1

BIRINCHI MA’RUZA

Dasturlash 1 · C++ · 1-mavzu

# Dasturlash tillarining tuzilmasi

Sintaksis va semantika · preprotsessor · operatorlar · tur o‘zgartirish · kutubxonalar

Sintaksis | Preprotsessor | Operatorlar | cmath

ISINISH

# Kompyuter “Salom, dunyo!” degan matnni qanday tushunadi?

Hozircha javob bermang. Avval dastur qanday yo‘l bosib o‘tishini taxmin qiling.

1

Kompyuter inson yozgan matnni to‘g‘ridan-to‘g‘ri bajara oladimi?

2

Kod yozilgandan keyin uni kim yoki nima “tarjima” qiladi?

3

Kodda bitta “;” tushib qolsa, nima bo‘ladi deb o‘ylaysiz?

O‘QUV MAQSADLARI

# Bu ma’ruza oxirida siz quyidagilarni bajara olasiz...

1. **Tushuntirish** — sintaksis va semantika farqini hamda C++ tilining asosiy xususiyatlarini
2. **Tavsiflash** — manba koddan bajariladigan faylgacha bo‘lgan kompilyatsiya bosqichlarini
3. **Qo‘llash** — #include, #define, #ifdef preprotsessor direktivalarini
4. **Aniqlash** — identifikator nomi to‘g‘ri yoki noto‘g‘ri ekanini qoidalar asosida
5. **Farqlash** — prefiks va postfiks inkrement, bitli va mantiqiy operatorlarni
6. **Hisoblash** — ifoda qiymatini operatorlar ustuvorligini hisobga olib
7. **O‘zgartirish** — ma’lumot turini (int)x va static_cast yordamida
8. **Foydalanish** — <cmath> kutubxonasidagi matematik funksiyalardan

01

# Tilning bazaviy tushunchalari

Sintaksis va semantika · C++ xususiyatlari · kompilyatsiya bosqichlari

Sintaksis | Semantika | main() | Kompilyatsiya

# Dasturlash tili — kompyuter bilan muloqot tili

**Dasturlash tili — kompyuterga buyruq berish uchun yaratilgan sun’iy til. U ikki qismdan iborat: sintaksis va semantika.**

## Sintaksis

Kod qanday yozilishi kerakligi: kalit so‘zlar, qavslar, `;` belgisi. Qoidaga zid yozuv — sintaktik xato.

## Semantika

Yozilgan kod nimani anglatishi. `a = b + c;` — b va c yig‘indisini a ga yozish degani.

## Kompilyator

Sintaksisni tekshiradi va to‘g‘ri yozilgan kodni protsessor tushunadigan mashina kodiga o‘giradi.

Sintaksis to‘g‘ri bo‘lsa ham, mantiq noto‘g‘ri bo‘lishi mumkin — bunday xatoni kompilyator topa olmaydi.

C++ XUSUSIYATLARI

# C++ qanday til?

| Xususiyat | Ma’nosi | Nima beradi |
|---|---|---|
| **Yuqori darajali** | Kod inson tiliga yaqin: `if`, `for`, `cout` | O‘qish va yozish oson |
| **Kompilyatsiyalanadigan** | Ishga tushirishdan oldin mashina kodiga o‘giriladi | Juda tez ishlaydi |
| **Statik tiplangan** | Har bir o‘zgaruvchining turi oldindan e’lon qilinadi | Xatolar erta topiladi |
| **Umumiy maqsadli** | O‘yinlar, OT, brauzerlar, ilmiy hisoblar | Keng qo‘llaniladi |

C++ dasturining elementlari: sintaktik tuzilmalar, o‘zgaruvchilar va turlar, operatorlar, funksiyalar, ma’lumotlar tuzilmalari.

# C++ dasturining tuzilishi

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

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

▶ NATIJA

```
Salom, dunyo!
```

**`#include <iostream>`**

kiritish-chiqarish kutubxonasini qo‘shadi

**`using namespace std;`**

`std::` prefiksini har safar yozmaslik uchun

**`int main()`**

dastur aynan shu funksiyadan boshlanadi

**`cout <<`**

ekranga chiqarish; `endl` — yangi qator

**`return 0;`**

dastur muvaffaqiyatli tugaganini bildiradi

KOMPILYATSIYA

# Manba koddan ishlaydigan dasturgacha

1

## Manba kod

`main.cpp` — dasturchi yozgan matn

2

## Preprotsessor

`#include`, `#define` ni ochadi, matnni tayyorlaydi

3

## Kompilyator

Sintaksisni tekshiradi, obyekt fayl (`.obj`) yaratadi

4

## Bog‘lovchi

Kutubxonalarni ulaydi, `.exe` fayl hosil qiladi

5

## Bajarish

Protsessor mashina kodini ishga tushiradi

Xato turlari: sintaktik (kompilyatsiyada), bog‘lash xatosi (linker), bajarish vaqtidagi xato (masalan, 0 ga bo‘lish).

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 1-bo‘lim yakuni: Tilning bazaviy tushunchalari

1

Sintaksis bilan semantikaning farqi nimada? Misol keltiring.

Savol 1

2

C++ dasturining bajarilishi qaysi funksiyadan boshlanadi?

Savol 2

3

`;` unutilgan kod qaysi bosqichda xato beradi?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

# 1-topshiriq: Salom, ism!

Foydalanuvchi o‘z ismini kiritadi. Dastur unga salom berishi kerak.

**Kirish:** Bitta so‘z — ism (bo‘sh joysiz).

**Chiqish:** `Salom, <ism>!` ko‘rinishidagi bitta qator.

| MISOL · KIRISH | MISOL · CHIQISH |
| --- | --- |
| `Aziza` | `Salom, Aziza!` |

Telefon kamerasi bilan skanerlang

1. Saytga kiring (shaxsiy kod)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

02

# Preprotsessor direktivalari

#include · #define · #ifdef / #ifndef / #endif · makro tuzog‘i

#include

#define

#ifdef

const

PREPROTSESSOR

# # bilan boshlanadigan buyruqlar

Preprotsessor kompilyatsiyadan oldin ishlaydi. U kodni tushunmaydi — faqat matnni almashtiradi.

### 1 #include

Kutubxona yoki faylni qo‘shadi:
`#include <iostream>`

### 2 #define

Makro yoki konstanta belgilaydi:
`#define PI 3.14`

### 3 #ifdef

Makro aniqlangan bo‘lsa, blokni qoldiradi

### 4 #ifndef

Makro aniqlanmagan bo‘lsa, blokni qoldiradi

### 5 #endif

Shartli blokni yakunlaydi

### 6 <...> va "..."

`<>` — standart kutubxona, `""` — o‘zingizning faylingiz

# Makro va shartli kompilyatsiya

```cpp
#include <iostream>
using namespace std;
#define PI 3.14
#define DEBUG

int main() {
    cout << "PI = " << PI << endl;
#ifdef DEBUG
    cout << "Tuzatish rejimi" << endl;
#endif
    return 0;
}
```

▶ NATIJA

```
PI = 3.14
Tuzatish rejimi
```

**`#define PI 3.14`**

kodda har bir `PI` so‘zi `3.14` bilan almashtiriladi

**`#define DEBUG`**

qiymatsiz makro — faqat “bayroq”

**`#ifdef DEBUG`**

DEBUG bor — ichidagi qator kompilyatsiyaga kiradi

**Sinab ko‘ring**

`#define DEBUG` ni o‘chirsangiz, 2-qator chiqmaydi

MAKRO TUZOG‘I

# Qavssiz makro xavfli

## Noto‘g‘ri: #define KV(x) x * x

- `KV(2 + 3)` → `2 + 3 * 2 + 3`
- Ko‘paytirish avval bajariladi: `2 + 6 + 3 = 11`
- Kutilgan 25 o‘rniga 11 chiqadi

## To‘g‘ri: #define KV(x) ((x) * (x))

- `KV(2 + 3)` → `((2 + 3) * (2 + 3))`
- Natija: `25`
- Yaxshiroq: `const double PI = 3.14159;` — turi bor, kompilyator tekshiradi

Maslahat: konstantalar uchun `#define` o‘rniga `const` yoki `constexpr` ishlating.

