

## **C++ PROGRAMMING LANGUAGE AND ENVIRONMENT, PROGRAM INTERFACE, PROGRAM STRUCTURE, PROGRAM ELEMENTS**

**Nodira Sharifovna Khodjaeva**

Tashkent State Transport University

**Annotation:** Object Oriented Programming (OBP) is a way of organizing programs. C++ is an Object Oriented Programming language. It was created by Bjarne Stroustrup in 1983 at the AT and T Laboratory in New Jersey, USA. He felt that C was not rich in images and added additional symbols and extended it using his favorite language, Simula 67. Simula 67 was one of the earliest object-oriented languages.

**Keywords:** Program, image, calculation, laboratory, ancient, object, level, document.

Bjarne Stroustrup actually called it "C oily with degrees". More precisely, C++ is a superset of C: almost everything that is true in C is also true in C++, even if the reverse is true. C++ (C plus plus) was named by Rick Mascitti. Here "++" is C's expansion operator. Version 1.0 was released in 1985, version 2 in 1989, and version 3 in 1992. C++ was developed to address problems encountered by users and discussed at AT&T. The excellence of C++ is confirmed by the following two events:

- (i) The C++ Committee at AMSI (American National Standards Institute) and
- (ii) Annotated C++ manuscripts published by Ellis and Stroustrup.

ANS/ISO published the final C++ standards document in 2003.

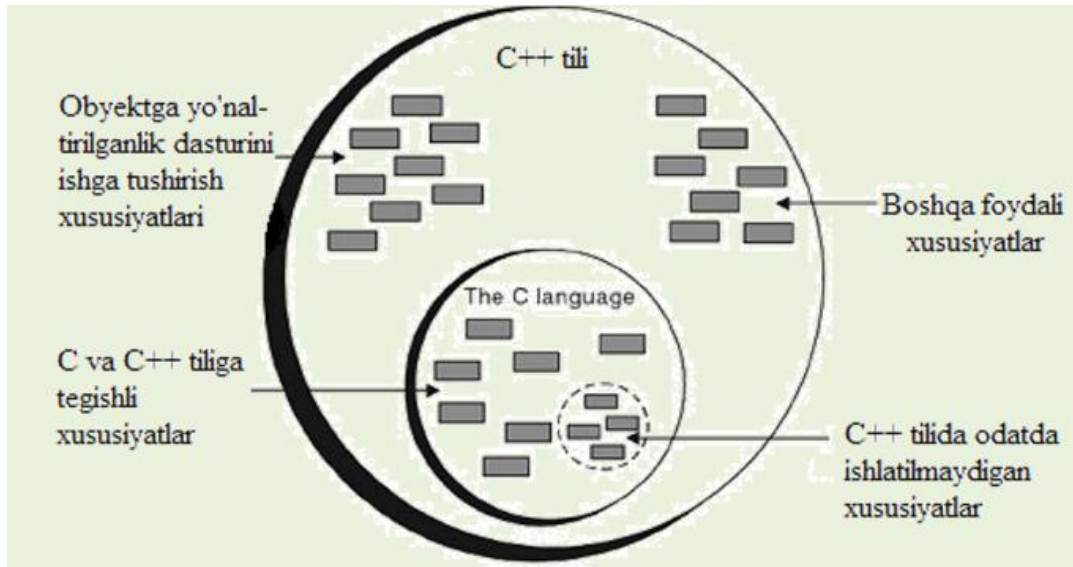
C++ has the following features:

- (i) Reduces the complexity of their problems while solving them.
- (ii) Guarantees accuracy of results.
- (iii) Affordable in terms of computer equipment and other resources.
- (iv) Existing software facilities are cheap and easy to integrate for vacuum cleaners.
- (v) Compact i.e. can be used on different computers without making any changes to the program.

An object-oriented program is a collection of information about discrete objects, that is, structures and functions that interact with other objects.

C++ **classes have functional overloads and operator overloads** adds. With these, we can create abstract data types; learn properties from existing data types, and use polymorphism.

Therefore, C++ language is an advanced level of C that extends the facilities of object orientation. In addition, C++ has other new features, namely: improvements in I/O and new possibilities for writing comments.



**Figure 1. C and C++ the connection between**

In general, the practical differences between C and C++ are greater than you might think. You can write a program in C++ just like a program written in C. C++ programmers prefer C++ not only for its new features, but they also note that C's typical features are presented differently than C programmers.

