



Ի՞նչ է Constant:

«Constant» հիմնարար հասկացություն է, որը ծառայում է ծրագրի կատարման ընթացքում արժեքի անփոփոխ պահպանմանը: Դա մի միջոց է հայտարարելու փոփոխական արժեքով, որը չի կարող փոփոխվել նշանակվելուց հետո: Կողի հստակության, պահպանման և որոշակի արժեքների պահպանումը հաստատուն և կանխատեսելի լինելու ապահովման համար կարևոր է կողի հստակությունը:

Օրինակ:

```
const int MAX_VALUE = 100;
```

Այս օրինակում MAX_VALUE-ը int տիպի հաստատուն է, և դրա արժեքը սահմանված է 100: Ծրագրում հետագայում այս արժեքը փոխելու ցանկացած փորձ կհանգեցնի կոմպիլյացիայի սխալի:

Ինչպես են աշխատում Constant-ը:

Կոնստանտները սկզբնավորվում են հայտարարագրման պահին և չեն թույլատրվում փոփոխել ծրագրի ողջ ընթացքում: Այս անփոփոխությունն ապահովում է մի քանի առավելություններ.

- Կանխատեսելիություն:** Մշտականները ապահովում են, որ որոշ արժեքներ մնան ֆիքսված՝ նպաստելով կանխատեսելիությանը և կանխելով անսպասելի փոփոխությունները:
- Ընթերցանություն և ըմբռնում.** Constants ծածկագիրը ավելի ընթեռնելի և ինքնաբացատրելի է դարձնում՝ ֆիքսված արժեքներին նշանակալից անուններ տալով:
- Անվտանգություն և վրիպագերծում.** Մշտականները օգնում են կանխել կրիտիկական արժեքների պատահական փոփոխությունները, բարելավելով ծրագրի կայունությունը և հեշտացնելով վրիպագերծումը:
- Compile-ժամանակի գնահատում.** Կոնստանտները կարող են գնահատվել կոմպիլյացիայի ժամանակ, ինչը կարող է օպտիմալացնել կոդը և բարելավել կատարումը:

C++-ում constant-ների տեսակները:

C++-ն աջակցում է constant-ների մի քանի տեսակների.

Բառացի հաստատուններ. սրանք բառացի արժեքներ են, ինչպիսիք են ամբողջ թվերը, լողացող կետով թվերը, նիշերը և տողերը: Օրինակ:

```
const int INTEGER_CONSTANT = 42;  
const double PI = 3.14;  
const char NEWLINE = '\\n';  
const std::string GREETING = "Hello, world!";
```

Անվանված Constant. դրանք անուններով հաստատուններ են, որոնց տրվում են իմաստալից նույնացուցիչներ: Դրանք հայտարարագրվում են՝ օգտագործելով const բանալի բառը: Օրինակ:

```
const int MAX_VALUE = 100;
```

Թվարկման Constant. Թվարկումները թույլ են տալիս ստեղծել անվանական հաստատուններ որոշակի շրջանակում: Օրինակ:

```
enum Color { RED = 1, GREEN = 2, BLUE = 3 };
```

Մակրո Constant. Նախապրոցեսորային մակրոները օգտագործվում են հաստատունները սահմանելու համար՝ օգտագործելով #define: Օրինակ:

```
#define MAX_VALUE 100
```

Օգտագործման դեպքեր և ինչու են դրանք անհրաժեշտ:

Constant կիրառություն են գտնում տարբեր սցենարներում.

1. **Կախարդական համարների փոխարինում.** Constant փոխարինում են «կախարդական թվերը» (կոշտ կոդավորված թվային արժեքներ) իմաստալից անուններով, ինչը մեծացնում է կոդերի ընթերցանությունը և պահպանելիությունը: Օրինակ, 3.14-ը PI-ով փոխարինելը կոդը ավելի հասկանալի է դարձնում:
2. **Կազմաձևման կարգավորումներ.** հաստատուններն օգտագործվում են կազմաձևման պարամետրերը պահելու համար, ինչպիսիք են ժամանակի

արժեքները, բուժերի չափերը կամ ցանկացած այլ պարամետր, որը կարող է ճշգրտման կարիք ունենալ, բայց ծրագրի կատարման ընթացքում մնում է անփոփոխ:

3. **Մաթեմատիկական և ֆիզիկական Constant.** Գիտական կամ մաթեմատիկական կիրառություններում π (pi) նման Constant կամ լույսի արագությունը հայտարարվում են որպես Constant ճշգրիտ հաշվարկների համար:
4. **Սխալներ կոդում.** Constant կարող են օգտագործվել սխալի կոդերը կամ կարգավիճակները ներկայացնելու համար՝ բարելավելով սխալների մշակումը և վրիպագերծումը:

Ամփոփելով, Constant-ը C++-ում ապահովում են որոշակի արժեքների անփոփոխելիությունն ապահովելու միջոց՝ նպաստելով կոդի անվտանգությանը, ընթեռնելիությանը և պահպանմանը: Նրանք վճռորոշ դեր են խաղում ժամանակակից ծրագրավորման մեջ՝ խթանելով լավ կոդավորման պրակտիկան և բարելավելով ծրագրային ապահովման ընդհանուր որակը: