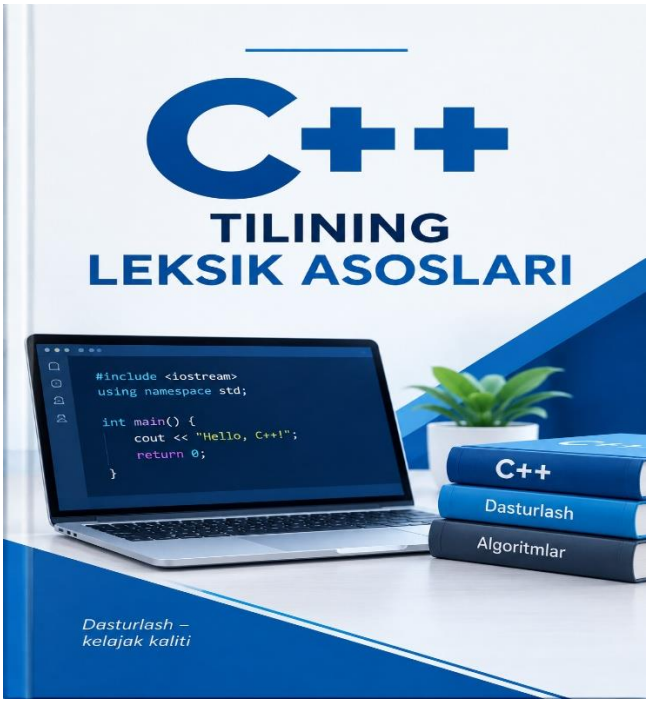
O‘ZBEKISTON RESPUBLIKASI
OLIY TA’LIM, FAN VA INNOVATSIYALAR VAZIRLIGI

GULISTON DAVLAT UNIVERSITETI

AXBOROT TEXNOLOGIYALARI KAFEDRASI

C++ TILINING LEKSIK ASOSLARI



Guliston-2026

Negmatulloyev Z.T. C++ tilining leksik asoslari- Guliston, 2026.- 20. s.

Ushbu ma’ruza matni “C++ tilining leksik asoslari” mavzusiga bag‘ishlangan. Unda C++ dasturlash tilining alifbosi, belgilari va asosiy leksik birliklari haqida ma’lumot berilgan. Dastur tuzishda qo‘llaniladigan kalit so‘zlar, identifikatorlar va o‘zgarmaslar yoritilgan. Shuningdek, o‘zgaruvchilar, ma’lumotlar turlari va ularni e’lon qilish qoidalari tushuntirilgan. C++ tilida operatorlar va ifodalarni yozishning asosiy qoidalari bayon etilgan. Dastur kodini yozishda izohlar, ajratkichlar va maxsus belgilardan foydalanish ko‘rsatib berilgan. Mavzuga oid nazariy ma’lumotlar amaliy misollar bilan mustahkamlangan. Ma’ruza talabalarning C++ dasturlash tiliga oid boshlang‘ich bilimlarini shakllantirishga xizmat qiladi.

Ma’ruza matni 1-bosqich Axborot tizimlari va texnologiyalari ta’lim yo‘nalishi talabalariga mo‘ljallangan.

Taqrizchilar:

Abduraximov D.B. – GulDU, Axborot texnologiyalari kafedrasi dotsenti, p.f.n.

Dushaboev O.N.- GulDU, Matematika kafedrasi dotsenti, p.f.f.d.(PhD).

Ma’ruza matni Axborot texnologiyalari va fizika-matematika fakulteti Ilmiy Kengashi qaroriga asosan (____ ____ 2026 yil ____-bayonnoma) nashrga tavsiya qilindi.

C++ TILINING LEKSIK ASOSLARI

Maqsad: Talabalarga C++ dasturlash tilining leksik asoslari, alifbosi, kalit so‘zlari, identifikatorlari, o‘zgaruvchilari, ma’lumotlar turlari va operatorlari haqida nazariy bilim berish hamda dastur kodini to‘g‘ri yozish bo‘yicha amaliy ko‘nikmalarni shakllantirish.

