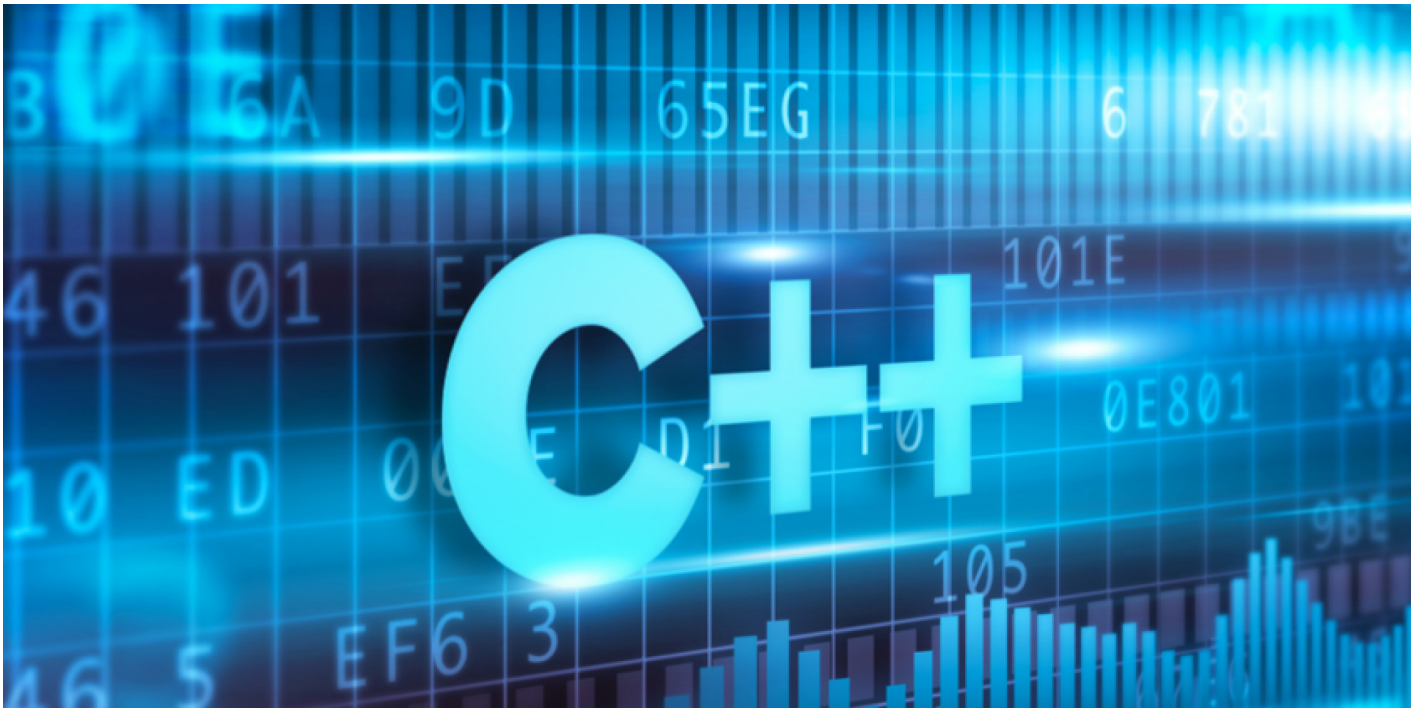
Funksiya elon qilish va funksiya tuzish ga doir masalalarni dasturini tuzishni o’rganish.



Funksiyalarni e’lon qilish va foydalanish

C++ tili boshqa dasturlash tillaridan faqat Funksiyalar bilan ishlay olishi bilan farqlanadi.