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 2-bo‘lim yakuni: Preprotsessor direktivalari

1

Kutubxona qo‘shish uchun qaysi direktiva ishlatiladi?

Savol 1

2

`#define KV(x) x * x` bo‘lsa, `KV(1 + 1)` nechaga teng?

Savol 2

3

Nima uchun `const` `#define` dan xavfsizroq?

Savol 3

03

# Identifikatorlar va o‘zlashtirish

Nomlash qoidalari · = va == · qisqa o‘zlashtirish operatorlari

| Identifikator | = | += | == |
|---|---|---|---|

# Nom berish qoidalari

## Qoidalar

- Faqat lotin harflari, raqamlar va `_`
- Raqam bilan boshlanmaydi: `2son` — xato
- Bo‘sh joy va `-`, `.`, `@` mumkin emas
- Kalit so‘z (`int`, `for`, `return`) nom bo‘lmaydi
- Katta-kichik harf farqlanadi: `son`, `Son`, `SON`

## Misollar

✓ `son2`, `talaba_ismi`, `talabaIsmi`, `_temp`

✗ `2son` — raqam bilan boshlangan

✗ `talaba ismi` — bo‘sh joy bor

✗ `narx-1` — minus belgisi

✗ `int` — kalit so‘z

Ko‘rinish sohasi bo‘yicha identifikatorlar lokal, global va static bo‘ladi — bu 8-darsda o‘rganiladi.

▶ INTERAKTIV

# Bu nom to‘g‘rimi?

Har bir identifikatorni to‘g‘ri qutiga joylashtiring.  (Interaktiv slaydlarda bosib bajariladi.)

`son2` `2son` `talaba_ismi` `talaba ismi` `_temp` `int` `narx-1` `umumiySumma` `e@mail` `MAX_SIZE`

## To‘g‘ri identifikator

## Noto‘g‘ri identifikator

# =, +=, -=, *=, /=, %=

```cpp
int x = 20;    // oddiy o'zlashtirish
x += 10;       // x = x + 10   -> 30
x -= 5;        // x = x - 5    -> 25
x *= 2;        // x = x * 2    -> 50
x /= 3;        // x = x / 3    -> 16
x %= 5;        // x = x % 5    -> 1
cout << x;
```

▶ NATIJA

1

**`x += 10`**

`x = x + 10` ning qisqa yozuvi

**`x /= 3`**

butun sonlarda bo‘lish butun: `50 / 3 = 16`

**`x %= 5`**

qoldiq: `16 % 5 = 1`

**Diqqat!**

`=` — o‘zlashtirish, `==` — taqqoslash. `a == 5` “a 5 ga tengmi?” deb so‘raydi

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 3-bo‘lim yakuni: Identifikatorlar va o‘zlashtirish

1

`x += 10` ifodasi qaysi amalni anglatadi?

Savol 1

2

`talaba-ismi` nima uchun noto‘g‘ri identifikator?

Savol 2

3

`a = 5` va `a == 5` ning farqi nimada?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

# 2-topshiriq: Uch amal

Ikki butun son berilgan. Ularning yig‘indisi, ayirmasi va ko‘paytmasini toping.

**Kirish:** Bitta qatorda ikki butun son `a` va `b` (|a|, |b| ≤ 10000).

**Chiqish:** Uch qator: `a + b`, `a - b`, `a * b`.

| MISOL · KIRISH | MISOL · CHIQISH |
|---|---|
| `7 3` | `10`<br>`4`<br>`21` |

Telefon kamerasi bilan skanerlang

1. Saytga kiring (shaxsiy kod)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

04

# Inkrement, siljitish va bitli amallar

++ / -- · prefiks va postfiks · << va >> · &, |, ^, ~

++a / a++

<< >>

& | ^ ~

Ikkilik

# Prefiks va postfiks shakllar

| Shakl | Nima qiladi | Misol (a = 5) |
|---|---|---|
| `++a` **(prefiks)** | Avval oshiradi, keyin yangi qiymatni beradi | `x = ++a;` → x = 6, a = 6 |
| `a++` **(postfiks)** | Avval eski qiymatni beradi, keyin oshiradi | `x = a++;` → x = 5, a = 6 |
| `--a` **(prefiks)** | Avval kamaytiradi, keyin qiymatni beradi | `x = --a;` → x = 4, a = 4 |
| `a--` **(postfiks)** | Avval eski qiymatni beradi, keyin kamaytiradi | `x = a--;` → x = 5, a = 4 |

Alohida qatorda `i++;` va `++i;` farqsiz. Farq faqat ifoda ichida seziladi.

