International Cartographic Association
Association Cartographique Internationale

Application of Blended Learning in Course of C++ Programming

Moscow State University of Geodesy and Cartography, Russia

Vladimir Zablotskii

Teaching programming in C++

- The CORONAVIRUS/COVID-19 pandemic had a significant impact on the programming training of cartographers and surveyors at the Moscow University of Geodesy and Cartography.

- In 2021 the educational process at the Moscow University was reorganized as blended learning.

- As for programming in C++, blended learning allows the teacher to conduct the main educational process by e-learning, but to check the outcomes in the form of a classic exam in the classroom.

- Blended learning combining the advantages of both conventional and remote teaching methods is suitable for a wide range of academic disciplines for example computer science associated with geodesy and cartography.

Teaching programming in C++

- A new approach to teaching C++ programming on the base of solving training geodetic tasks has been developed. In lectures of this course the topics of C++ programming language are study by general geodesy problems.

- Lectures, practicals and homework encourage students to use knowledge of programming to solve various cartographic and geodetic problems.

- A large set of geodetic and cartographic training programs covering the main topics of C++ programming has been developed. The programs are designed to be executed under the MS-DOS environment because this environment is the most convenient environment for initial learning the basic of programming.

Actual problems of blended learning

- In the educational process, three main components of distance learning are implemented. They are distance lectures, distance practicals, and distance examination or assessments of the students' knowledge.

- Distance lectures and practicals were less affected than the distance exam. Indeed, it is currently difficult to determine which way of teaching is more effective in classrooms or online. The same is true in the case of practical. Obviously, there is no need to complete the programming task having the entire group of students in the classroom. It's quite possible to work on it at home.

- The **assessment of the knowledge**, **conducting examinations** and **tests** remains a complicated process. This is probably the most problematic part of the learning process in terms of a pandemic.

Use of IT in the learning process

- The role of video lectures in the educational process has significantly increased, and now this is an indisputable fact. Many teachers prepared their lectures in video format. For this purpose, programs such as Zoom, Mind, MOODLE, and others were used.

- The main advantage of a video lecture is the ability to stop the lecture, "pause" it to analyse some complex issues, think it over, and examine in detail what the lecturer said. It is also possible to re-listen and watch the video lecture at the end of the training course to remind the material before passing the exam.

- Computer testing is well known and should be an alternative to exam tickets. The testing program sets an intense rhythm for the exam, so there is no time to look for answers on the Internet. Lack of time for searching for the answers on the Internet is the main advantage of online examination tests.

Examples of Training C++ programs

Title of the program	*Goal of the program*
Welcome to Moscow University of Geodesy and Cartography	introduction to the process of creation of a code of a program, compilation, and debugging
Number Pi	output to the console of the number *pi* with a different number of significant digits is studied
Average convergence of meridians	instructions of console output for different symbols and their composition are studied
The Radius of Earth's orbit	an error of memory overflow is studied
Height of a point on the map	types of variables and arithmetic operators are studied
Column and zone number	ternary operator (*?:*) is studied
Collimation error	*if-else* instructions in full and short form for computing the angular error of a theodolite's telescope are studied

Examples of Training C++ programs

Title of the program	*Goal of the program*
Rhumb of the line (reduced bearing)	"logical chain" formed by nested *if* instructions are studied
The denominator of map scale	*switch* instruction on the example of computing the horizontal equivalent is studied
Converting an angle from radian to degrees	the instruction of a *for* loop is studied
Rectangular coordinates	the infinite *while* loop on the example of calculating the coordinates of points is studied
Gaussian convergence of meridians	convergence of meridians using a *while* loop for a point given by the latitude and longitude is studied
The slope of the line	instructions of the *do–while* loop on the example of calculating the slope of a line on a topographic map are studied
Calculating the angle	declaration and calling the *simple function* which calculating degrees, minutes and seconds are studied

Examples of training programs

The following examples of computer training programs will be discussed in more detail :

- The program **Simple arithmetic calculations** illustrate several computational examples taken from the theory of errors in geodetic measurements,
- The program **Earth's curvature** illustrates the influence of the Earth curvature on the absolute measurement errors of horizontal and vertical line lengths,
- The program **Calculating the Bearing of Line** illustrates nested branch operators and a logical ladder,
- The program **Eratosthenes' radius** of the Earth illustrates the method of the ancient Greek scientist Eratosthenes who was the first to calculate the radius of the Earth.

