1. Чи буде помилковим виклик f(1.5, 1.5); для функцій

void f(int, float);

void f(float, int);

Да. Немає відповідность, у нас 2 флота а в фун тільки по 1, буде повідомлення про помилку. Про це сказано в лекції так: якщо не буде встановлена відповідність жодній з перевантажений функцій чи буде встановлена неоднозначна відповідність, на етапі компіляції генерується повідомлення про помилку.

2.Які дії виконують перевантажені оператори в наведеному наприкінці лекції прикладі.

Додавання і віднімання.

3.Наведіть власні приклади перевантаження функцій.

#include <iostream>

**using** **namespace** std;

**float** areaRectangle(**float** a, **float** b);

**float** areaRectangle(**float** a\_m, **float** a\_sm, **float** b\_m, **float** b\_sm);

**int** main()

{

cout << "S1 = " << areaRectangle(32,43) << endl; // вызов перегруженной функции 1

cout << "S2 = " << areaRectangle(4, 43, 2, 12) << endl; // вызов перегруженной функции 2

**return** 0;

}

// перегруженная функция 1

**float** areaRectangle(**float** a, **float** b) //функция, вычисляющая площадь прямоугольника с двумя параметрами a(см) и b(см)

{

**return** a \* b; // умножаем длинны сторон прямоугольника и возвращаем полученное произведени

}

// перегруженная функция 2

**float** areaRectangle(**float** a\_m, **float** a\_sm, **float** b\_m, **float** b\_sm) // функция, вычисляющая площадь прямоугольника с 4-мя параметрами a(м) a(см); b(м) b(cм)

{

**return** (a\_m \* 100 + a\_sm) \* (b\_m \* 100 + b\_sm);

}

4.Наведіть власні приклади перевантаження операторі.

class Integer

{

private:

int value;

public:

Integer(int i): value(i)

{}

const Integer operator+(const Integer& rv) const {

return (value + rv.value);

}

};