# << va >> — bitlarni siljitish

```
int a = 8;                // 0000 1000
int b = a << 2;           // 0010 0000 = 32
int c = a >> 1;           // 0000 0100 = 4
cout << b << " " << c << endl;
cout << (1 << 10) << endl;   // 2^10
cout << (13 >> 1) << endl;   // 13 / 2
```

▶ NATIJA

```
32 4
1024
6
```

**a << k**

chapga siljitish ≈ a × 2^k

**a >> k**

o‘ngga siljitish ≈ a / 2^k (butun qismi)

**1 << 10**

2 ning 10-darajasi = 1024

**13 >> 1**

13 / 2 = 6 (qoldiq tashlanadi)

BITLI OPERATORLAR

# &, |, ^, ~ — har bir bit alohida

| Operator | Nomi | Natija biti 1, agar... | 5 va 3 (0101, 0011) |
|---|---|---|---|
| & | bitli AND | ikkala bit ham 1 | 5 & 3 = 0001 = 1 |
| \| | bitli OR | kamida bitta bit 1 | 5 \| 3 = 0111 = 7 |
| ^ | bitli XOR | bitlar har xil | 5 ^ 3 = 0110 = 6 |
| ~ | bitli NOT | bit 0 bo‘lsa (teskari) | ~5 = -6 |

Adashtirmang: `&` va `|` — bitli; `&&` va `||` — mantiqiy operatorlar. `n & 1` — sonning toq/juftligini tekshiradi.

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 4-bo‘lim yakuni: Inkrement, siljitish va bitli amallar

1

`a = 5; x = a++;` dan keyin x va a nechaga teng?

Savol 1

2

`3 << 2` ning qiymati nechaga teng?

Savol 2

3

`6 & 3` natijasini ikkilik ko‘rinishda hisoblang.

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

# 3-topshiriq: Bitlarni siljitish

Butun son `a` va siljish miqdori `k` berilgan. `a << k` va `a >> k` qiymatlarini chiqaring.

**Kirish:** Ikki butun son: `a` ($0 \le a \le 1000000$) va `k` ($0 \le k \le 10$).

**Chiqish:** Ikki qator: `a << k` va `a >> k`.

| MISOL · KIRISH | MISOL · CHIQISH |
| --- | --- |
| 8 2 | 32<br>2 |

Telefon kamerasi bilan skanerlang

1. Saytga kiring (shaxsiy kod)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

05

# Ustuvorlik va tur o‘zgartirish

Amallar ketma-ketligi · butun bo‘lish tuzog‘i · (int)x va static_cast

| Ustuvorlik | 7 / 2 | static_cast | (double) |
|---|---|---|---|

AMALLAR KETMA-KETLIGI

# Qaysi amal birinchi bajariladi?

| Ustuvorlik | Operatorlar | Yo‘nalish |
|---|---|---|
| **1 (eng yuqori)** | `()`, `a++`, `a--` | chapdan o‘ngga |
| **2** | `++a`, `!`, `~`, unar `-`, `(tur)` | o‘ngdan chapga |
| **3** | `*`, `/`, `%` | chapdan o‘ngga |
| **4** | `+`, `-` | chapdan o‘ngga |
| **5** | `<<`, `>>` | chapdan o‘ngga |
| **6** | `< <= > >=`, keyin `== !=` | chapdan o‘ngga |
| **7** | `&`, keyin `^`, keyin `\|` | chapdan o‘ngga |
| **8** | `&&`, keyin `\|\|` | chapdan o‘ngga |
| **9 (eng past)** | `=`, `+=`, `-=` ... | o‘ngdan chapga |

`2 + 3 * 4 = 14`, lekin `(2 + 3) * 4 = 20`. Shubha bo‘lsa — qavs qo‘ying.