Simple arithmetic calculations

Example 1. True errors of ten equally accurate measurements of a certain known horizontal angle on the ground are –5″, +1″, +2″, –3″, –2″, +4″, –4″, 0″, –3″, +4″. It is required to calculate the mean error of the angular measurement data.

Example 2. The same line was measured on the ground by 4 teams of students. The first team measured the line 4 times, the second team measured it 7 times, the third team performed measurement 5 times, and the fourth team measured the line 3 times. The results of measurements are shown in Table 1. It is required to calculate the general mean or the weighted average of the line length.

Team number	1	2	3	4
Line length, m	127.456	127.450	127.461	127.443

Table 1. Results of measurements line length

Example 3. The angle on the ground was measured with the same tool twice. The first measurement was completed in four steps, the average equals 45°01'00". The second time the measurement was carried out in six steps, the average equals 44 ° 59'30 ". It is necessary to calculate the most reliable value of the angle.

Program code

```
01: #include <iostream>
02:  #include <iomanip>
03:  using namespace std;
04:
05:  int main(void)
06:  {
07:   cout  << " Mean error of angle measurements: "<<"-" << (5 + 1 + 2 + 3 + 2 + 4 + 4 + 3 + 4)/10.0;
07:          << "\""  <<"..." << "+" <<(5 + 1 + 2 + 3 + 2 + 4 + 4 + 3 + 4)/10.0   << "\""  << endl;
08:   cout << " Weighted average of the line length: " << setprecision(6) << (127.456*4 + 127.450*7 +          \
08:              127.461*5 + 127.443*3) /(4 + 7 + 5 + 3)  << " m" << endl;
09:   cout.fill('0');
10:   cout << " The most reliable value of angle measurements :"<< ((45*3600+1*60+0)*4 + (44* 3600 +          \
10:             59*60+30)*6)/(4+6)  /3600 << char(248);
11:   cout << setw(2) << ((45*3600+1*60+0)*4+(44*3600+59*60+30)*6)  /(4+6)/60%60  << "\'";
12:   cout << setw(2) << ((45*3600+1*60+0)*4+(44*3600+59*60+30)*6) /(4+6)%60  << "\""  << endl;
13:   return 0;
14:   }
```

The program and these exercises can be used in blended learning mode to start familiarizing students with C++ programming. The program illustrates the use of basic arithmetic operators. Students learn the theory of errors at the beginning of a general surveying course and are ready for such an example in C++ programming

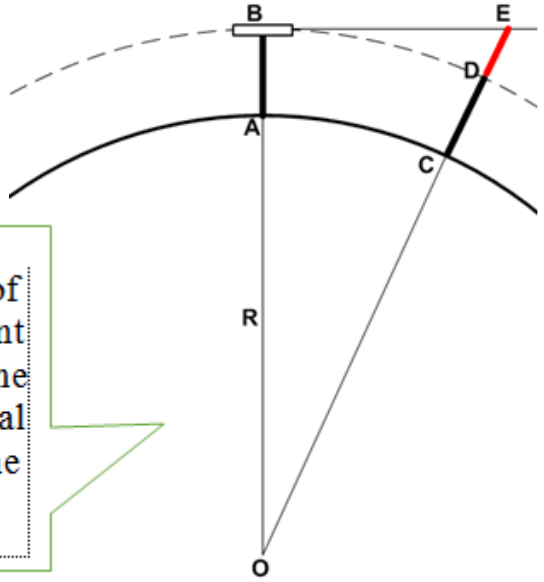
Earth's curvature and measurements of lengths

The influence of the Earth curvature on the absolute measurement errors of horizontal and vertical line lengths is well known for surveyors and determined using the following formulae:

$$\Delta d \cong \frac{d^3}{3R^2}$$

$$\Delta h \cong \frac{d^2}{2R}$$

O is Earth's center, R is the radius of the Earth, AB is a geodetic instrument for determining length and height of the lines, BE – AC is an error of horizontal line and DE is an error of vertical line due to curvature of the Earth.



where Δd the error in determining horizontal length , Δh the error in determining the vertical length and d the horizontal distance for which this error is calculated, R- the average radius of the Earth (6371.11 km).

