

C++ PROGRAMMING FOR CARTOGRAPHERS AND GEODESIST: TEACHING THE PROGRAMMING LANGUAGE BY GEODESIC EXERCISES AND TOPOGRAPHIC TASKS

Vladimir Zablotskiy

Associated Professor, Cand. Sci., Vladimir Zablotskiy
Moscow State University of Geodesy and Cartography (MIIGAiK), Department of applied cosmonautics and photogrammetry, Gorokhovskiy pereulok 4, Moscow, Russia, 105064
telephone: +7916-380-14-61,
e-mail: V.R.Zablotskii@Yandex.ru

Abstract

A new course of C++ programming for surveyors and cartographers has been developed. Pedagogical experiments were carried out in the period of 2009-2017 in Moscow State University of Geodesy and Cartography. Students have to deal with common geodetic tasks both on lectures and during practical sessions in computer class. This is the main characteristic of the training. A wide range of training programs in the field of geodesy has also been developed. As example the object-oriented program including friend functions and friend classes is considered. The program has two classes - a right-angled triangle and an angle, which further could be declared as a friend class. This program highlights the possibilities of object-oriented programming on the basis of friend relations between classes that have no sibling connection. Experience of using an interactive board and a program for remote administration PC to organize studies in a computer class has been generalized.

Keywords: teaching, higher education, C++ programming, geodetic tasks

INTRODUCTION

The programming is a rapidly growing science and it was included into the courses of higher educational institutions approximately 25 years ago. In Russia, specialists in the field of computational mathematics, who participated in space research and a nuclear project, created the basics of programming training. As a result teaching the programming has been carried out in a framework of computational mathematics for example typical tasks were the calculating of the definite integral, finding the sum of convergent series with taking error in account or evaluating a function of several arguments, etc. Even though this tradition is preserved today not only mathematicians but also different engineers are learn programming. And programming itself is used to solve problems in different fields of science which are not of concern of computational mathematics for example in economy, soil science, and other. Author suggests that in modern context it is possible to break from this tradition and create training courses suitable for a particular higher educational institutions taking into account their scientific specialization. For instance, cartographers and surveyors learning C++ programming are to solve tasks from the related areas of cartography and geodesy. Therefore training course should be geared towards the needs of surveying and cartographic industry.

Nowadays many helpful textbooks and manuals on the programming language C++ exist. Classic multipage book written by B.Stroustrup must be mentioned first (Stroustrup, 1997). An other classic book written by S.Pratt is *C++ Primer Plus* (Prata, 2011). There also exists number of smaller manuals such as H. Schildt book (Schildt, 2003), J. Liberty and M. Cashman's book for self-study (Liberty and Cashman, 2002) or S.R. Danis' book consisted of comics (Davis, 2014). However there are no specific manuals aimed to train surveyors and cartographic engineers. To improve this situation the author has published a range of articles mostly in Russian scientific magazines (Zablotskiy, 2009).

SOME CHARACTERISTICS OF A NEW TRAINING COURSE OF PROGRAMMING

The C++ programming course for undergraduate students has been developed in Moscow State University of Geodesy and Cartography (MIIGAiK). This course is based on geodetic training tasks and intended to meet the needs of surveying industry where the university plays one of the leading roles. The lectures of this course can be illustrated with tasks of general geodesy. Practical lessons at the computer classes and homework are to prepare students to use knowledge of C++ programming methods to solve various engineering and surveying problems. This approach to the course structure has many advantages. Here are some of them.

Firstly, students form an image of programming as a discipline which has practical nature and would be a part of their future profession. Holders of a bachelor's degree and a master's degree active in surveying and Earth's remote sensing acquire some knowledge and practical skills in programming to complete specific engineering tasks. It should also be stressed that via programming a geodetic task a student scrutinizes it, and understands logic of task solution. It would be useful to consider a common geodetic task from the viewpoint of programming. Implementation of the course required improving training workshops in programming, applying new technologies in teaching, and developing new training tasks for homework, tests and exams.

