#include "std\_lib\_facilities.h"

void swap\_v(int d1, int d2)

{

int temp = d1;

d1 = d2;

d2 = temp;

}

//this part is executable ,bec int type of variable

void swap\_r(int& d1, int& d2)

{

int temp = d1;

d1 = d2;

d2 = temp;

}

void swap\_c(int& d1, int& d2)

{

int temp = d1;

d1 = d2;

d2 = temp;

}

//this part is const valude, so we can not change the value of the variable where is bounded by const

int main()

{

int x = 7;

int y = 9;

swap\_v(x, y);

swap\_v(7, 9);

/\*its excutable\*/

double dx = 7.7;

double dy = 9.9;

swap\_r(dx, dy);

///\*its not excutable, a reference of type 'int&' cannot initialized with a value of type double.

const int cx = 7;

const int cy = 9;

swap\_c(cx, cy);

//qualifiers dropped in binding reference of type "int& to initializer of type "const int",

//and binded by const value cannot change the value,its already bounded

cout << " x= " << x << " y= " << y << endl;

swap(x, y);

cout << "x=" << dx << " y=" << dy << endl;

swap(x, y);

cout << "x=" << cx << " y=" << cy << endl;

swap(x, y);

return 0;

}

drills 8