Reja:

1. C++ dasturlash tili va uning leksik asoslari
2. C++ tilining alifbosi va maxsus belgilari
3. Kalit so‘zlar, identifikatorlar va o‘zgarmaslar
4. O‘zgaruvchilar va ma’lumotlar turlari
5. Operatorlar, ifodalar va izohlardan foydalanish qoidalari

1. C++ DASTURLASH TILI VA UNING LEKSIK ASOSLARI

Hozirgi kunda juda ko‘p algoritmik tillar mavjud. Bular ichida ***Java, C++*** va ***C#*** (Si sharp) dasturlash tillari universal tillar hisoblanib, boshqa tillarga qaraganda imkoniyatlari kengroqdir. So‘ngi yillarda ***Java, C++*** va ***C#*** dasturlash tillari juda takomillashib, tobora ommalashib bormoqda. Mazkur tillardagi vositalar zamonaviy kompyuter texnologiyasining hamma talablarini o‘z ichiga olgan va unda dastur tuzuvchi uchun ko‘pgina qulayliklar yaratilgan.

C++ 1980 yillar boshida **Bjarne Stroustrup** tomonidan C tiliga asoslangan tarzda tuzildi. C++ juda ko‘p qo‘shimchalarni o‘z ichiga olgan,lekin eng asosiysi u obyektlar bilan dasturlashga imkon beradi. Dasturlarni tez va sifatli yozish hozirgi kunda katta ahamiyat kasb etmoqda. Buni ta’minlash uchun obyektli dasturlash g’oyasi ilgari surildi. Huddi 1970 yillar boshida strukturali dasturlash kabi, dasturlarni hayotdagi jismlarni modellashtiruvchi obyektlat orqali tuzish dasturlash sohasida inqilob qildi.

C++ tilidan turli xil dasturiy ta’minotlar, kompyuter o‘yinlari, operatsion tizimlar, ma’lumotlar bazalari, mobil va boshqa tizimlarni ishlab chiqishda foydalaniladi.

C++ tilining asosiy xususiyatlari:

- **Tezkorlik** - dasturlar yuqori tezlikda ishlashi mumkin.
- **Obyektga yo‘naltirilganlik** - sinf va obyektlar yordamida dastur yaratish imkoniyati mavjud.
- **Ko‘p imkoniyatlilik** - turli sohalardagi masalalarni yechishda qo‘llaniladi.
- **Xotirani boshqarish** - dasturchi xotiradan foydalanishni boshqarishi mumkin.
- **Ko‘chma kod** - standartlarga mos yozilgan dasturlarni turli platformalarda kompilyatsiya qilish mumkin.

C++ tilining leksik asoslari

Leksik asoslar - dasturlash tilidagi eng kichik ma’noli elementlar va ularni yozish qoidalaridir. C++ tilida dastur maxsus belgilar, kalit so‘zlar, identifikatorlar, o‘zgarmaslar, operatorlar va boshqa leksik elementlardan tashkil topadi.

C++ tilining asosiy leksik elementlari quyidagilardan iborat:

1. Alifbo va maxsus belgilar
2. Kalit so‘zlar
3. Identifikatorlar
4. O‘zgarmaslar
5. O‘zgaruvchilar
6. Operatorlar va ifodalar
7. Izohlar

2. C++ TILINING ALIFBOSI VA MAXSUS BELGILARI

C++ tilining alifbosi dastur yozishda ishlatiladigan belgilar to‘plamidan iborat. Asosiy belgilar quyidagilar:

№	Belgilar turi	Misollar
1	Lotin harflari	A-Z, a-z
2	Raqamlar	0-9
3	Arifmetik belgilar	+, -, *, /, %
4	Taqqoslash belgilari	==, !=, <, >, <=, >=
5	Mantiqiy belgilar	&&, `
6	Ajratuvchi belgilar	;, ,, (,), {, }
7	Boshqa maxsus belgilar	_, #, "

C++ tilida katta va kichik harflar farqlanadi. Masalan, *son, Son* va *SON* nomlari uchta alohida identifikator hisoblanadi.