Program code

```
01: #include <iostream>
02: #include <iomanip>
03: using namespace std;
04:
05: int main(void)
06: {
07:  const double radiusOfEarth = 6371.11;
08:  double absoluteErrorOfDistance, relativeErrorOfDistance, m;
09:  double absoluteErrorOfHeight, scale;
10:  int d;
11:
12:  cout <<"----------------------------------------------"<< endl;
13:  cout <<"  d, km   AbsoluteError d, m   RelativeError d   "<< endl;
14:  cout <<"----------------------------------------------"<< endl;
15:
16:  for( d = 1; d < 26 ; d += 1)
17:  {
18:   absoluteErrorOfDistance =
18:              (d*d*d)*1e+3/(3*radiusOfEarth*radiusOfEarth);
19:   relativeErrorOfDistance = absoluteErrorOfDistance/(1000*d);
20:   scale = 1/relativeErrorOfDistance;
22:   cout << setw(5) << d  << setw(15)  << setprecision(2)
22:         << setiosflags(ios::showpoint)<< absoluteErrorOfDistance;
23:   cout << setw(15)  <<"1:" <<  setiosflags(ios::scientific)
23:         << setprecision(1) << scale << resetiosflags(ios::scientific)
23:         << endl;
24:  }
25:
26:   cout <<"----------------------------------------------"<< endl;
27:   cout <<"  d, m         AbsoluteError H, cm          "<< endl;
28:   cout <<"----------------------------------------------"<< endl;
29:
30:  for( d = 100; d < 2100 ; d += 100)
31:  {
32:   absoluteErrorOfHeight = (d*d)/(2*radiusOfEarth*10);
33:
34:   cout << setw(5)  <<  d  << setw(25)  << setprecision(3)
34:         << setiosflags(ios::showpoint) << absoluteErrorOfHeight 34:
<< endl;
35:  }
36: return 0;
37:}
```

Using the program, the typical dimensions of surface areas were obtained, which can be conditionally considered flat with an accuracy of 1: 1000000 and 1: 200000.

Earth's curvature and measurements of lengths

Table 1. Errors in measuring horizontal line length

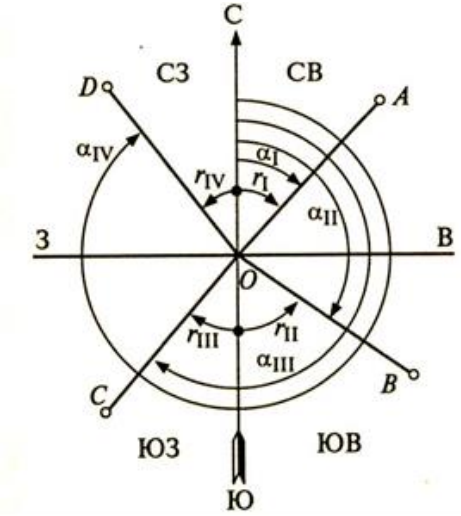
d, km	AbsoluteError d, m	RelativeError d
1	0.0000082	1 : 120000000
2	0.000066	1 : 30000000
3	0.00022	1 : 14000000
4	0.00053	1 : 7600000
5	0.0010	1 : 4900000
…	…	…
10	0.0082	1 : 1200000
…	…	…
15	0.028	1 : 540000
…	…	…
20	0.066	1 : 300000
…	…	…
25	0.13	1 : 190000

d, m	AbsoluteError H, cm
100	0.0785
200	0.314
300	0.706
400	1.26
500	1.96
…	…
1000	7.85
…	…
1500	17.7
…	…
2000	31.4

Table 2. Errors in measuring vertical line length

Calculating the line bearing

The program demonstrates the usage of the *if-else* construction to calculate the bearing of the line (the known value is the grid angle of the given direction).



Algorithm for calculating line bearing is:

- If the angle A_1 lies between 0° - 90°, the bearing is calculated as $r_1 = A_1$,
- If the angle A_2 lies between 90° - 180°, the bearing is calculated as $r_2 = 180° - A_2$,
- If the angle A_3 lies between 180° - 270°, the bearing is calculated as $r_3 = A_3 - 180°$,
- If the angle A_4 lies between 270° - 360°, the bearing is calculated as $r_4 = 360° - A_4$,
- Otherwise, an error message is output to the user.

The algorithm is implemented using nested *else-if* statements forming a logical ladder of 6 chain link.