Once C++ programming is conducted via Windows operating system, programming for Windows using rapid application design (RAD) has been included into training process. However, programming for Windows consists of different techniques and methods based on extensive use of RAD. The process of creating the program's interface involves selection of relevant components from the toolbar, their further transferring onto the form, as well as defining properties of these components. All of this requires knowledge about various elements of the programming environment. So the conventional way of teaching by recording the program code on the chalk board seems to be inappropriate. Students are to learn how to operate programming environment, in other words; which element must be chosen to implement a particular function; how to work with a variety of menus and submenus; which options for typical control elements exist, and how to set these options, etc. An interactive electronic board which allows demonstrating the work with programming environment while developing programs has become very convenient tool.

EXPERIENCE IN APPLICATION OF INTERACTIVE ELECTRONIC BOARD

Interactive electronic board is a new technical feature aimed to improve the quality of teaching process. Pedagogical experiments were conducted in MIIGAiK from 2009 till 2015. The interactive board was used while teaching students and post-graduates C++ programming language. The board has been successfully implemented while teaching undergraduate students as well. By this moment we have come up to some important conclusions. Nowadays it appears to be hard to teach C++ programming using variable names, functions and other objects due to some difficulties which arise while organizing a training process. In most cases it is associated with increasing code size. Application of an interactive board technology allows a teacher to concentrate on analyzing issues of programming. Conventional training method suggested usage of white board was not effective and put some extra load on the teacher. Moreover it has become difficult to work with a modular program which is based on a code consisted of more than 500 instructions. Application of such a technology increases the efficiency of the teaching process and allows reducing the load on the teacher significantly. Even using an interactive whiteboard as a display device, improve the efficiency and intensiveness of practicals, makes them interesting for students.

Working on program development and program run, if all the stages are clear to everybody in classroom, encourages collaboration among students. Practical lessons become interesting, students do not get bored and a task is to be solved faster. The high point of a practical lesson is successful program compilation and program run and it creates a specific atmosphere in a classroom where students feel satisfied with learning process and have reached the aims set.

The use of the electronic board significantly reduces the number of students' requests to provide assistance. A teacher does not have to join a student at his computerized workplace to revise the code or find a specific solution of a particular problem. Application of this technology allows a teacher to set a pace of class and to give a complete and clear example how to solve tasks via a program. Interactive electronic board is especially effective in training the development of programs consisted of several modules and contained more than fifty lines of code. Using interactive mode it is possible to find function calls, monitor the transfer of control from one part of the program to another, and observe transitions from a module to other modules quickly and conveniently. Interactive electronic board would make practical programming lessons fun and effective.

EXPERIENCE IN APPLICATION A PROGRAM FOR REMOTE ADMINISTRATION

A program for remote administration PC might be another technical solution for organizing training process at practical lessons of programming. All the computers in computer class are integrated into a local network, which could be used in order to reduce a load put on a teacher and to increase efficiency of training process. It has become possible due to a specific program for remote administration, for example, Remote Manipulator System, set to the computers in class. Such a program was used in 2011-2014 during the C++ programming practicals, IT and program development in a computer center of the MIIGAiK geodetic faculty. With its help students are able to see from their workstations what teacher's computer screen displays. And at the same time teacher might observe the work of each student.

Consider the experience of organizing practical classes on C ++ programming. Practical classes were organized as follows. At the beginning of the lesson, students established a connection using the client-end portion of the remote administration program. Then they opened a window on their computers that displays the desktop of the teacher's

computer. In this mode, all actions that the instructor performs on his computer are immediately being displayed on screens of all students' computers. In this part of the lesson, to observe and memorize the actions of the teacher, who solved a specific problem on his computer, was the main task given to students. Over the second part of the lesson each student was given a task. The task was to be completed using the tutorial prepared by a teacher. At the same time, any student at any time had the opportunity to return to the screen of the teacher's computer and compare their records with the teacher's notes. In the second part of the practical lesson, the teacher, using the program, could control the work of each student.

