

1-MAVZU: C++ ASOSLARI (SINTAKSIS VA MANTIQ)

1. Nazariy qism

1. C++ tili nima?

C++ — bu:

- kompilyatsiyalanadigan,
- tez ishlaydigan,
- past darajadagi boshqaruv + yuqori darajadagi qulaylikni birlashtirgan dasturlash tili.

Olimpiadalarda C++ tanlanish sababi:

- vaqt cheklovi qat'iy bo'lgan masalalarda eng tez ishlaydi;
- xotirani aniq boshqarish mumkin;
- murakkab algoritmlar uchun ideal.

2. Kompilyator va dastur ishlash jarayoni

C++ dastur quyidagi bosqichlarda ishlaydi:

1 Manba kod (.cpp fayl) yoziladi

2 Kompilyator (masalan g++) uni mashina kodiga o'giradi

3 Xatolar bo'lmasa → ishga tushadi

Muhim farq:

- Python → interpretatsiya
- C++ → kompilyatsiya (oldindan)

3. Eng sodda C++ dastur tuzilishi

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

```
int main() {
    cout << "Salom, C++!";
    return 0;
}
```

Tahlil:

```
#include <iostream>
```

- kiritish–chiqarish kutubxonasi

- `cin` va `cout` shu yerda e'lon qilingan

```
using namespace std;
```

- `std::cout` o'rniga `cout` yozish uchun Olimpiadalarda deyarli doim ishlatiladi

```
int main()
```

- dastur shu yerdan boshlanadi
- `int` — qaytariladigan qiymat turi

```
return 0;
```

- dastur muvaffaqiyatli tugadi degani

4. Kiritish va chiqarish

Chiqarish (`cout`)

```
cout << "Natija = " << x;
```

`<<` — oqimga uzatish operatori

Kiritish (`cin`)

```
cin >> x;
```

`>>` — oqimdan olish operatori

Bir nechta o'zgaruvchi:

```
cin >> a >> b >> c;
```

5. O'zgaruvchilar va ma'lumot turlari

Asosiy turlar:

Tur	Izoh
<code>int</code>	butun son ($\approx \pm 2 \cdot 10^9$)
<code>long long</code>	katta butun son
<code>double</code>	haqiqiy son
<code>char</code>	bitta belgi
<code>bool</code>	<code>true</code> / <code>false</code>

Misol:

```
int a = 5;  
long long b = 100000000000;  
double x = 3.14;  
char c = 'A';  
bool ok = true;
```

Olimpiadalarda long long juda muhim.

6. Arifmetik amallar

Amal	Belgisi
Qo'shish	+
Ayirish	-
Ko'paytirish	*
Bo'lish	/
Qoldiq	%

Eslatma:

- $7 / 2 = 3$ (butun bo'lish!)
- $7 \% 2 = 1$

7. Taqqoslash amallari

Amal	Ma'nosi
==	teng
!=	teng emas
<	kichik
>	katta
<=	kichik yoki teng
>=	katta yoki teng

Natija bool bo'ladi.

8. Mantiqiy amallar

Amal Belgisi

VA &&

YOKI ^

EMAS !

Misol:

```
if (a > 0 && b > 0)
```

9. Shart operatori if

```
if (shart) {  
    amal;  
}
```

if - else

```
if (a % 2 == 0) {  
    cout << "Juft";  
} else {  
    cout << "Toq";  
}
```

Ichma-ich if

```
if (a > 0) {  
    if (a % 2 == 0) {  
        cout << "Musbat juft";  
    }  
}
```

10. switch - case

```
switch (x) {  
    case 1:  
        cout << "Bir";  
        break;  
    case 2:  
        cout << "Ikki";  
        break;  
    default:  
        cout << "Boshqa";  
}
```

break bo'lmasa — keyingi case ham bajariladi (xato manbai!).

11. Takrorlash operatorlari

for

```
for (int i = 1; i <= n; i++) {  
    cout << i;
```

```
}
```

```
while
```

```
while (x > 0) {  
    x--;  
}
```

```
do - while
```

```
do {  
    cin >> x;  
} while (x <= 0);
```

do-while kam ishlatiladi, lekin bilish shart.