TYPE CASTING

# Ma’lumot turini o‘zgartirish

```cpp
int a = 7, b = 2;
cout << a / b << endl;          // butun bo'lish
cout << a % b << endl;          // qoldiq
cout << (double)a / b << endl;  // an'anaviy usul
cout << a / 2.0 << endl;        // yashirin o'zgartirish
cout << static_cast<int>(7.9) << endl;
cout << (char)('A' + 1) << endl;
```

▶ NATIJA

```
3
1
3.5
3.5
7
B
```

**`7 / 2 = 3`**

ikkala operand butun — kasr qismi tashlanadi

**`(double)a / b`**

bitta operand haqiqiy bo‘lsa, natija 3.5

**`static_cast<int>(7.9)`**

C++ uslubi; yaxlitlamaydi, kesadi: 7

**`'A' + 1`**

char → int: 66; `(char)66` → 'B'

▶ INTERAKTIV

# Natijani oldindan ayting

Har bir kod parchasi nima chiqarishini tanlang.  (Interaktiv slaydlarda bosib bajariladi.)

**1** **Natija qanday?**

```
cout << 2 + 3 * 4;
```

A) 20
B) 14
C) 24
D) 9

**2** **Natija qanday?**

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

A) 3.5
B) 4
C) 3
D) 3.0

**3** **Natija qanday?**

```
cout << static_cast<int>(9.99);
```

A) 10
B) 9
C) 9.99
D) 0

**4** **Natija qanday?**

```
int a = 5;
int x = a++ + 1;
cout << x << " " << a;
```

A) 7 6
B) 6 6
C) 6 5
D) 7 5

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 5-bo‘lim yakuni: Ustuvorlik va tur o‘zgartirish

1

`10 - 2 * 3` ning qiymati nechaga teng?

Savol 1

2

`static_cast<int>(7.9)` natijasi nima?

Savol 2

3

`7 / 2` 3.5 chiqishi uchun kodni qanday o‘zgartirasiz?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

# 4-topshiriq: Butun va haqiqiy bo‘lish

Ikki butun son `a` va `b` (b ≠ 0) berilgan. Butun bo‘lish natijasi, qoldiq va haqiqiy bo‘linmani chiqaring.

**Kirish:** Ikki butun son `a` va `b`, b ≠ 0, 1 ≤ a, b ≤ 100000.

**Chiqish:** Uch qator: `a / b` (butun), `a % b` va haqiqiy bo‘linma — verguldan keyin aniq 2 xona.

| MISOL · KIRISH | MISOL · CHIQISH |
| --- | --- |
| `7 2` | `3`<br>`1`<br>`3.50` |

Telefon kamerasi bilan skanerlang

1. Saytga kiring (shaxsiy kod)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

06

# Kutubxona va matematik funksiyalar

<iostream>, <cmath>, <string>, <iomanip> ... · sqrt, pow, round, sin

#include | <cmath> | pow() | Radian

# Tayyor vositalar qayerda?

| Kutubxona | Vazifasi | Misol funksiyalar |
|---|---|---|
| `<iostream>` | kiritish va chiqarish | `cin`, `cout`, `endl` |
| `<cmath>` | matematik funksiyalar | `sqrt`, `pow`, `sin`, `abs` |
| `<string>` | zamonaviy satrlar | `string`, `getline` |
| `<cstring>` | C-satrlar | `strlen`, `strcpy` |
| `<iomanip>` | chiqarishni formatlash | `setw`, `setprecision` |
| `<ctime>` | vaqt bilan ishlash | `time`, `ctime` |
| `<cstdlib>` | umumiy yordamchilar | `rand`, `exit` |
| `<algorithm>` | tayyor algoritmlar | `sort`, `max`, `min` |

Kutubxonadan foydalanish uchun uning sarlavha fayli `#include` bilan qo‘shiladi.