An important advantage of using the remote administration system is that diligent students are able to make copies of the teacher screen on their computers and store these copies in their file in order to write down the solution of a specific task performed by the teacher in more detailed way. In regard to a teacher the main advantage of using the is that he does not have to skip from one computer to another and work with the mouse, explaining once more what actions were necessary to complete the task or correct his mistakes. Sitting behind the screen of his computer, the teacher can monitor the execution on the educational task by students, since the system can visualize the desktop of each student' on teacher's monitor. In this case, the teacher has an opportunity to detect the error of a particular student, identify typical mistakes made by most of the students and correct them in time. It is possible to organize the educational process in such a way that errors could be corrected right after they have been committed, which will make the learning process more effective. Also, a teacher can apply the control mode of the remote computer, that is, launch / stop a program on it or remotely correct the student's mistake.

In MIIGAiK, the C ++ programming course is taught during one or two first semesters on the sidelines of the general geodesy course. Therefore, problems, arisen in general geodesy, could be successfully used in the field of programming. For practical exercises, two types of programs have been developed. Programs of the first type are small and intended to fix language constructions and features. Such programs could be successfully used both in the lecture part and in practical classes because of their clarity and simplicity. Programs of the second type consist of several modules and files that go through the whole workshop and illustrate the process of real programs development. Such programs are designed to make students work without teacher's supervision. The training process is based on the constant modification of such programs in order to improve its functionality and data protection. The program of the second type is, for instance, THEODOLITE. A run-time version of this program consists of 700 lines and 4 modules.

AN EXAMPLE OF TRAINING PROGRAM

This program illustrates the friend classes and friend functions. It is known that the main method, which makes possible for the program to access particular members of a class, is to use the interface functions. But, sometimes it is necessary to allow one class to access the particular members of another. For this purpose a class is declared as a friend of another class. Then the compiler would allow the friend class to refer to particular elements of the required class. The training program contains two classes: TRIANGLE and its friend class - ANGLE by which the program changes the angles of a triangle.

The program code is considered. Lines 05-13 consist of the definition of the *RightTriangle* class. There it is specially noted that angles of a triangle *angleA*, *angleB*, *angleC* are closed elements. It is necessary to use a key word *friend* in line 10, which is the *public* section of the class definition, to declare the *Angle* class as a friend class to the *RightTriangle* class. Next, in the constructor of the *RightTriangle* class (lines 14-19) the unknown angle of the triangle is calculated, and each angle of the triangle takes the initial value, i.e. 45°, 45° and 90°. In lines 25-34 class *Angle* is defined, which contains a *ChangeTriangle* function which has a particular member *angle* and an access to the triangle via the link. The definition of this function is given in lines 35 to 40. The attention is given to lines 37 and 39 where the closed elements are the angle A and B. *AngleA* and *angleB* convert from a friend class and acquire new values. The main function is defined in lines 42 to 53. In line 44 an object is created. It is a flat right-angled triangle *plainRightTriangle* with angles of 45°, 45° and 90°. In line 46 one more object is created. It is a new angle *newAngle* equal to 60°. Then it calls a *ChangeTriangle* method (line 48) and the address of the *plainRightTriangle* is determined as a parameter, which leads to changes in the values of the acute angles of the triangle.

```
01: #include <iostream>
02: #include <string>
03: using namespace std;
04:
05: class RightTriangle
06: {
07:     public:
```

```

08:         RightTriangle( int angleA );
09:         void ShowAngles( void );
10:         friend class Angle;
11:     private:
12:         int angleA, angleB, angleC;
13: };
14: RightTriangle::RightTriangle( int angleA )
15: {
16:     RightTriangle::angleA = angleA;
17:     RightTriangle::angleB = 90 - angleA;
18:     RightTriangle::angleC = 90;
19: }
20: void RightTriangle::ShowAngles( void )
21: {
22:     cout<<"A = " <<angleA <<"° " <<" B = " <<angleB <<"° " <<" C = "
23:         << angleC << "° " << endl;
24: }
25: class Angle
26: {
27:     public:
28:         Angle(int angle){
29:             Angle::angle = angle;
30:         }
31:         void ChangeTriangle(RightTriangle *);
32:     private:
33:         int angle;
34: };
35: void Angle::ChangeTriangle( RightTriangle * triangle )
36: {
37:     triangle->angleA = angle;
38:     triangle->angleB = 90 - angle;
39: }
40: }
41:
42: int main( void )
43: {
44:     RightTriangle plainRightTriangle( 45 );
45:
46:     Angle newAngle(60);
47:
48:     newAngle.ChangeTriangle( & plainRightTriangle );
49:
50:     cout<<"Angles of Triangle is:"; plainRightTriangle.ShowAngles();
51:
52:     return 0;
53: }
    
```