12. break va continue

📖 break → siklni to‘liq to‘xtatadi

📖 continue → joriy aylanishni o‘tkazib yuboradi

2. METODIK YORDAM ()

1. Mavzuni berish ketma-ketligi (tavsiya)

Darsni sintaksisdan emas, mantiqdan boshlash samaraliroq:

1 Kiritish → qayta ishlash → chiqarish modeli

2 O‘zgaruvchi nima uchun kerakligi

3 if va for real hayot misollarida

4 Keyin sintaksisni rasmiylashtirish

Natija: o‘quvchi kod yodlamaydi, fikrlaydi.

2. cin va cout ni to‘g‘ri o‘rgatish

Ko‘p o‘quvchilar faqat yozishni biladi, oqim mantiqini tushunmaydi.

Tavsiya:

- << ni “chiqarish trubkasi”
- >> ni “kiritish trubkasi” deb izohlash

Misol:

```
cin >> a >> b;  
cout << a + b;
```

3. Ma'lumot turlarini solishtirib berish

Faqat ro'yxat qilib o'tmaslik kerak.

Vaziyat	To'g'ri tanlov
Hisoblagich	int
Juda katta son	long long
O'rtacha qiymat	double
Ha/yo'q	bool

Olimpiada qoidasi:
shubha bo'lsa — long long.

4. Shart operatorini sxema bilan tushuntirish

if — qaror qabul qilish nuqtasi.

Tavsiya:

- Har bir if ga savol berib o'qitish

“a juftmi?”

“a musbatmi?”

Shartni tabiiy til → kod tarzida o'tkazish.

5. Sikllarni “necha marta?” savoli bilan berish

Sikl tanlashda asosiy savol:

- Necha marta? → for
- Qachongacha? → while
- Kamida bir marta? → do-while

Bu jadvalni yodlatish foydali.

6. Qoldiq (%) amaliga alohida urg'u

Olimpiadalarda eng ko'p ishlatiladi:

- juft/toq
- karrali
- oxirgi raqam
- bo'linish sharti

Shu amal asosida ko'p mini-masala berish tavsiya qilinadi.

7. Kod yozish madaniyatini boshidan shakllantirish

- To‘g‘ri chekinish (indentatsiya)
- Aniq o‘zgaruvchi nomlari
- Keraksiz kod yozmaslik

Bu keyinchalik debugni osonlashtiradi.

8. Har bir mavzudan keyin mini-test yoki mini-masala

Masalan:

- faqat `if`
- faqat `for`
- faqat `%`

Bu mustahkamlashni tezlashtiradi.

3. TIPIK XATOLAR (kengaytirilgan va tahlilli)

1. `=` va `==` ni adashtirish

```
if (a = 5) // XATO
```

To‘g‘risi:

```
if (a == 5)
```

Sabab: tayinlash va solishtirishni farqlamaslik.

2. Butun bo‘lishni unutish

```
double x = 5 / 2; // x = 2
```

To‘g‘risi:

```
double x = 5.0 / 2;
```

Juda ko‘p testlarda shu bilan yiqiladi.

3. `long` o‘rniga `int` ishlatish

```
int n;  
cin >> n; // n = 10^12 bo‘lsa - XATO
```

Olimpiada masalalarida eng xavfli xato.

4. switch da break ni unutish

```
case 1:  
    cout << "Bir";  
    // break yo'q
```

Natija noto'g'ri chiqadi.

5. Sikl chegarasida xato

```
for (int i = 0; i <= n; i++) // ko'pincha ortiqcha  
    <= va < farqini tushuntirish shart.
```

6. if shartida ortiqcha qavs

```
if ((a > 0) == true)
```

To'g'risi:

```
if (a > 0)
```

7. continue va break ni chalkashtirish

- break → sikldan chiqadi
- continue → faqat shu aylanishni o'tkazadi

8. Kiritishsiz ishlatish

```
int a;  
cout << a; // noma'lum qiymat
```

Boshlovchilar xatosi, lekin juda xavfli.

9. Qavslar bilan ishlashda mantiqiy xato

```
if (a > 0 && b > 0 || c > 0)
```

To'g'risi:

```
if ((a > 0 && b > 0) || c > 0)
```

METODIK XULOSA

Agar:

- mantiq → keyin sintaksis

- real misol $\rightarrow$ keyin kod
- xato $\rightarrow$ sabab $\rightarrow$ to'g'ri yechim

usuli bilan o'tilsa, C++ asoslari juda tez va mustahkam o'zlashtiriladi.