**Exercitiul 4**

**var 1**

#include <iostream>

using namespace std;

class C {

float z;

public:

C() { z = 1.3; }

float op(float x, float y, float t) {

return x + y + z + t;

}

float op(float x, float y = 1.0) {

return x + y + z;

}

float op() {

return z;

}

};

int main() {

C c;

float i,j, k;

i=c.op(1);

j=i+c.op(2.2, 4.8);

k=c.op(2.2, 3.5, 4);

return 0;

}

//Compileaza

//Se afiseaza

//3.3

//11.6

//11

//compileaza si valori rezonabile 0.1

//pentru 3.3 0.2

//pentru 11.6 0.1

//pentru 11 0.1

------------------------------------------------------

**var 2**

#include <iostream>

using namespace std;

class C {

float z;

public:

C() { z = 1.3; }

float op(float x, float y, float t) {

return x + y + z + t;

}

float op(float x, float y = 1.0) {

return x + y + z;

}

float op(float x) {

return x + z;

}

};

int main() {

C c;

float i,j, k;

i=c.op(1.2);

j=i+c.op(2.2, 4.8);

k=c.op(2.2, 3.5, 4);

return 0;

}

//Nu compileaza 0.1

//Eroarea apare deoarece apelul din main de la linia 22 (primul apel catre op din main) este ambiguu

//0.2

//Se considera candidati valizi atat metoda op de la linia 11 cat si cea de la linia 14

//Solutia pentru a rezolva eroarea este sa indepartam valoarea default de la linia 11 pentru variabila y (a doua metoda op din clasa C)

// 0.2

-------------------------------------------------------

**var 3**

#include <iostream>

using namespace std;

class C {

float z;

public:

C() { z = 1.3; }

float op(float x, float y, float t) {

return x + y + z + t;

}

float op(float x = 1.0, float y) {

return x + y + z;

}

float op() {

return z;

}

};

int main() {

C c;

float i,j, k;

i=c.op(1.2);

j=i+c.op(2.2, 4.8);

k=c.op(2.2, 3.5, 4);

return 0;

}

//Nu compileaza 0.1

//Parametrii default trebuie sa apara ultimii in enumeratia de la linia 11 (a doua metoda op din clasa C) 0.2

//Solutia pentru a rezolva eroarea este sa indepartam valoarea default de la linia 12 pentru variabila x si sa o punem la y, sau sa inversam x si y intre ele in definitia metodei

//0.2

-------------------------------------------------------

**var 4**

#include <iostream>

using namespace std;

class C {

float z;

public:

C() { z = 1.3; }

float op(float x = 2.0, float y = 1.2, float t = 1.5) {

return x + y + z + t;

}

int op(int y) {

return y + z;

}

double op(double y) {

return y + z;

}

};

int main() {

C c;

float i,j, k;

i=c.op();

j=i+c.op(1.2);

k=c.op(2);

return 0;

}

//Compileaza

//Se afiseaza

//6

//8.5

//3

//compileaza si valori rezonabile 0.1

//pentru 6 0.1

//pentru 8.5 0.2

//pentru 3 0.1

-------------------------------------------------------

**var 5**

#include <iostream>

using namespace std;

class C {

float z;

public:

C() { z = 1.3; }

float op(float x = 2.0, float y = 1.2, float t = 1.5) {

return x + y + z + t;

}

float op(float y) {

return y + z;

}

double op(double y) {

return y + z;

}

};

int main() {

C c;

cout << c.op() << "\n";

cout << c.op(1.2) << "\n";

cout << c.op(1) << "\n";

return 0;

}

//Nu compileaza 0.1

//Problema apare la apelul functiilor, mai exact la linia 24, ultimul apel este ambiguu, posibili candidati sunt atat metoda op de la linia 12, cat si metoda op de la linia 15 (penultima si ultima metoda op din clasa C)

//0.2 p daca zice de linia 23 0.1p si atat

//O posibila solutie este sa modificam semnatura metodei de la linia 12 incat sa preia un parametru int y si nu float y

// 0.2p