Misol:

```
#include <iostream>
using namespace std;
int main() {
   cout << "Salom, dunyo!";
   return 0;
}
```

Ushbu dastur ekranga **Salom, dunyo!** matnini chiqaradi.

3. KALIT SO‘ZLAR, IDENTIFIKATORLAR VA O‘ZGARMASLAR

🔑 **3.1. Kalit so‘zlar** - C++ dasturlash tilida oldindan belgilangan, maxsus ma’noga ega bo‘lgan va muayyan vazifani bajaradigan so‘zlardir. Ular dastur tuzilishini, boshqaruv jarayonlarini va ma’lumotlar bilan ishlashni tashkil etishda qo‘llaniladi.

Masalan, *int, float, if, else, for, while, return* kabi so‘zlar C++ tilining kalit so‘zlari hisoblanadi.

Quyidagi jadvalda ayrim kalit so‘zlar keltirilgan:

Kalit so‘z	Vazifasi	Misol
int	Butun sonli tur	int a = 10;
float	Haqiqiy sonli tur	float x = 3.5f;
double	Aniqligi yuqori haqiqiy son turi	double y = 5.25;
char	Bitta belgi saqlaydi	char belgi = 'A’;
bool	Mantiqiy qiymat saqlaydi	bool holat = true;
if	Shartni tekshiradi	if (a > 0)
else	Muqobil shart holati	else
for	Takrorlash operatori	for (int i=0; i<5; i++)
while	Shart asosida takrorlash	while (a < 10)
return	Funksiyadan qiymat qaytaradi	return 0;
const	O‘zgarmas qiymat e’lon qiladi	const int n = 5;
void	Qiymat qaytarmaydigan funksiya turi	void salom()

Muhim qoida: kalit so‘zlarni o‘zgaruvchi yoki funksiya nomi sifatida ishlatib bo‘lmaydi.

Noto‘g‘ri yozilish:

```
int for = 10;
```

To‘g‘ri yozilish:

```
int son = 10;
```

3.2. Identifikatorlar va ularni yozish qoidalari

Identifikator - dasturda o‘zgaruvchi, funksiya, massiv, sinf yoki boshqa dasturiy elementlarga beriladigan nom.

Masalan:

```
int yosh = 20;
double maosh = 4500000;
char baho = 'A’;
```

Ushbu misollarda yosh, maosh va baho identifikatorlardir.

🏷️ Identifikator yozish qoidalari

- Identifikator harf (A-Z, a-z) yoki pastki chiziq (_) bilan boshlanishi kerak
- Raqam bilan boshlanishi mumkin emas
- Identifikator tarkibida harflar, raqamlar va pastki chiziq ishlatilishi mumkin
- Bo‘sh joy ishlatilmaydi
- C++ kalit so‘zlaridan identifikator sifatida foydalanilmaydi
- Katta va kichik harflar farqlanadi

To‘g‘ri identifikatorlar	Noto‘g‘ri identifikatorlar	Sababi
son	2son	Raqam bilan boshlangan
talaba_yoshi	talaba yoshi	Bo‘sh joy bor
summa1	summa-1	- belgisi ishlatilgan
_qiymat	int	int - kalit so‘z
umumiySumma	float	float - kalit so‘z

C++ tilida *son, Son* va *SON* uchta turli identifikator hisoblanadi.

Masalan:

```
int son = 10;
int Son = 20;
int SON = 30;
```

Bu dasturda uchta alohida o‘zgaruvchi mavjud.

Tavsiya: identifikatorlarni mazmuniga mos tanlash maqsadga muvofiq. Masalan, *talabaSoni*, *kitobNarxi*, *yigindi* kabi nomlar dastur kodini tushunishni osonlashtiradi.

3.3. O‘zgarmaslar (konstantalar)

💡 **O‘zgarmas** - dastur bajarilishi davomida qiymati o‘zgartirilmasligi kerak bo‘lgan ma’lumotdir. C++ tilida o‘zgarmaslarni e’lon qilish uchun const kalit so‘zidan foydalaniladi.