## Net programming

Programmers are the kind of people who always seem to be missing something, because they are always looking for ways to improve the efficiency and usability of the program. In some cases, we need more tools when working, that is, mainly when working with programming languages. Although such languages make up the majority, some of them are considered strong. The effectiveness of the language depends on its speed and flexibility. The syntax of the language should be concise and understandable. It should be able to build correct code and be flexible, but not old-fashioned (and dead-end) solutions. In short, a good language should have a quality that is not found in a material, that is, the person working on it should enjoy it. One such language is S#.

NET Framework, the Microsoft-developed S# language draws on a rich programming heritage. Its chief architect is Anders Hejlsberg, a leading specialist in this field.

S# is a descendant of two of the world's most successful computer languages, S and C++. It inherits syntax, keywords and operators from the C language. It has the ability to build and improve the object model defined in C++. Also, S# is closely related to another successful Java language. Since S# and Java have a common origin, but they differ in many important aspects, they can be called "siblings". For example, they both support a common programming framework and use intermediate code to work well, but differ in implementation details.

Building on a powerful foundation of legacy features, S# has significant innovations that take the art of programming to new heights. For example, the elements of the S# language include concepts such as delegates (implemented), properties, indexers, and events. Also, the syntax supporting targets (attributes) has been added; Solve problems with COM (Component Object Model - Microsoft's component object model - a standard mechanism that interfaces using an object that exposes its services to other objects) the construction of components is simplified at the expense of solidification. Also, like Java, S# provides tools for dynamic error detection, security, and program control. However, the difference between S# and Java is that it allows programmers to refer to pointers. Thus, S# combines the original power of C++ with the safety of Java, which uses the correct template classes

and has a type checking mechanism. Also, the trade-off between usability and reliability in S# has been moderated and eliminated (unknown to the user or the program).

The evolution of programming languages as computing advances changes the computing environment, the way programmers think, and the way they approach programming. C# is no exception. S# is currently at the forefront of improving usability and innovation. It is a language that any professional programmer should consider.

## Programs

The word program is used both as a word defining a separate block of commands (given code) and as a word defining an executable software product as a whole.

This duality can confuse the reader. Therefore, we clarify it. So we understand a program as a set of commands written by a programmer, or a computer product that performs actions.

### Issues faced by developers

Over time, the issues faced by programmers are changing. Twenty years ago, programs were created to process large amounts of data. In this case, both the writer of the program and its user were required to be professionals in the field of computer knowledge. Many changes have taken place now. Computers are mostly used by people who do not have an understanding of its hardware and software. The computer is not a tool for people to study it in depth, but rather a tool for solving problems that are put in front of them, related to their work left.<sup>1</sup>

As this new generation of software is made easier for users to work with, the complexity of the software itself increases. Modern programs should have an interface consisting of a large number of windows, menus, dialog boxes and visual graphic environments that are highly user-friendly.

The changing demand for programming has led not only to a change in languages, but also to a change in the technology of writing it. Although there are many stages in the history of programming evolution, in this course we will look at the transition from procedural programming to object-oriented programming.

## Procedural, structural, and object-oriented programming

Until then, programs consisted of a sequence of procedures that performed an action on the given data. A procedure or function also consists of a defined set of sequentially executed commands. In this case, references to the given information are carried out separately in procedures.

The basic idea of structured programming is completely consistent with the principle of "divide and rule". We consider a computer program as a set of problems. We divide an arbitrary problem that is too complex for simple description into several relatively smaller structural problems, and continue to divide until the problems are simple enough to understand.

**The C++ programming language is based on the C language.** C, in turn, is derived from B and BCPL. BCPL was created by Martin Richards in 1967 and was intended for writing operating systems.

In 1983, as the C language became more common, efforts to standardize it began. For this, the X3J11 technical committee was formed under the American National Standards Committee (ANSI). And in 1989, this standard was adopted. In 1990, ANSI and the International Organization for Standardization (ISO) jointly adopted ANSI/ISO 9899:1990 for C in order to spread the standard around the world. Because of this, programs written in C can run on most computer platforms with little or no changes. C++ was developed in the early 1980s by Bjarne Stroustrup, based on C. C++ has many extensions,

---

<sup>1</sup>J.B.Dixit. Fundamental soft computer programming and Information texnology. India. 2009 (192-193betlar)

but the main one is that it allows programming with objects. Fast and high-quality writing of programs is of great importance nowadays. To ensure this, the idea of object-oriented programming was put forward. Like structured programming in the early 1970s, building programs through objects that model real-life objects revolutionized programming. In addition to C++, other multi-objective programming languages appeared. The most prominent of these is the Smalltalk programming language developed at Xerox's Palo Alto Research and Development Center (PARC). Everything in Smalltalk is object-oriented. And C++ is a hybrid language. It can be structured programming similar to C or a new one, object-oriented programming. What we say is new is relative. The philosophy of object-oriented programming has been around for over twenty years. C++ has a very rich library of functions and objects. That is, learning to program in C++ is divided into two parts. The first is to learn C++ itself, and the second is to learn how to use the ready-made objects/functions in the C++ standard box.