C++ tilida dastur ishlab chiqishda ***Funksiya*** – eng asosiy tushunchalardan biri sanaladi. CHunki, birinchidan, ihtiyoriy dastur hech bo’lmaganda ***main*** nomli bitta Funksiyani (asosiy Funksiya) o’z ichiga oladi. Aynan ana shu Funksiya dasturga kirish nuqtasini belgilab beradi. Dastur kodiga ***main*** Funksiyasidan tashqari bir qator yordamchi funkiyalar ham kirishi mumkin. Ularning barchasi ***main*** Funksiyasi yordamida boshqariladi. Bu Funksiyalarning barchasi global hisoblanadi.

Funksiyalar bilan ishlash imkoniyatiga ega bo’lish uchun, ularga birinchi marta murojaat qilinishidan avval aniqlangan bo’lishi lozim. Funksiyani aniqlashda unga murojaat qilinganda bajarilishi lozim bo’lgan amallar ketma-ketligi va bu amallarda ishtirok etayotgan o’zgaruvchilarning tiplari,

Funksiyaning nomi va u qayta uzatadigan ma’lumotning (natija) tipi, Funksiyaga kiruvchi rasmiy o’zgaruvchilarning tiplari ko’rsatiladi. Bunda Funksiyaning nomi va rasmiy o’zgaruvchilar ro’yxati Funksiyaning umumiy belgisi hisoblanadi va ana shu belgiga ko’ra Funksiyaga murojaat qilinadi.

Funksiyani aniqlash uning umumiy belgilari va jismini ko’rsatishdan iborat bo’lib, umumiy ko’rinishi quyidagicha:

Funksiya_tipi Funksiya_nomi (ro’yxat);

{

Funksiya_jismi

}

Bu yerda ***Funksiya_tipi***- Funksiya qaytaradigan ma’lumotning tipi (agar Funksiya ma’lumot qaytarmasa bu tip ***void*** bo’ladi); ***Funksiya_nomi*** – o’zgaruvchi-identifikatorni anglatadi. Bu nom boshqa o’zgaruvchilar kabi takrorlanmas bo’lishi lozim; ***ro’yxat*** – bo’sh yoki ***void*** bo’lishi yoki alohida ko’rsatiladigan rasmiy o’zgaruvchilarning tiplari va nomlarini o’z ichiga olishi mumkin; ***Funksiya_jismi*** – turli amallar va ko’rsatmalar ketma-ketligidan iborat bo’lib, odatda alohida olingan kichik bir masalani hal qilishga qaratiladi. Jismning so’nggi bajariladigan buyrug’i ***return*** bo’lib, u boshqaruvni Funksiyaga murojaat qilish nuqtasiga uzatish (qaytarish) amalini bajaradi. Bu buyruq umumiy ko’rinishda

return ifoda ;

yoki

return;

ko’rinishida yoziladi. ***return*** operatoridan keyin ko’rsatilgan ifoda yordamchi Funksiya tomonidan asosiy Funksiyaga uzatiladigan qiymatni belgilab beradi. Agar Funksiyaning hech qanday qiymatni qaytarishi ko’zda tutilmagan bo’lsa (ya’ni Funksiya ***void*** tipida bo’lsa) ***return*** opreatoridan keyin ifoda yozilmaydi. Bitta Funksiya jismida bir nechta ***return*** buyruqlaridan foydalanish mumkin. S++ tilida agar dastur o’z

ishini muvaffaqiyatli yakunlasa, 0 qiymatini qaytarish qabul qilingan.

Funksiyalarni e’lon qilishga namunalar keltiramiz:

void print (char x, int y) // Hech narsa qaytarilmaydi

{

cout **«"*n*" « *x* « *y;*** **// return** tushirib qoldirilgan

}

float min(float a, float b); // Funksiya natijasi ***float*** tipida

{ ***if a<b) return a;*** // a va b sonlaridan kichigini qaytaradi **}**

Funksiyaga murojaat qilinganda rasmiy o’zgaruvchilar joriy o’zgaruvchilar bilan almashtiriladi va bunda tiplarining o’zaro mosligi qat’iy nazorat qilinadi. Zarur bo’lganda (moslik bo’lmaganda) o’zgaruvchilarning tiplari to’g’ridan-to’g’ri almashti-rilishi ko’zda tutilgan. S++ haqida fikr yuritilganda, “tiplarning qat’iy mosligi”ga alohida e’tibor beriladi. SHunga ko’ra, rasmiy va joriy o’zgaruvchilarning tiplarining o’zaro mosligi kompilyatsiya qilish jarayonidayoq tekshiriladi.