Program code


```
01: #include <iostream>
02: using namespace std;
03:
04: int main (void)
05: {
06:  double degrees, minutes, gridAzimuth, rhumbLine;
07:
08:  cout <<" Enter the grid azimuth of the direction: "
08:       <<" (degrees, whitespace, minutes): ";
09:  cin >> degrees >>  minutes;
10:  gridAzimuth =  degrees + minutes/60;
11:
12:  if((gridAzimuth == 0) || (gridAzimuth == 360))
13:  {
14:   rhumbLine = 0;
15:   cout <<" Rhumb of the direction is N: ";
16:  }
17:  else if((gridAzimuth > 0)&&(gridAzimuth < 90))
18:  {
19:  rhumbLine = gridAzimuth;
20:  cout <<" Rhumb of the direction is NE: ";
21: }
22: else if (gridAzimuth == 90)
23: {
24:  rhumbLine = 90;
25:  cout <<" Rhumb of the direction is E: ";
26: }
27: else if((gridAzimuth > 90)&&(gridAzimuth < 180))
28: {
29:  rhumbLine = 180 - gridAzimuth;
30:  cout  <<" Rhumb of the direction is SE: ";
31: }
32: else if(gridAzimuth == 180)
33: {
34:  rhumbLine = 0;
35:  cout  <<" Rhumb of the direction is S: ";
36: }
37: else if((gridAzimuth > 180)&&(gridAzimuth < 270))
38: {
39:  rhumbLine = gridAzimuth - 180;
40:  cout  <<" Rhumb of the direction is SW: ";
41: }
42: else if(gridAzimuth == 270)
43: {
44:  rhumbLine = 90;
45:  cout <<" Rhumb of the direction is W: ";
46: }
47: else if((gridAzimuth > 270)&&(gridAzimuth < 360))
48: {
49:  rhumbLine = 360 - gridAzimuth;
50:  cout  <<" Rhumb of the direction is NW: ";
51: }
52: else  {
53:   cout <<" The grid azimuth should be in interval 0°-360° "
53:        <<" The angle entered: " << gridAzimuth << endl;
54:   return  -1;  }
55:  degrees = (int)rhumbLine;
56:  minutes = (rhumbLine - degrees)*60;
57:  cout << degrees <<"° " << minutes <<"'" << endl;
58:  return 0;
59: }
```

How Eratosthenes calculated the radius of the Earth

The ancient Greek scientist Eratosthenes, who lived in the city of Alexandria around 240 BC, was the first to calculate the radius of the Earth.