### Basic terminology

Let's take a look at the basic terminology of C++.

#### A set of features

A feature set is a set of important symbols that a language can understand. An attribute can represent any letter, single-digit number, or other character. C++ has two feature sets. They are:

(i) *Data Features*

(ii) *Transition Features / Termination features.*<sup>2</sup>

(i) *Data Features*

Information text is created through information properties. They are:

**Letters** A to Z, a to z and - (underscore)

**Single digit numbers** 0 to 9

**Special characters** Space + - \* 1 A- % = ! & ( ) { | I I ? " ; : \ # ' < > . <= >= @

(ii) **Transition Features / Termination Features**

*These properties are used in time-related processes.*

C++ also has non-displayable or non-typeable symbols such as; line feed, form feed, tab etc.

This set of characters is considered a single character even if it is followed by a backslash ( \ ). appears as follows.

For example:

' \ n ' Newline ` \ b Backspace \\ Backslash

\ t Horizontal tab '\ f ' Form feed \ ? Question mark

' \ a 1 Bell (beep) '\ r ` Carriage return \ N Octal constant (where N is an octal constant)

' \ 0 ' Null I " ' Double quote \ xN Hexadecimal

constant (where N is a hexadecimal constant)

' \ v ' Vertical tab '\ I I Single quote

---

<sup>2</sup>J.B.Dixit. Fundamental soft computer programming and Information texnology. India. 2009 (198-200 betlar)

### Symbols or logical parts

The smallest single characters in a program are called symbols or logical units.

C++ has the following features.

1. *keywords*
2. *Determiners*
3. *Constants (literals)*
4. *Punctuation marks (separators)*
5. *Operators*

All C++ programs are written using symbols, white boxes, and the syntax (grammar) of the language. Most C++ characters are similar to C characters, with only some additions and minor changes.

### Literature

1. Islomovna M. F. et al. DESIGNING THE METHODOLOGICAL SYSTEM OF THE TEACHING PROCESS OF COMPUTER GRAPHICS FOR THE SPECIALTY OF ENGINEER-BUILDER //Journal of Contemporary Issues in Business & Government. – 2021. – Т. 27. – №. 4.
2. Khodjaeva N. S. METHOD OF TEACHING DIFFERENTIATED ISSUES. – 2021.
3. KHODJAEVA N. S., YUNUSOVA N. S. SPIRITUAL AND MORAL EDUCATION OF YOUTH IN MAKHALLA //THEORETICAL & APPLIED SCIENCE Учредители: Теоретическая и прикладная наука. – 2021. – №. 12. – С. 685-688.
4. Sharifovna Y. N., Sharipovna K. N. ESTS Credit System in Education //CENTRAL ASIAN JOURNAL OF THEORETICAL & APPLIED SCIENCES. – 2021. – Т. 2. – №. 11. – С. 32-35.
5. Olimov, S. S., & Mamurova, D. I. (2021). Graphic Information Processing Technology and its Importance. *European Journal of Life Safety and Stability (2660-9630)*, 10, 1-4.
6. Khodjayeva N. S., Mamurova D. I., Nafisa A. IMPORTANCE IN PEDAGOGICAL TECHNIQUES AND EDUCATIONAL ACTIVITY //International Engineering Journal For Research & Development. – 2020. – Т. 5. – №. CONGRESS. – С. 5-5.
7. Olimov, S. S., & Mamurova, D. I. (2021). Graphic Information Processing Technology and its Importance. *European Journal of Life Safety and Stability (2660-9630)*, 10, 1-4.
8. Olimov, S. S., & Mamurova, D. I. (2022). Information Technology in Education. *Pioneer: Journal of Advanced Research and Scientific Progress*, 1(1), 17-22.
9. Toshtemirovna T. N. et al. ZAMONAVIY DARSLARNI TASHKIL ETISHDA O'QITUVCHINING PSIXOLOGIK XUSUSIYATLARI //Научно-практическая конференция. – 2021.
10. Toshtemirovna T. N., Mahmudovna I. A. ZAMONAVIY DARSLARNI TASHKIL ETISHDA TEXNIK VOSITALARDAN FOYDALANISH IMKONIYATLARI //Научно-практическая конференция. – 2021.
11. Olimov, S. (2020). The differentiation of education is an important factor of pedagogical technology. *European Journal of Research and Reflection in Educational Sciences*, 8(11).
12. Олимов, Ш. Ш. (2015). Маънавий-ахлоқий тарбия асослари.(Монография). Т.:“Fan va texnologiya”,-2015, 228.

