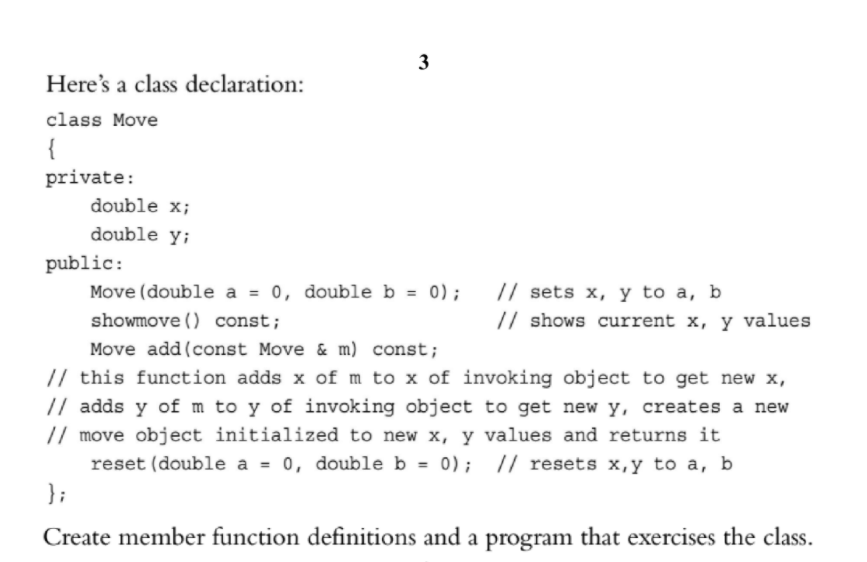
Rudenko Ruslan, SE-TE 2.01

**Task:**

****

**Code:**

#include <iostream>

#include <string>

class Move

{

private:

double x;

double y;

public:

Move(double a = 0, double b = 0); // sets x, y to a, b

void showmove() const; // shows current x, y values

// this function adds x of m to x of invoking object to get new x,

// adds y of m to y of invoking object to get new y, creates a new

// move object initialized to new x, y values and returns it

Move add(const Move& m) const;

void reset(double a = 0, double b = 0); // resets x,y to a, b

};

// sets x, y to a, b

Move::Move(double a, double b) {

this->x = a, this->y = b;

}

// shows current x, y values

void Move::showmove() const {

using std::cout;

using std::endl;

cout << "x: " << this->x << ", y: " << this->y << endl;

}

// this function adds x of m to x of invoking object to get new x,

// adds y of m to y of invoking object to get new y, creates a new

// move object initialized to new x, y values and returns it

Move Move::add(const Move& m) const {

double newX = this->x + m.x;

double newY = this->y + m.y;

return Move(newX, newY);

}

// resets x,y to a, b

void Move::reset(double a, double b) {

this->x = a;

this->y = b;

}

void showmove(std::string, Move);

int main() {

using std::cout;

using std::endl;

using std::cin;

//Set x, y to 0

Move move1;

Move move2(5, 10);

Move move3(15);

showmove("move1 - ", move1);

showmove("move2 - ", move2);

showmove("move3 - ", move3);

Move move4 = move2.add(move3);

showmove("move4 - ", move4);

move4.reset();

showmove("move4 - ", move4);

move4.reset(-5);

showmove("move4 - ", move4);

move4.reset(20, 20);

showmove("move4 - ", move4);

return 0;

}

void showmove(std::string aString, Move move) {

std::cout << aString;

move.showmove();

}

**Results**