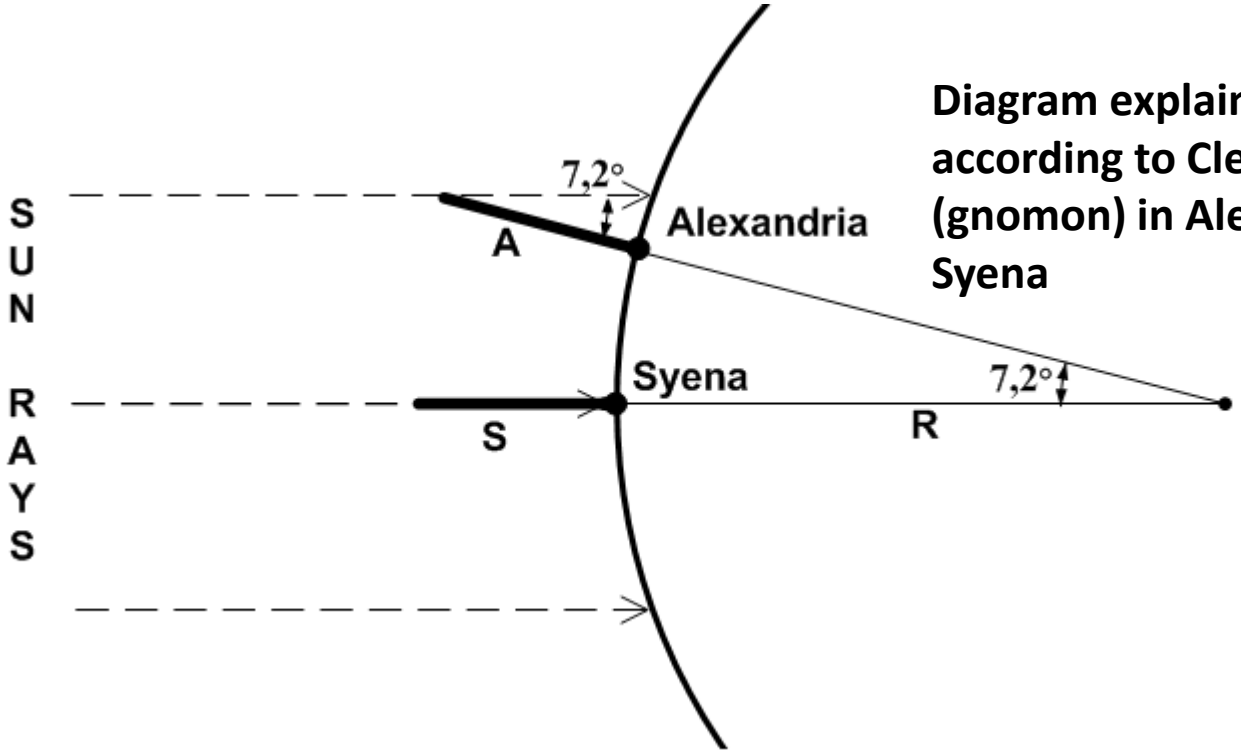


Diagram explaining the Eratosthenes' method according to Cleomedes: A - vertical rod (gnomon) in Alexandria, S - vertical rod in Syena

The scientist measured the angle between the Sun at Zenith in Alexandria on the summer solstice day when, objects do not cast a shadow, and the result obtained was approximately 7.2°.

Program code

```
01: #include <iostream>
02: #include <iomanip>
03: using namespace std;
04:
05: int main(void)
06: {
07:  float angle = 7.2;    // degree
08:  float distanceFromAlexandriaToSyene = 5000;   // stadia
09:  float EarthCircumference;
10:  EarthCircumference = 360/angle *
distanceFromAlexandriaToSyene;
11:  float radiusOfEarthInStadia = EarthCircumference/(2 * M_PI);
12:  cout <<"Eratosthenes' problem" << endl;
13:  cout <<"Earth radius in stadia: "<< setiosflags(ios::fixed)
13:       << setprecision(0)<< radiusOfEarthInStadia  << endl;
14:
15:  float oneStadium = 157.5;  // Egyptian stadium in meters
16:  float radiusOfEarth = radiusOfEarthInStadia * oneStadium/1000;
17:
18:  cout <<"Earth radius in km: " << radiusOfEarth  << endl;
19:
20:  float absoluteError, relativeError;
21:  absoluteError =  6371 - radiusOfEarth;
22:  relativeError = absoluteError/6371 *100;
23:
24:  cout <<"Absolute error, km: " << absoluteError  << endl;
25:  cout <<"Relative error, %:  " <<relativeError << endl;
26:  return 0;
27:  }
```

Eratosthenes' problem	
Earth radius in stadia:	39789
Earth radius in km:	6267
Absolute error, km:	104
Relative error, %:	2

As can be seen from the result, the error in the determination of the radius of the Earth by the Eratosthenes method is insignificant even from the point of view of modern times.

Conclusion

- Effective blended learning is based on a carefully designed and tailored learning process. A new approach to teaching students of surveyors and cartographers programming in C++ has been developed. The main characteristic of this method of teaching is that students in the course deal with problems of geodetic content both in lectures and in practical classes.

- There are written about seventy training programs on different topics of C++ programming course. Programs are illustrating all major language constructs. The code of the training computer program is written in a modern programming style and is read and understood easily.

- Modern internet technologies make it possible to bring the educational process to students' homes, not to lock them in classrooms and laboratories. The following IT technologies are used in blended learning:
  - ❖ programs such as Zoom, Mind, MOODLE, and others were used;
  - ❖ global computer network Internet.

Literature

- *Kiselev M.I., Mikhelev D.Sh.* Geodesy. Textbook. M, “Academy “, 2008.
- *Klyushin E.B., Kiselev M.I., Mikhelev D.Sh., Feldman V.D.* Engineering geodesy, Textbook for universities, M, “Academy “, 2004.
- *Davis S.R.* C++ For Dummies, 7th Edition, Published 2014 by John Wiley & Sons, Inc.
- *Liberty J., Cashman M.* SAMS Teach Yourself C++ in 10 Minutes, 2nd Edition, Published 2002 by SAMS.
- *Kernighan B., Ritchie.D.* The C programming language, AT&T Bell Laboratories, Murray Hill, New Jersey, 1988.
- *Schildt, H.* C++ from the Ground Up, McGraw-Hill/Osborne, Berkley, CA, 2003.

Thank you for your attention