;
public:   X(int ii = 5) { i = ii; cout<< i<< " ";};
          const int tipareste(int j) const { cout<<i<< " "; return i+j; };
};
int main()
{ X O (7);
  O.tipareste(5);
  X &O2=O;
  O2.tipareste(6);
  const X* p=&O;
  cout<<p->tipareste(7);
  return 0;
}
```

7 7 7 7 14 0.5p
Alte valori decente (a uitat un 7 pe parcurs) 0.3p
Alte valori 0.1p

XI. Descrieţi pe scurt lista de iniţializare a constructorilor şi utilizarea ei (sintaxă, utilizare, moştenire, compunere).

XII. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
class A
{   protected: int x;
    public: A(int i):x(i){}
            int get_x(){ return x; } };
class B: A
{   protected: int y;
    public: B(int i,int j):y(i),A(j){}
            int get_y(){ return get_x()+y; } };
class C: protected B
{   protected: int z;
    public: C(int i,int j,int k):z(i),B(j,k){}
            int get_z(){ return get_x()+get_y()+z; } };
int main()
{   C c(5,6,7);
    cout<<c.get_z();
    return 0;
}
```

Nu compileaza, clasa B mosteneste private din A astfel ca functia get_x() nu mai este accesibila in C 0.5p

XIII. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
class cls
{     int x;
public: cls(int i=2) { 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[15];
      for(int i=2;i<8;i++) p[i].set_x(i);
        for(int i=1;i<6;i++) cout<<p[i].get_x();
  return 0;
}
```

22345 0.5p
Alte valori 0.1p

XIV. Descrieți pe scurt diferențele dintre metodele virtuale şi cele nevirtuale (minim 5 diferențe).

XV. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
struct X {   int i;
public:   X(int ii ) { i = ii; };
     int f1() { return i; }
X f2() const {   int i=this->f1(); return X(34-i); }};
const X f3() {   return X(16); }
int f4(const X& x) { return x.f1(); }
int main() {X ob(11);
    cout<<f4(ob.f2().f1());
    return 0;
}
```

Nu compileaza din cauza ca incercam sa apelam o functie neconst pentru un obiect const (sunt mai multe situatii in acest program in care nu tinem cont de calificativul const) 0.5p

XVI. Spuneţi dacă programul de mai jos este corect. În caz afirmativ, spuneţi ce afişează, în caz negativ propuneți o (singură) modificare prin care programul devine corect.

```
#include <iostream>
using namespace std;
class A
{ protected: static int x;
  public: A(int i=1) {x=i; }
  int get_x() { return x; }
  int& set_x(int i) { int y=x; x=i; return y;}
  A operator=(A a1) { set_x(a1.get_x()); return a1;}
} a(33);
int main()
{ A a(18), b(7);
  cout<<(b=a).set_x(27);
  return 0;
}
```

Static initializat in afara clasei, eroare de linking 0.5p

XVII. Descrieți pe scurt metodele constante (sintaxă, particularități, importanță, exemplu).

XVIII. Scrieți un program scurt care să aibă cel puțin o moștenire şi să calculeze suma numerelor naturale divizibile cu 17 dintre 50 şi 1000.