It is known that it is possible to expand access to members of a class via converting access modifier, for example, replacing *private* to *protected*. The difference between this type of transformation and a transformation with the use of friend classes is emphasized. Protected members of a class allow derived classes to access protected members of the base class directly, using the “dot” operator. The difference is that protected members of a class can refer only to classes that derive from this base class or classes that inherit base class elements. Friend classes in C++ usually are not related by inheritance. Therefore, the only way to access private members of another class for such unrelated classes is to associate those classes as friend classes.

After a class has been defined as a friend class to another class, all function members of a friend class can access private elements of another class. However, to limit the number of user-friendly methods that can access particular data of a class, C++ allows specifying a friend function. To specify a function, the keyword *friend* followed by the function prototype, which, in fact, refers to particular members of a class, is also used. For example, in the *RightTriangle* class, it is possible to specify a friend function *friend void Angle::ChangeTriangle (RightTriangle *)*, opening access to particular data elements of the class triangle. This, however, might result in syntax errors because of wrong order of

class definitions. In our case the use proactive definition of class triangle *class RightTriangle* is needed. Then it would be necessary to define the class angle and the class a right-angled triangle, consequently it would be possible to write the definition of this friend functions.

CONCLUSION

The experience of teaching C++ programming language for students of cartographers and surveyors at the Moscow University of Geodesy and Cartography since 1999 to 2017 has been generalized. Pedagogical experiments have resulted in a new C++ programming course for cartographers and surveyors. The course is based on training geodetic tasks. On lectures and practical lessons an interactive electronic board is effectively used. Another technical feature aimed to increase the effectiveness of training is a remote administration program for computers linked up to a computer network. Using only MS-DOS environment while developing programs is dull for students and therefore this method of programming is not effective. So it is used only at the initial stage of training to illustrate the basic language scheme. A shift to Windows environment is made. An educational object-oriented program of friend functions and friend classes is considered. The program has two classes - a right-angled triangle and an angle, which further could be declared as a friend class. After the program has been launched, the angle, using its relations, changes the sharp angles of a right-angled triangle. This program highlights the possibilities of object-oriented programming on the basis of friend relations between classes that have no sibling connection.

REFERENCES

Davis, S.R. *C++ For Dummies*, John Wiley & Sons, Inc, Hoboken, NJ, 2014.

Liberty, J. and Cashman, M. *SAMS Teach Yourself C++ in 10 Minutes*, SAMS Publishing, Indianapolis, IN, 2002.

Prata, S. *C++ Primer Plus*, Addison-Wesley, Reading, MA, 2011.

Schildt, H. *C++ from the Ground Up*, McGraw-Hill/Osborne, Berkley, CA, 2003.

Stroustrup, B. *The C++ Programming Language. Special Edition*, Addison-Wesley, Reading, MA, 1997.

Zablotskiy, V.R. Learning the C/C++ programming language on the basis of programming geodetic training tasks. *A collection of articles prepared to The international scientific-technical conference, dedicated to the 230th anniversary of MIIGAiK*, issue 2, part 1, Moscow: MIIGAiK, 2009, pp. 199-202.

BIOGRAPHY



Vladimir R. Zablotskiy (b. 1953) graduated from the Lomonosov Moscow State University (soil-science faculty in 1976 and physics faculty in 1985). Candidate of Science (Biology), Associated Professor at Moscow University of Geodesy and Cartography, department of applied cosmonautics and photogrammetry and Associated Professor at Bauman Moscow State Technical University, department of the Physics. Author of more than 50 publications in the field of physics of soils, remote sensing of ecological systems, digital processing of aerospace data and data from geoinformation systems, application of computer technologies in educational process.