Misol:

```
const int yosh = 20;
const double PI = 3.14159;
```

Bu yerda yosh va PI o‘zgarmas qiymatlardir. Ularning qiymatini keyinchalik o‘zgartirishga urinish kompilyatsiya xatosiga olib keladi.

C++ dasturlash tilida o‘zgarmaslar (constants):

№	O‘zgarmas turi	Qiymat misollari	Kodda qo‘llanilishi (misol)
1	Butun sonli o‘zgarmaslar	10, 25, -7, 100	const int n = 10;
2	Haqiqiy sonli o‘zgarmaslar	3.14, 2.5, -0.75	const double PI = 3.14;
3	Belgili o‘zgarmaslar	'A’, 'B', '7'	const char harf = 'A’;
4	Satrli o‘zgarmaslar	"Salom", "C++ dasturlash tili"	const char* matn = "Salom";
5	Mantiqiy o‘zgarmaslar	true, false	const bool faol = true;

💡**O‘zgarmaslarni e’lon qilish:**

O‘zgarmas quyidagi shaklda e’lon qilinadi:

const ma’lumot_turi nom = qiymat;

Masalan:

```
const int a = 15;
const double narx = 12500.5;
const char belgi = 'K';
const bool natija = true;
```

4. O‘ZGARUVCHILAR VA MA’LUMOTLAR TURLARI

4.1. Ma’lumotlar turlari (Data Types)

Har qanday dasturlash tilining asosi ma’lumotlarni saqlash va ular ustida amallarni bajarishdan iborat. C++ tilida ma’lumotlar xotirada saqlanadi va ular bilan ishlash uchun **o‘zgaruvchilar** hamda **ma’lumotlar turlari** ishlatiladi. Ushbu ma’ruzada biz C++ dagi asosiy ma’lumotlar turlari, o‘zgaruvchilarni e’lon qilish qoidalari va ulardan foydalanish usullarini ko‘rib chiqamiz.

💡**C++ tilining asosiy ma’lumotlar turlari**

Ma’lumot turi	Kalit so‘z	Tavsifi	O‘lchami (odatda)	Misol
Butun sonlar	**int**	Butun qiymatlarni saqlaydi	4 bayt	int age = 25;
Haqiqiy sonlar	**float**	Kasr qismiga ega sonlar (aniqligi pastroq)	4 bayt	float temperature = 36.6f;
Haqiqiy sonlar	**double**	Katta va aniqroq kasr sonlar	8 bayt	double pi = 3.1415926535;
Belgi (Simvol)	**char**	Bitta belgini saqlaydi (ASCII jadvali bo‘yicha)	1 bayt	char grade = 'A’;
Mantiqiy qiymat	**bool**	Faqat ikki qiymatdan birini oladi: true yoki false	1 bayt	bool isOpen = true;
Matnli qiymatlar	**string**	Matnlarni saqlash uchun	Matn uzunligiga qarab xotiradan joy egallaydi.	string matn = "Salom";
Bo‘sh tur	**void**	Qiymat qaytarmaslikni bildiradi	-	Asosan funksiyalarda ishlatiladi

4.2. O‘zgaruvchilar (Variables)

💡**O‘zgaruvchi** – bu kompyuter xotirasidagi ma’lum bir qiymatni saqlash uchun ajratilgan nomlangan joy. o‘

O‘zgaruvchini e’lon qilish qoidalari:

- O‘zgaruvchi ishlatilishidan oldin uning turi ko‘rsatilib e’lon qilinishi shart.
- Nomlashda lotin alifbosi harflari (a-z, A-Z), raqamlar (0-9) va pastki chiziqcha (_) ishlatiladi.
- Nom raqam bilan boshlanmasligi kerak.
- C++ tili case-sensitive (katta-kichik harflarni farqlaydi), ya’ni summa va Summa ikki xil o‘zgaruvchidir.
- C++ kalit so‘zlari (int, return, double va hokazo) o‘zgaruvchi nomi sifatida ishlatilmaydi.