Funksiyaga murojaat qilish (soddaroq aytganda, uni chaqirish) oddiy qavslar yordamida amalga oshiriladi. Qavslar ichida esa joriy o’zgaruvchilar ro’yxati ko’rsatiladi:

Funksiya_nomi (joriy o’zgaruvchilar ro’yxati**);**

Funksiyaga murojaat natijasida u qaytaradigan va tipi tipiga mos bo’lgan qiymatga ega bo’linadi.

Joriy o’zgaruvchilar (Funksiyaning argumentlari) – va rasmiy o’zgaruvchilar o’rtasidagi moslik rasmiy va joriy o’zgaruvchilarning ro’yxatdagi o’rniga ko’ra aniqlanadi.

Joriy o’zgaruvchilar murojaat qiluvchi dastur tomonidan uzatiladi va Funksiya jismidagi ko’rsatmalarni bajarishda aynan an shu

o’zgaruvchilarning qiymatlaridan foydalaniladi.

SHunday qilib, joriy o’zgaruvchilar ro’yxati yoki bo’sh, yoki ***void***, yoki vergul bilan ajratilgan o’zgaruvchilar ro’yxatidan iborat bo’lishi mumkin.

YUqoridagi fikrlarni amaliyotga tatbiq etishga urinib ko’raylik.

Masala-1: Haqiqiy *a*, *b, s* va *d* sonlari berilgan bo’lsin. Ularning eng kichigini toping.

Echish g’oyasi. Dastlab, *a* va *b* sonlarining eng kichigini aniqlaymiz va uni *p* bilan belgilaymiz. So’ngra, *c* va *d* larning eng kichigini *q* bilan belgilaymiz. Ishning yakunida *p* va *q* larning eng kichigini topish qoladi halos. Ko’rinib turibdiki, ikki sonning eng kichigini topish masalasidan uch marta foydalanilmoqda. SHuning uchun, rasmiy o’zgaruvchi sifatida *n* va *m* sonlarni tanlab olib, ularning kichigini topish uchun Funksiya tashkil qilamiz. Bu mulohazalani e’tiborga olib, kodni quyidagicha tashkil qilamiz:

```
# include ;

float min(float n, float m)

{

float k;

if (n<m) k=n; else k=m;

return k;

}

void main()

{ float a, b, c, d;

cin >>a>>b>>c>>d;

float p=min(a, b);

float q=min(c, d);
```

float kichik=min(p, q);

cout<<kichik;

return;

}

Eslatma: ***return*** buyrug’idan bir necha marta foydalanish mumkin bo’lgani uchun, Funksiyani

float min(float n, float m)

{

if (n<m) return n; else return m;

}

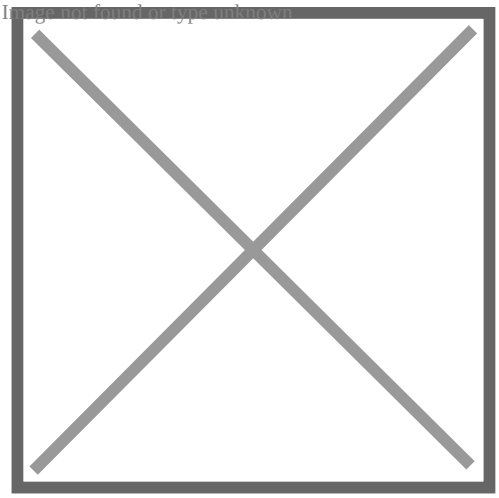
ko’rinishida ham yozish mumkin.

