

11 ஆம் வகுப்பு – கணினி அறிவியல்

**பாடம் 11.**

**C++ - ன் செயற்கூறுகள்**

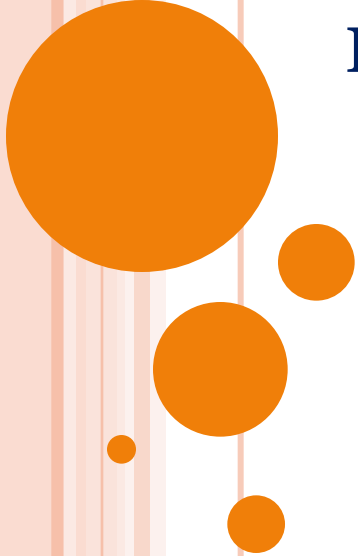
Prepared by,

**J. KAVITHA, B.Sc,B.Ed,M.C.A,M.Phil.,**

**Computer Instructor Gr - I,**

**GHSS, S.S.KULAM,**

**Coimbatore.**



# கற்றலின் நோக்கங்கள்

- செயற்கூறுகளின் வரையறை மற்றும் செயற்கூறுகளின் பயன்களைப் பற்றி அறிந்து கொள்ளுதல்.
- முன்னரேவரையறுக்கப்பட்டமற்றும் பயனர் வரையறுக்கும் செயற்கூறுகளை பற்றி அறிந்து கொள்ளுதல்.
- சிக்கல்களைத் தீர்க்க கணித செயற்கூறுகளைப் பயன்படுத்துதல்.
- சரம் மற்றும் குறியுறு செயற்கூறுகளைப் பயன்படுத்தி சரம் மற்றும் குறியுறு தரவுகளை கையாளுதல்.
- பெரிய சிக்கல்களை சிறு சிறு செயற்கூறுகளாக உருவாக்கி அவற்றை செயல்படுத்தும் நிரலாக்கம் (Modular Programming) பற்றி அறிந்து கொள்ளுதல்.
- மாறிகள் மற்றும் செயற்கூறுகளின் வரையெல்லையைக் கண்டறிதல்.



# அறிமுகம்

## செயற்கூறுகள்:

- ஒரு பெரிய நிரலை சிறிய துணை நிரலாக பிரிக்க முடியும். அவ்வாறு பிரிக்கப்படும் துணை நிரல்கள் செயற்கூறு என்று அழைக்கப்படுகிறது.

## செயற்கூறுவின் பயன்கள்:

- நிரலின் நீளத்தையும் மற்றும் சிக்கற்பாட்டையும் குறைக்கிறது.
- நிரலை எளிதில் புரிந்து கொள்ளவும், பிழைகளைக் கண்டறிந்து திருத்தவும் செயற்கூறுகள் வழிவகுக்கிறது.
- செயற்கூறுகளை மறுபயனாக்கம் செய்யவும் முடியும்.

## செயற்கூறுகளின் வகைகள்:

- செயற்கூறுகளை இரு வகைகளாக பிரிக்கலாம்.
  - முன் வரையறுக்கப்பட்ட (அல்லது) உள்ளமைந்த (அல்லது) நூலக செயற்கூறுகள்
  - பயனர் வரையறுக்கப்பட்ட செயற்கூறுகள்



# C++ -ல் உள்ள தலைப்பு கோப்புகள் மற்றும் உள்ளமைந்த செயற்கூறுகள்

- நூலக செயற்கூறுகளுக்கு தேவையான செயற்கூறு முன்மாதிரி மற்றும் வரையறுப்புகளை தலைப்புக் கோப்புகள் கொண்டுள்ளது. தலைப்பு கோப்பின் விரிவாக்கம் .h என்று அறியப்படும்.

- ஒரு தலைப்புக்கோப்பில் பல உள்ளிணைந்த செயற்கூறுகளைப் பற்றிய வரையறுப்புகள் கொண்டிருக்கும்.

உதாரணமாக: `stdio.h` என்ற தலைப்பு கோப்பில் உள்ளீட்டு / வெளியீட்டிற்கான செயற்கூறுகளை பற்றி முன்னரே வரையறுக்கப்பட்டுள்ளது.



## உள்ளீடு / வெளியீடு (STDIO.H)

- o **stdio.h** தலைப்பு கோப்பில் உள்ளீடு / வெளியீடு (I/O) செயற்கூறுகளான `getchar()`, `putchar()`, `gets()` மற்றும் `puts()` போன்றவற்றை முன்னரே வரையறுக்கப்பட்டுள்ளது.

### **getchar() மற்றும் putchar() செயற்கூறுகள்:**

- o முன்னரே வரையறுக்கப்பட்ட `getchar()` என்ற செயற்கூறினைப் பயன்படுத்தி விசைப்பலகையின் மூலம் ஒரு எழுத்தையும் உள்ளீடு செய்யலாம் மற்றும் `putchar()` என்ற செயற்கூறின் மூலம் அந்த எழுத்தை வெளியீடும் செய்யலாம்.

### **gets() மற்றும் puts() செயற்கூறுகள்:**

- o `gets()` செயற்கூறின் ஒரு சரத்தை உள்ளீடு செய்து அதை சரத்திற்கான மாறியில் சேமித்து வைக்கலாம். `puts()` செயற்கூறு, `gets()` செயற்கூறின் மூலம் உள்ளீடாக பெற்ற சரத்தை ஒரு புதிய வரியில் வெளியிடச் செய்யும்.



# எடுத்துகாட்டு நிரல் - 1

C++ மொழியில் ஒரு எழுத்தை உள்ளிடவும் மற்றும் வெளியிடுவதற்கான நிரல்.