Sintaksis:

C++:

```
ma’lumot_turi o‘zgaruvchi_nomi = qiymat;
```

Misol:

C++:

```
int a = 10;                 // Butun sonli o‘zgaruvchi
double salary = 1200.50;    // Haqiqiy sonli o‘zgaruvchi
char letter = 'X';          // Belgi turidagi o‘zgaruvchi
```

4.3. O‘zgaruvchilarni e’lon qilish va nomlash qoidalari

C++ tilida o‘zgaruvchilarni to‘g‘ri nomlash dastur kodini tushunarli va tartibli yozishga yordam beradi.

💡**O‘zgaruvchilarni nomlashda quyidagi qoidalarga amal qilinadi:**

- O‘zgaruvchi nomi lotin harflari, raqamlar va pastki chiziq (_) belgisidan tashkil topishi mumkin.
- O‘zgaruvchi nomi raqam bilan boshlanmaydi.
- Nom tarkibida bo‘sh joy ishlatilmaydi.
- C++ tilining kalit so‘zlarini o‘zgaruvchi nomi sifatida ishlatib bo‘lmaydi.
- Katta va kichik harflar farqlanadi.

Misollar:

To‘g‘ri nomlar	Noto‘g‘ri nomlar	Sababi
yosh	2yosh	Raqam bilan boshlangan
talaba_ismi	talaba ismi	Bo‘sh joy mavjud
son1	int	int kalit so‘z
umumiySumma	umumiy-summa	Defis ishlatilgan
bahosi	bahosi!	Taqiqlangan belgi ishlatilgan

💡**O‘zgaruvchilarni e’lon qilishning bir nechta usuli mavjud:**

</> C++:

1-holat:

```
int a;
int b;
int c;
```

2-holat:

Yoki bir xil turdagi bir nechta o‘zgaruvchini bir qatorda e’lon qilish mumkin:

```
int a, b, c;
```

💡O‘zgaruvchiga qiymat berish esa quyidagicha amalga oshiriladi:

```
int a = 10;
int b = 20;
int yigindi = a + b;
```

Natijada **yigindi** o‘zgaruvchisida **30** qiymati saqlanadi.

4.4. Ma’lumotlar turlarining xususiyatlari

Ma’lumot turi	Vazifasi	Misol
short	Nisbatan kichik diapazondagi butun sonlar	short a = 100;
int	Butun sonlar	int a = 50;
long long	Katta diapazondagi butun sonlar	long long n = 9000000000LL;
float	Haqiqiy sonlar	float x = 5.5f;
double	Aniqligi yuqoriroq haqiqiy sonlar	double x = 5.55;
char	Bitta belgi	char c = 'B';
bool	Mantiqiy qiymat	bool holat = false;

Ma’lumot turlarining xotirada egallaydigan hajmi dastur ishlayotgan tizim va kompilyatorga bog‘liq bo‘lishi mumkin. Ko‘p uchraydigan tizimlarda int 4 bayt, float 4 bayt va double 8 baytni egallaydi. char esa 1 baytni egallaydi.

💡**Aniq hajmni sizeof operatori yordamida aniqlash mumkin:**

</>C++:

```
#include <iostream>
using namespace std;

int main() {
   cout << sizeof(int) << endl;
   cout << sizeof(float) << endl;
   cout << sizeof(double) << endl;
   cout << sizeof(char) << endl;

   return 0;
}
```

Bu dastur tegishli ma’lumot turlarining xotiradagi hajmini baytlarda chiqaradi.

4.5. Matnli ma’lumotlarni saqlash

C++ tilida matnlarni saqlash uchun **string** turidan foydalanish mumkin. Buning uchun **<string>** kutubxonasi ulanadi.

Misol:

```
#include <iostream>
#include <string>
using namespace std;

int main() {
   string ism = "Ali";
   cout << ism;

   return 0;
}
```

Natija:

```
Ali
```

char bitta belgini, **string** esa belgilar ketma-ketligini, ya’ni matnni saqlash uchun ishlatiladi.

