

POO – Test 2 A-Barem

14.06.2010

Observații:

1. Nu este permisă consultarea bibliografiei. **2.** Toate întrebările sunt obligatorii. **3.** Dacă nu este precizat altfel, fiecare întrebare este notată cu 3 puncte. Repartiția punctelor la întrebările grilă este: 1 punct alegerea corectă a variantei, 2 puncte justificarea. Alegerea corectă se punctează numai dacă justificarea este total sau parțial corectă. **4.** Nu este permisă utilizarea de foi suplimentare.

1) În codul de mai jos sunt folosite concepte C++: încapsulare, membri static, derivare, polimorfism, spații de nume, operator de alocare, constructori. Explicați 3 din aceste concepte și precizați locul unde au fost folosite. Care este rezultatul execuției programului?

<pre>#include <iostream> using namespace std; class A{public:virtual char label() = 0;}; class B:public A{ public: char label(){return 'b';} protected: static int b; }; class C:public B{ public: C(){b++;} char label(){ cout << b; return 'c';} }; int B::b = 0; int main() { A* a = new C; cout << a->label() << " "; B* b = new C; cout << b->label() << " "; delete a;delete b;return 0; }</pre>	<p>Răspuns.</p> <p>3x0.5p = 1.5p main()(explicatii+rezultat) 1.5p (Rezultatul : 1c 2c)</p>
<p>2) Ce va afișa programul de mai jos după execuție?</p> <pre>#include <iostream> using namespace std; struct pereche{int x, y;}; pereche operator!(const pereche& p){ pereche q; q.x = p.y; q.y = p.x; return q; } ostream& operator<<(ostream& o, pereche p){ o << "(" << p.x << ", " << p.y << ")"; return o; } int main(){ pereche w = { 2, 5 }; cout << !w << endl; return 0; }</pre>	<p>a) nimic deoarece conține erori de sintaxa b) (2, 5) c) (5, 2) CORECT d) (0, 0)</p> <p>Justificare. Membrii struct sunt public: 0.5p Supraincarcare operator!: 0.5p Supraincarcare operator<< 0.5p Concluzie 0.5p</p>
<p>3) Ce va afișa programul de mai jos după execuție?</p> <pre>#include <iostream> using namespace std; class A { public: A() { cout << "A() "; } };</pre>	<p>a) C() B() A() E() CORECT b) B() A() C() E() c) A() B() C() E() d) E() C() B() A() e) nimic deoarece conține erori de sintaxă</p>

<pre> class B { public: B() { cout << "B() "; } }; class C { public: C() { cout << "C() "; } }; class E { public: E() { cout << "E() "; } E(A una, B unb, C unc) : b(unb), a(una), c(unc) { cout << "E() "; } private: C c; B b; A a; }; int main() { E e; return 0; } </pre>	<p>Justificare.</p> <p>Obiecte membru in clasa E 0.5</p> <p>Lista de initializare 0.5</p> <p>Ordinea de apel este cea din declaratii 0.5</p> <p>Concluzia 0.5</p>
<p>4)</p> <pre> #include <iostream> using namespace std; class A { public: A(){cout <<"A() ";} ~A(){cout <<"~A() ";} virtual void f() { cout<<"A::f ";} void g(){cout << "A::g ";} }; class B: public A { public: B(){ cout << "B() "; } ~B(){cout <<"~B() ";} void f() { cout<<"B::f ";} void g() { cout << "B::g() ";} }; void main() { A *b = new B; b->f(); ((B*)b)->f(); b->g(); ((B*)b)->g(); delete b; } </pre>	<p>Ce va afișa programul alăturat?</p> <p>a) A() B() A::f B::f A::g B::g() ~A()</p> <p>b) A() B() B::f B::f A::g B::g() ~B() ~A()</p> <p>c) A() B() B::f B::f A::g B::g() ~A() CORECT</p> <p>d) A() B() A::f B::f A::g B::g() ~B() ~A()</p> <p>Justificare.</p> <p>B derivată din A, constructori, destructori 0.5p</p> <p>f polimorfă, g nepolimorfă 0.5p</p> <p>apel f in main 0.5p</p> <p>apel g in main 0.5p</p>
<p>5) Explicați codul de mai jos și descrieți un scenariu de execuție.</p> <pre> #include <iostream> using namespace std; template <class X> class input { X data; public: input(char *s, X min, X max); }; template <class X> input<X>::input(char *s, X min, X max){ do { cout << s << ": "; cin >> data; } while(data < min data > max); } int main(){ input<int> i("enter int", 0, 10); input<char> c("enter char", 'A', 'Z'); return 0; } </pre>	<p>Răspuns.</p> <p>Clasa input parametrizată 0.5p</p> <p>Constructorul clasei 0.5p</p> <p>Obiectul i 0.5p</p> <p>Obiectul c 0.5p</p> <p>Un scenariu 1p</p> <pre> enter int: 45 enter int: 83 enter int: 9 enter char: a enter char: f enter char: H </pre>