13. Shavkatovich A. A., Sharifovna X. N. DEVELOPMENT OF DESIGN SKILLS OF HIGH SCHOOL STUDENTS //International Engineering Journal For Research & Development. – 2020. – Т. 5. – №. 7. – С. 5-5.
14. Mamurova, F. I., & Mustafoev, E. (2021, October). Aksonometrik Proyeksiyalarning Asosiy Teoremasi. Dimmetrik Aksonometriya Qurish. In " *ONLINE-CONFERENCES*" PLATFORM (pp. 100-103).
15. Mamurova, F. I., & ugli Mustafayev, E. I. (2021). SHADOWS IN A PERSPECTIVE BUILDING. *Conferencious Online*, 16-18.
16. Mamurova, F. I., & oglu Akmalov, J. O. (2021). ORGANIZATION OF GEODESIC WORK. STATE GEODESIC NETWORKS. *Conferencious Online*, 21-23.
17. Sharipov S. THE MAIN ASPECTS OF THE DEVELOPMENT AND HISTORICAL AND CHRONOLOGICAL FEATURES OF TRANSLATED LEXICOGRAPHY IN UZBEKISTAN //ЦЕНТР НАУЧНЫХ ПУБЛИКАЦИЙ (buxdu. uz). – 2022. – Т. 15. – №. 15.
18. Sharipov S. BILINGUALISM-AS A PHENOMENON OF BILINGUAL COGNITIVE COMMUNICATION //ЦЕНТР НАУЧНЫХ ПУБЛИКАЦИЙ (buxdu. uz). – 2022. – Т. 15. – №. 15.
19. Sharipov S. ЛЕКСИКОГРАФИЯ (ТАРЖИМА ЛЕКСИКОГРАФИЯСИ) РИВОЖЛАНИШИНИНГ АСОСИЙ ЖИҲАТЛАРИ //ЦЕНТР НАУЧНЫХ ПУБЛИКАЦИЙ (buxdu. uz). – 2022. – Т. 15. – №. 15.
20. Mamurova, F. T., Abdullayeva, N. K., & Mallaboyev, N. (2019). USING THE «ASSESSMENT» METHOD IN ASSESSING STUDENTS KNOWLEDGE. *Theoretical & Applied Science*, (11), 80-83.
21. Mamurova, F. I. (2021, May). ARTIST OF UZBEKISTAN MAKSUD SHEIKHZADE. In *E-Conference Globe* (pp. 176-178).
22. Mamurova, F. I. (2021). Factors for Forming the Professional Competence of Building Engineers in the Context of Information Education. *EFFLATOUNIA-Multidisciplinary Journal*, 5(2).
23. Mamurova F. I., oglu Amirkulov A. F. COORDINATE AND HEIGHT SYSTEMS USED IN GEODESY //Conferencious Online. – 2021. – С. 19-20.
24. MAMUROVA F. I. FACTORS OF FORMATION OF PROFESSIONAL COMPETENCE IN THE CONTEXT OF INFORMATION EDUCATION //THEORETICAL & APPLIED SCIENCE Учредители: Теоретическая и прикладная наука. – 2021. – №. 9. – С. 538-541.
25. Islamovna, M. F. I. F., & Akhmadzhonovich, M. B. (2021, January). ARCHITECTURAL DESIGN TECHNIQUE. In *Euro-Asia Conferences* (Vol. 1, No. 1, pp. 310-312).
26. Мамурова Ф. И. ЭФФЕКТИВНОСТЬ ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНОЙ КОМПЕТЕНТНОСТИ БУДУЩИХ ИНЖЕНЕРОВ-СТРОИТЕЛЕЙ В СОВРЕМЕННЫХ УСЛОВИЯХ //Наука и образование сегодня. – 2021. – №. 4 (63). – С. 92-93.
27. Мамурова Ф. И. РОЛЬ ЗНАЧЕНИЕ И ПРИМЕНЕНИЕ ИНЖЕНЕРНЫХ КОММУНИКАЦИОННЫХ СИСТЕМ В ЗДАНИЯХ И КОНСТРУКЦИЯХ //В научный сборник вошли научные работы, посвященные широкому кругу современных проблем науки и образования, вопросов образовательных технологий 2020.-436 с. – 2020. – С. 414.