Masalan:

```
char harf = 'A';
string ism = "Aziza";
```

E’tibor bering: char qiymati odatda bitta qo‘shtirnoqsiz, ya’ni yakka tirnoq ichida ('A’), string qiymati esa qo‘shtirnoq ichida (“Aziza”) yoziladi.

5. OPERATORLAR, IFODALAR VA IZOHLARDAN FOYDALANISH QOIDALARI

5.1. C++ dasturlash tilida operatorlar haqida tushuncha

💡C++ dasturlash tilida **operator** - ma’lumotlar ustida muayyan amal bajarishni ifodalovchi belgi yoki kalit so‘zdir. Operatorlar yordamida sonlarni qo‘shish, ayirish, ko‘paytirish, bo‘lish, taqqoslash va mantiqiy shartlarni tekshirish mumkin.

Operatorlar dasturdagi hisoblash jarayonlarini tashkil etadi va ma’lumotlarni qayta ishlashda muhim o‘rin tutadi.

Masalan:

</>C++:

```
int a = 10;
int b = 5;
int c = a + b;
```

Ushbu dasturda:

- **int** - butun sonli o‘zgaruvchi turini bildiradi.
- = - qiymat berish operatori.
- + - qo‘shish operatori.
- **a + b** - arifmetik ifoda.
- **;** - operator yoki e’lonni yakunlovchi belgi.

Natijada **c** o‘zgaruvchisiga **15** qiymati beriladi.

5.2. Operatorlarning asosiy turlari

C++ dasturlash tilida operatorlar bajaradigan vazifasiga qarab bir necha guruhga bo‘linadi.

💡Arifmetik operatorlar

Arifmetik operatorlar matematik hisoblashlarni bajarish uchun ishlatiladi.

Operator	Vazifasi	Misol	Natija
+	Qo‘shish	10 + 5	15

-	Ayirish	10 - 5	5
*	Ko‘paytirish	10 * 5	50
/	Bo‘lish	10 / 5	2
%	Qoldiqni aniqlash	10 % 3	1

💡 Muhim qoida: agar ikkala operand ham butun son bo‘lsa, / operatori butun sonli bo‘lishni bajaradi.

Masalan:

```
cout << 7 / 2;
```

Natija:

```
3
```

Agar kasrli natija kerak bo‘lsa, operandlardan kamida bittasi kasrli son bo‘lishi kerak:

```
cout << 7.0 / 2;
```

Natija:

```
3.5
```

% operatori butun sonlar bilan ishlatiladi va bo‘lishdan qolgan qoldiqni qaytaradi.

5.3. Qiymat berish operatorlari

Qiymat berish operatorlari o‘zgaruvchilarga qiymat yozish yoki ularning qiymatini o‘zgartirish uchun ishlatiladi.

Operator	Misol	Mazmuni
=	a = 10;	a ga 10 qiymatini beradi
+=	a += 5;	a = a + 5
-=	a -= 3;	a = a - 3
*=	a *= 2;	a = a * 2
/=	a /= 2;	a = a / 2

Masalan:

```
int a = 10;
a += 5;
cout << a;
```

Natija: **15.**

5.4. Taqqoslash operatorlari

Taqqoslash operatorlari ikkita qiymatni solishtirish uchun qo‘llaniladi. Ularning natijasi mantiqiy qiymat bo‘lib, true yoki false ko‘rinishida ifodalanadi.

Operator	Mazmuni
==	Teng
!=	Teng emas
>	Katta
<	Kichik
>=	Katta yoki teng
<=	Kichik yoki teng

Misol:

```
int a = 10;
int b = 5;
cout << (a > b);
```

Natija:

1

Odatda C++ da true qiymati **1**, false qiymati esa **0** sifatida chiqariladi.

💡 Eslatma: = va == operatorlarini farqlash kerak. = qiymat beradi, == esa qiymatlarning tengligini tekshiradi.

5.5. Mantiqiy operatorlar

Mantiqiy operatorlar bir yoki bir nechta shartni tekshirishda ishlatiladi.

