Problema 12:

Varianta 1: identic cu varianta 3

#include <iostream>

using namespace std;

class A

{ int x;

public:

A(int i=2):x(i){}

int get\_x(){ return x;}

A operator+(int);

};

ostream& operator<<(ostream& o, A a)

{o<<a.get\_x(); return o; }

A A::operator+(int i){return x+i;}

int main()

{ A a=-33; int b=44;

cout<<a+b<<" "<<b+3;

return 0;}

Varianta 2: NU COMPILEAZA ca in operatorul << avem a.x si nu avem acces la x

#include <iostream>

using namespace std;

class A

{ int x;

public:

A(int i=2):x(i){}

int get\_x(){ return x;}

A operator+(int);

};

ostream& operator<<(ostream& o, A a)

{o<<a.x; return o; }

A A::operator+(int i){return x+i;}

int main()

{ A a=-33; int b=44;

cout<<a+b<<" "<<b+3;

return 0;}

-se apeleaza constructorul de initializare cu -33 si

se apeleaza op + supraincarcat 0.1p

-se intoarce un int care e convertit prin constructor la un obiect de tip A 0.1p

-pentru acesta se apeleaza op << supraincarcat

- a.x e privat 0.1p

-varianta corecta 0.1p

Varianta 3: COMPILEAZA si afiseaza 11 47

#include <iostream>

using namespace std;

class A

{ int x;

public:

A(int i=2):x(i){}

int get\_x(){ return x;}

A operator+(int);

};

ostream& operator<<(ostream& o, A a)

{o<<a.get\_x(); return o; }

A A::operator+(int i){return x+i;}

int main()

{ A a=-33; int b=44;

cout<<a+b<<" "<<b+3;

return 0;}

-se apeleaza constructorul de initializare cu -33 0.1p

- se apeleaza op + supraincarcat 0.1p

-se intoarce un int care e convertit prin constructor la un obiect de tip A 0.1p

-pentru acesta se apeleaza op << supraincarcat si op << pt int 0.1p

- rezultatul 11=-33+44 47=44+3 0.1 p

Varianta 4: COMPILEAZA si afiseaza -11 -41

#include <iostream>

using namespace std;

class A

{ int x;

public:

A(int i=2):x(i){}

int get\_x(){ return x;}

A operator+(int);

};

ostream& operator<<(ostream& o, A a)

{o<<a.get\_x(); return o; }

A A::operator+(int i){return x+i;}

int main()

{ A a=33; int b=-44;

cout<<a+b<<" "<<b+3;

return 0;}

-se apeleaza constructorul de initializare cu 33 0.1p

- se apeleaza op + supraincarcat 0.1p

-se intoarce un int care e convertit prin constructor la un obiect de tip A 0.1p

-pentru acesta se apeleaza op << supraincarcat si op << pt int 0.1p

-rezultatul -11=33-44 -41=-44+3 0.1p

Varianta 5: COMPILEAZA si afiseaza -77 -47

#include <iostream>

using namespace std;

class A

{ int x;

public:

A(int i=2):x(i){}

int get\_x(){ return x;}

A operator+(int);

};

ostream& operator<<(ostream& o, A a)

{o<<a.get\_x(); return o; }

A A::operator+(int i){return x+i;}

int main()

{ A a=-33; int b=-44;

cout<<a+b<<" "<<b-3;

return 0;}

-se apeleaza constructorul de initializare cu -33 0.1p

- se apeleaza op + supraincarcat 0.1p

-se intoarce un int care e convertit prin constructor la un obiect de tip A 0.1p

-pentru acesta se apeleaza op << supraincarcat si op << pt int 0.1p

-rezultatul -77=-33-44 -47=-44-3 0.1p