<CMATH>

# Asosiy matematik funksiyalar

| Funksiya | Vazifasi | Misol |
|---|---|---|
| `sqrt(x)` | kvadrat ildiz | `sqrt(16)` = 4 |
| `pow(a, b)` | a ning b-darajasi | `pow(2, 10)` = 1024 |
| `abs(x)` | modul | `abs(-3.5)` = 3.5 |
| `round(x)` | eng yaqin butunga yaxlitlash | `round(2.5)` = 3 |
| `floor(x)` / `ceil(x)` | pastga / yuqoriga yaxlitlash | `floor(2.7)` = 2, `ceil(2.1)` = 3 |
| `sin`, `cos`, `tan` | trigonometrik (radianda!) | `sin(PI / 2)` = 1 |
| `log(x)` / `log10(x)` | natural / o‘nlik logarifm | `log10(1000)` = 3 |

AMALIYOT

# Gipotenuza va burchak

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

int main() {
    const double PI = 3.1415926535897793;
    cout << pow(2, 3) << endl;
    cout << hypot(3, 4) << endl;
    cout << floor(-3.7) << " " << ceil(-3.7) << endl;
    cout << sin(30 * PI / 180) << endl;
    return 0;
}
```

▶ NATIJA

```
8
5
-4 -3
0.5
```

**pow(2, 3)**

$2^3 = 8$

**hypot(3, 4)**

`sqrt(3*3 + 4*4)` = 5

**floor(-3.7)**

pastga: -4; `ceil(-3.7)` yuqoriga: -3

**Gradus → radian**

`radian = gradus * PI / 180;` `sin(30)` 0.5 emas!

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 6-bo‘lim yakuni: Kutubxona va matematik funksiyalar

1

Sonni darajaga ko‘tarish uchun qaysi funksiya ishlatiladi?

Savol 1

2

`setw` va `setprecision` qaysi kutubxonada?

Savol 2

3

`sin(90)` nima uchun 1 chiqmaydi?

Savol 3

XULOSA

# Asosiy xulosalar

### 1 Til = sintaksis + semantika

Kompilyator sintaksisni tekshiradi va kodni mashina kodiga o‘giradi.

### 2 Dastur main() dan boshlanadi

`#include`, `using namespace std;`, `int main()`, `return 0;`

### 3 Preprotsessor — matn almashtirish

Makrolarni qavsga oling; konstantalar uchun `const` ishlating.

### 4 Nomlash qoidalari

Harf yoki `_` bilan boshlang; bo‘sh joy, belgilar, kalit so‘zlar — yo‘q.

### 5 Ustuvorlik va qavslar

`* / %` → `+ -`; shubha bo‘lsa, qavs qo‘ying.

### 6 7 / 2 = 3

Haqiqiy natija uchun `7 / 2.0` yoki `static_cast<double>`.

TAKRORLASH

# Yakunlashdan oldin sakkizta savol

1. Kompilyator nima ish qiladi?
2. `#include <cmath>` nima uchun kerak?
3. `#define KV(x) x*x` da `KV(2+1)` nechaga teng?
4. `_son1` va `1son` dan qaysi biri to‘g‘ri nom?
5. `a = 3; b = ++a * 2;` dan keyin b = ?
6. `5 | 2` nechaga teng?
7. `(double)9 / 2` natijasi?
8. `round(2.5)` va `floor(2.5)`?

LABORATORIYA ISHI

# 1-laboratoriya ishi: 6 ta dasturlash topshirig‘i

**5** **Kvadrat va kub**
Oson · 10 ball

**6** **Soat, daqiqa, sekund**
Oson · 10 ball

**7** **Gipotenuza va yuza**
O‘rta · 20 ball

**8** **Uch xonali son**
O‘rta · 20 ball

**9** **Doira**
O‘rta · 20 ball

**10** **Yaxlitlash laboratoriyasi**
Qiyin · 30 ball

Telefon kamerasi bilan skanerlang

Kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

O‘qituvchi kodingizni ko‘rib, baho qo‘yadi.

CHIQISH CHIPTASI

# Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

C++ dasturi manba koddan .exe gacha qaysi bosqichlardan o‘tadi?

2

```
int x = 7 / 2 * 2;
```

bo‘lsa, x nechaga teng va nima uchun?

3

Bugungi darsda sizga eng qiyin tuyulgan tushuncha qaysi?