Operator	Nomi	Vazifasi
&&	Mantiqiy VA (AND)	Ikkala shart ham rost (true) bo‘lsagina, natija rost bo‘ladi.
||	Mantiqiy YOKI (OR)	Kamida bitta shart rost (true) bo‘lsa, natija rost bo‘ladi.
!	Mantiqiy EMAS (NOT)	Mantiqiy qiymatni teskarisiga o‘zgartiradi (rostni yolg‘onga, yolg‘onni rostga).

Misol:

```
int yosh = 20;
if (yosh >= 18 && yosh <= 30)
{
   cout << "Shart bajarildi";
}
```

Bu yerda yosh 18 dan kam bo‘lmasligi va 30 dan katta bo‘lmasligi kerak. Ikkala shart bajarilganda ekranga *Shart bajarildi* yozuvi chiqadi.

5.6. Inkrement va dekrement operatorlari

Bu operatorlar o‘zgaruvchining qiymatini bittaga oshirish yoki kamaytirish uchun ishlatiladi.

++ - qiymatni 1 ga oshiradi.

-- - qiymatni 1 ga kamaytiradi.

Misol:

```
int a = 5;
a++;
cout << a;
```

Natija: 6.

💡 Inkrement va dekrement operatorlarining ikki xil yozilish shakli mavjud:

- ++**a** - *prefiks shakli.*
- **a**++ - *postfiks shakli.*

Ular murakkab ifodalarda farq qiladi: *prefiks shakli* avval qiymatni o‘zgartiradi, *postfiks shakli* esa ifodada avval eski qiymatdan foydalanadi.

5.7. Ifodalarni yozish qoidalari

1. O‘zgaruvchilar oldindan e’lon qilingan bo‘lishi kerak
2. Amallarda ma’lumot turlariga mos qiymatlardan foydalanish lozim
3. Har bir operatorning vazifasini to‘g‘ri tanlash kerak
4. Zarur holatlarda qavslar qo‘llanilishi lozim
5. Nolga bo‘lish kabi xatolarga yo‘l qo‘ymaslik kerak
6. Murakkab ifodalarni tushunarli shaklda yozish tavsiya etilad

5.8. Dastur matnida izohlardan foydalanish qoidalari

💡 **Izoh (comment)** - dastur kodining vazifasini tushuntirish uchun yoziladigan, kompilyator tomonidan bajarilmaydigan matn. Izohlar dasturchiga kodni tushunish, xatolarni aniqlash va dasturni keyinchalik o‘zgartirishda yordam beradi.

💡 C++ tilida izohlarning ikkita asosiy turi mavjud:

1. **Bir qatorli izoh**

Bir qatorli izoh // belgisi yordamida yoziladi. Ushbu belgidan keyingi matn qator oxirigacha izoh hisoblanadi.

Misol:

```
int a = 10; // a o‘zgaruvchisiga 10 qiymati berildi
int b = 5;  // b o‘zgaruvchisiga 5 qiymati berildi
int c = a + b; // ikki son qo‘shildi
cout << c;
```

Natija:

```
15
```

Izohlar dastur bajarilishiga ta’sir qilmaydi.

2. **Ko‘p qatorli izoh**

Ko‘p qatorli izoh /* va */ belgilari orasida yoziladi.

Misol:

```
/* Ushbu dastur ikkita butun sonni qo‘shadi.
Natija ekranga chiqariladi.*/
int a = 10;
int b = 20;
cout << a + b;
```

Natija:

```
30
```

GLOSSARY (ГЛОССАРИЙ)