S++ tilida bitta Funksiya tarkibida boshqa Funksiyalarga ham murojaat qilish mumkin.

Masala-1: Uchburchak uchlarining koordinatalari (***x1,y1***), (***x2,y2***), (***x3,y3***) berilgan bo’lsin. (***x, u***) koordinatali nuqta shu uchburchakka tegishli bo’la oladimi ?

Echish g’oyasi: Berilgan uchburchak yuzasi S bo’lsin. Uchburchak uchlarini (***x, u***) koordinatali nuqta bilan tutashtirib, 3 ta uchburchak hosil qilamiz. Ularning yuzalari mos ravishda ***S1, S2*** va ***S3*** bo’lsin. Agar (***x, u***) nuqta berilgan uchburchak ichida yotsa,

Image not found or type unknown



S=S1+S2+S3

bo’ladi. Aks holda nuqta uchburchak ichida yotmaydi.

Ko’rinib turibdiki, bu yerda uchlarining koordinatalari berilgan to’rtta uchburchak yuzini hisoblashga to’g’ri kelmoqda. Bu koordinatalar uchun formal o’zgaruvchilarni (***a1, b1***), (***a2, b2***) va (***a3, b3***) tarzida tanlash mumkin. Uchburchakning tomonlari va yarim perimetrini belgilash uchun mos ravishda ***A, V, S*** va ***R*** o’zgaruvchilar olinadi. Ular masala shartida ko’rsatilmagani uchun lokal o’zgaruvchilar hisoblanadi.

Tanlangan formal o’zgaruvchilarni hisobga olib, uchburchak yuzini topish buyruqlaridan protsedura hosil qilinadi. So’ngra ketma-ket to’rt marta (***x1, x2, x3, y1, y2, y3***), (***x, x1, x2, y, y1, y2***), (***x, x2, x3, u, u2, u3***) va (***x, x1, x3, y, y1, y3***) joriy o’zgaruvchilar uchun bu protseduraga murojaat qilinadi, ya’ni uchlari ana shu nuqtalarda yotgan uchburchaklariing ***S1, S2, S3*** va ***S*** yuzalari hisoblanadi. So’ng masalani yechish g’oyasiga ko’ra, ***S=S1+S2+S3*** munosabatning qiymatiga qarab hulosa chiqariladi.

```
# include ;

# include ;

float kesma(float a1, float b1, float a2, float b2)

{

float k=sqrt((a2-a1)*(a2-a1)+(b2-b1)*(b2+b1));

return k;
```

```
}
loat yuza(float a1, float b1, float a2, float b2, float a3, float b3)
{
float n1=kesma(a1,b1,a2,b2);
float n2=kesma(a1,b1,a3,b3);
float n3=kesma(a2,b2,a3,b3);
float p=(n1+n2+n3)/2;
float yuz=sqrt(p*(p-n1)*(p-n2)*(p-n3));
return yuz;
}
void main()
{
float x, x1, x2, x3, y, y1, y2, y3;
cin >>x>>y;
cin >>x1>>x2>>x3>>y1>>y2>>y3;
float s1=yuza(x, y, x1, y1, x2, y2);
float s2=yuza(x, y, x1, y1, x3, y3);
float s3=yuza(x, y, x2, y2, x3, y3);
float s=yuza(x1, y1, x2, y2, x3, y3);
if (s==s1+s2+s3) cout <<"Xa"; else cout <<"Yo'q";
return;
}
```

SAVOL VA TOPSHIRIQLAR

1. Formal va joriy o’zgaruvchilarning farqini tushuntiring.
2. Oraliq o’zgaruvchi nima va undan qachon foydalaniladi?
3. Funksiyalar qachon va qanday tashkil qilinadi?
4. Funksiyalarning umumiy tarkibini aytib bering?
5. Funksiyalardan qanday foydalanish mumkin?
6. Funksiyalarga qanday murojaat qilish mumkin?
7. Quyidagi masalalar uchun dastur ishlab chiqing.