mobile.h

```cpp
#pragma once
#include <iostream>
#include <string>

using namespace std;

class Mobile
{
private:
	string model;
	string company;
	int price;
	int cam;
	string color;

public:
	Mobile();
	Mobile(string a, string b, int c, int d, string e);
	void inputMobile();
	void readMobile(string model);
	void addMobile();
	void editMobile();
	void delMobile();

};
```

implementation.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include "mobile.h"

using namespace std;

Mobile::Mobile() : model(""), company(""), price(0), cam(0), color("")
{}

Mobile::Mobile(string a, string b, int c, int d, string e) : model(a), company(b),
price(c), cam(d), color(e)
{}

void Mobile::inputMobile()
{
        cout << "\nEnter model: ";
        getline(cin, model);
        cin.ignore();
        cout << "\nEnter company: ";
        getline(cin, company);
        cin.ignore();
        cout << "\nEnter price: ";
        cin >> price;
        cout << "\nEnter cam: ";
        cin >> cam;
        cin.ignore();
        cout << "\nEnter color: ";
        getline(cin, color);
}

void Mobile::readMobile(string model)
{
        Mobile temp;
        ifstream in;
        in.open("mobiledata.dat", ios::in);

        while (!in.eof())
        {
                in.read((char*)&temp, sizeof(Mobile));

                if (temp.model == model)
                {
                        *this = temp;
                        break;
                }
        }
}

void Mobile::addMobile()
{
        fstream out;
        out.open("mobiledata.dat", ios::binary | ios::app);
```

```
	if (!out)
	{
		cerr << "\nError opening file.";
	}
	else
	{
		Mobile temp;
		temp.inputMobile();
		out.write((char*)&temp, sizeof(Mobile));
	}

	out.close();
}

void Mobile::editMobile()
{
	fstream inNOut("mobiledata.dat", ios::in | ios::out);

	if (!inNOut)
	{
		cerr << "\nError opening file.";
	}
	else
	{
		Mobile temp;
		string mod;
		cout << "\nEnter model you want to edit: ";
		getline(cin, mod);

		while (inNOut.read((char*)&temp, sizeof(Mobile)))
		{
			if (temp.model == mod)
			{
				break;
			}
		}

		cout << "\nEnter new values: ";
		temp.inputMobile();

		inNOut.seekp((int)inNOut.tellg() - (int)sizeof(temp));
		inNOut.write((char*)&temp, sizeof(temp));
	}

	inNOut.close();
}



//void Mobile::delMobile();
```

main.cpp

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include "mobile.h"

using namespace std;

int main()
{
       Mobile h;

       //for (int i = 0; i < 3; ++i)
       //{
       //     h.addMobile();
       //}

       h.editMobile();
}
```