Atamaning nomlanishi				
T/r	**O‘zbek tilida**	**Rus tilida**	**Ingliz tilida**	**Atamaning ma’nosi**
1	**C++ dasturlash tili**	Язык программирования C++	C++ Programming Language	Dastur yaratish uchun ishlatiladigan dasturlash tili
2	**Leksik asoslar**	Лексические основы	Lexical Fundamentals	Dasturdagi belgilar, kalit so‘zlar, identifikatorlar va boshqa leksik elementlar majmui
3	**Alfavit**	Алфавит	Alphabet	Dastur matnida foydalaniladigan harflar, raqamlar to‘plami
4	**Token**	Токен	Token	Dastur matnidagi eng kichik leksik birlik
5	**Kalit so‘z**	Ключевое слово	Keyword	C++ tilida maxsus ma’noga ega bo‘lgan oldindan belgilangan so‘z
6	**Identifikator**	Идентификатор	Identifier	O‘zgaruvchi, funksiya yoki sinfga beriladigan nom
7	**O‘zgaruvchi**	Переменная	Variable	Dastur bajarilishi davomida qiymati o‘zgarishi mumkin bo‘lgan nomlangan xotira sohasi
8	**O‘zgarmas**	Константа	Constant	Dasturda qiymati o‘zgartirilmaydigan kattalik
9	**Literal**	Литерал	Literal	Dastur kodida bevosita ko‘rsatilgan qiymat, masalan, 25 yoki "Salom"
10	**Ma’lumot turi**	Тип данных	Data Type	Ma’lumotning turi, saqlanishi va u bilan bajariladigan amallarni belgilaydi
11	**Butun son**	Целое число	Integer	Kasr qismisiz ifodalanadigan son. Masalan, 10.
12	**Haqiqiy son**	Вещественное число	Floating-point Number	Kasr qismini ham ifodalash mumkin bo‘lgan son. Masalan, 3.14
13	**Belgi**	Символ	Character	Bitta belgi yoki harfni ifodalovchi ma’lumot. Masalan, 'A’
14	**Mantiqiy tur**	Логический тип	Boolean Type	true yoki false qiymatlaridan birini qabul qiladigan ma’lumot turi
15	**Satr**	Строка	String	Belgilar ketma-ketligi. Masalan, "Hello"

16	**Operator**	Оператор	Operator	Ma’lumotlar ustida muayyan amal bajaruvchi belgi yoki kalit so‘z
17	**Ifoda**	Выражение	Expression	Qiymat hosil qiluvchi operandlar va amallar birikmasi
18	**Izoh**	Комментарий	Comment	Dastur kodini tushuntirish uchun yoziladigan, bajarilmaydigan matn
19	**Bir qatorli izoh**	Однострочный комментарий	Single-line Comment	// belgisi bilan boshlanadigan bir qatorli izoh
20	**Ko‘p qatorli izoh**	Многострочный комментарий	Multiline Comment	/* va */ belgilari orasida yoziladigan izoh
21	**Ajratkich**	Разделитель	Separator	Dastur elementlarini ajratish yoki guruhlash uchun ishlatiladigan belgi
22	**Nuqtali vergul**	Точка с запятой	Semicolon	C++ tilida ko‘plab operatorlar oxiriga qo‘yiladigan ; belgisi
23	**Kompilyator**	Компилятор	Compiler	Dastur kodini mashina kodi yoki boshqa bajariladigan shaklga tarjima qiluvchi vosita
24	**Sintaksis**	Синтаксис	Syntax	Dastur elementlarini to‘g‘ri yozish va birlashtirish qoidalari
25	**Registrga sezgirlik**	Чувствительность к регистру	Case Sensitivity	Katta va kichik harflarning farqlanishi. Masalan, son va Son boshqa-boshqa nomlardir
26	**Direktiva**	Директива	Directive	Kompilyatsiyadan oldin qayta ishlanadigan ko‘rsatma. Masalan, #include
27	**Kutubxona**	Библиотека	Library	Dasturlarda foydalanish mumkin bo‘lgan tayyor funksiyalar va boshqa dasturiy vositalar to‘plami
28	**Sarlavha fayli**	Заголовочный файл	Header File	Deklaratsiyalar va boshqa kerakli ma’lumotlarni o‘z ichiga oluvchi fayl. Masalan, <iostream>
29	**Nomlar fazosi**	Пространство имён	Namespace	Dasturdagi nomlar to‘qnashuvining oldini olishga yordam beruvchi mexanizm
30	**Dastur kodi**	Исходный код программы	Source Code	Dasturchi tomonidan dasturlash tilida yozilgan buyruqlar va ifodalar matni