<p>6) Explicați codul de mai jos și precizați rezultatul execuției</p> <pre>#include <iostream> using namespace std; void f(int n) { try { if (n > 0) throw "numar>0; "; else throw "numar<=0; "; } catch(const char * s) { cout << "in f(): " << s << ' '; throw ; } } int main() { cout << "Start\n"; try{ f(-30); } catch(const char * s) { cout << "in main: " << s << ' ' ; } cout << "End"; return 0; }</pre>	<p>Răspuns.</p> <table border="0"> <tr> <td>Bloc try in f</td><td>0.5p</td></tr> <tr> <td>catch in f</td><td>0.5p</td></tr> <tr> <td>Rethrow in f</td><td>0.5p</td></tr> <tr> <td>Bloc try in main</td><td>0.5p</td></tr> <tr> <td>catch în main</td><td>0.5p</td></tr> <tr> <td>rezultat</td><td>0.5p</td></tr> </table> <p>Start in f(): numar<=0; in main: numar<=0; End</p>	Bloc try in f	0.5p	catch in f	0.5p	Rethrow in f	0.5p	Bloc try in main	0.5p	catch în main	0.5p	rezultat	0.5p
Bloc try in f	0.5p												
catch in f	0.5p												
Rethrow in f	0.5p												
Bloc try in main	0.5p												
catch în main	0.5p												
rezultat	0.5p												
<p>7) Explicați codul de mai jos și precizați rezultatul execuției</p> <pre>#include <iostream> #include <stack> #include <list> using namespace std; int main() { int data[] = {21, 43, 23, 12, 15, 42}; stack<int, list<int>> s; for (int i = 1; i < 5; ++i) s.push(data[i]); s.pop(); cout << s.top() << ' '; s.push(data[0]); while (!s.empty()) { cout << s.top() << ' '; s.pop(); } cout << s.size() << endl; return 0; }</pre>	<p>Răspuns.</p> <p>12 21 12 23 43 0</p> <table border="0"> <tr> <td>S – stiva de int implementată cu list<int></td><td>0.5p</td></tr> <tr> <td>Conținut după for: 15 12 23 43</td><td>0.5p</td></tr> <tr> <td>top după pop: 12</td><td>0.5p</td></tr> <tr> <td>s.push(data[0]) 21</td><td>0.5p</td></tr> <tr> <td>dupa while: 21 12 23 43</td><td>0.5p</td></tr> <tr> <td>s.size() = 0</td><td>0.5p</td></tr> </table>	S – stiva de int implementată cu list<int>	0.5p	Conținut după for: 15 12 23 43	0.5p	top după pop: 12	0.5p	s.push(data[0]) 21	0.5p	dupa while: 21 12 23 43	0.5p	s.size() = 0	0.5p
S – stiva de int implementată cu list<int>	0.5p												
Conținut după for: 15 12 23 43	0.5p												
top după pop: 12	0.5p												
s.push(data[0]) 21	0.5p												
dupa while: 21 12 23 43	0.5p												
s.size() = 0	0.5p												

8) (6 puncte)

Să se scrie codul C++ pentru implementarea unei clase NumarMare a cărei instanțe sa fie reprezentări de numere întregi pozitive foarte mari. Numărul maxim de cifre este 30(reprezentarea se poate face cu string, vector de caractere, tablou de caractere).

Responsabilitățile clasei sunt:

- | | |
|--|------|
| Declarația clasei | 0.5p |
| a) inițializează un număr mare având ca intrare un șir de cifre; | 0.5p |
| b) inițializează un număr mare având ca intrare alt număr mare(copiere); | 1p |
| c) atribuie un număr mare altui număr mare (operator=); | 1p |
| d) afișează numărul (operator<<); | 1p |
| e) adună două numere mari (operator+). | 2p |