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main()
{
cout<<"\n Type a Character : ";
char ch = getchar();
cout << "\n The entered Character is: ";
putchar(ch);
return 0;
}
```

வெளியீடு:

Type a Character : T

The entered Character is: T



## எடுத்துகாட்டு நிரல் - 2

சரத்தை உள்ளீடு மற்றும் வெளியீடு செய்வதற்கான  
C++ நிரல்:

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main()
{
char str[50];
cout<<"Enter a string : ";
gets(str);
cout<<"You entered: "<< puts(str);
return(o);
}
```

வெளியீடு:

Enter a string : Computer Science

You entered: Computer Science



## குறியுறு செயற்கூறுகள் (CTYPE.H)

- இந்த தலைப்பு கோப்பில் குறியுறுக்கு தேவையான பல்வேறு செயல்பாடுகள் வரையறுக்கப்பட்டுள்ளது.
- **isalnum():** isalnum() செயற்கூறு உள்ளீடு செய்யப்பட்ட குறியுறு ஆங்கில எழுத்தா அல்லது எண்ணா எனக் கண்டறியலாம்.
- **isalpha():** isalpha() செயற்கூறு உள்ளீடு செய்யப்பட்ட குறியுறு ஆங்கில எழுத்தாக உள்ளதா, இல்லையா என்பதை சரிபார்க்கப் பயன்படுகிறது.
- **isdigit():** உள்ளீடு செய்யப்பட்டுள்ள குறியுறு எண்ணாக உள்ளதா அல்லது இல்லையா என்பதைச் சரிபார்க்க இந்த செயற்கூறு பயன்படுகிறது.





## எடுத்துக்காட்டு நிரல் - 1

```
#include<iostream>
#include<stdio.h>
#include<ctype.h>
using namespace std;
int main()
{
char ch;
int r;
cout<<"\n Type a Character :";
ch = getchar();
r = isalnum(ch);
cout<<"\nThe Return Value of isalnum(ch) is :"<<r;
}
```

வெளியீடு-1: Type a Character :A

The Return Value of isalnum(ch) is :1

வெளியீடு-2: Type a Character :?

The Return Value of isalnum(ch) is : 0



## எடுத்துக்காட்டு நிரல் - 2

```
#include<iostream>
#include<stdio.h>
#include<ctype.h>
using namespace std;
int main()
{
char ch;
cout << "\n Enter a charater: ";
ch = getchar();
cout<<"\n The Return Value of isalpha(ch) is :" << isalpha(ch) ;
}
```

வெளியீடு -1: Enter a charater: A  
The Return Value of isalpha(ch) is :1

வெளியீடு - 2: Enter a charater: 7  
The Return Value of isalpha(ch) is :0



## எடுத்துகாட்டு நிரல் - 3

```
using namespace std;
#include<iostream>
#include<ctype.h>
int main()
{
char ch;
cout << "\n Enter a Character: ";
cin >> ch;
cout<<"\n The Return Value of isdigit(ch) is :" << isdigit(ch) ;
}
```

வெளியீடு -1 Enter a Character: 3  
The Return Value of isdigit(ch) is :1

வெளியீடு -2 Enter a Character: A  
The Return Value of isdigit(ch) is :0



## குறியறு செயற்கூறுகள் (CTYPE.H)

- **islower():** இந்த செயற்கூறு உள்ளீடு செய்யப்பட்ட குறியறு எழுத்து ஆங்கில சிறிய எழுத்தாக உள்ளதா அல்லது இல்லையா என்று சரிபார்க்கும்.
- **isupper():** உள்ளீடு செய்யப்பட்டுள்ள குறியறு எழுத்து ஆங்கில பெரிய எழுத்தாக உள்ளதா என்று சரிபார்க்க இந்த செயற்கூறு பயன்படும்.
- **toupper():** உள்ளீடு செய்யப்பட்டுள்ள குறியறு எழுத்து ஆங்கில பெரிய எழுத்தாக மாற்ற இந்த செயற்கூறு பயன்படுகிறது.
- **tolower():** உள்ளீடு செய்யப்பட்டுள்ள குறியறு எழுத்தை ஆங்கில சிறிய எழுத்தாக மாற்ற இந்த செயற்கூறு பயன்படுகிறது.



## சரங்களை கையாளுதல் (STRING.H)

- **string.h** (**cstring**) என்ற நூலக கோப்பில் குறியீடுகளின் அணியில் உள்ள சரங்களைக் கையாளுவதற்கு என பல்வேறு செயற்கூறுகள் உள்ளன .
- **strcpy()**: இந்த செயற்கூறு மூலத்திலுள்ள சரங்களை இலக்கு சரத்தின் நினைவகத்தில் நகல் எடுக்கும்.
- **strlen()**: **strlen()** என்ற செயற்கூறு மூல சரத்தை அதன் செயலுருபாக எடுத்துக் கொண்டு அதன் நீளத்தை திருப்பி அனுப்பும்.
- **strcmp()**: இந்த செயற்கூறு **string1** மற்றும் **string2** உள்ளடக்கத்தை அகரவரிசையில் ஒப்பீடு செய்யும்.
- **strcat()**: இந்த செயற்கூறு மூலசரத்தின் நகலை இலக்கு சரத்தின் இறுதியில் இணைக்கும்.
- **strupr()**: **strupr()** செயற்கூறு உள்ளீடாக கொடுக்கப்பட்டுள்ள சரத்தை ஆங்கில பெரிய எழுத்துக்களாக மாற்றும்.
- **strlwr()**: **strlwr()** செயற்கூறு உள்ளீடாக கொடுக்கப்பட்டுள்ள சரத்தை ஆங்கில சிறிய எழுத்துக்களாக மாற்றும்.



## எடுத்துகாட்டு நிரல் - 1

```
#include <string.h>
#include <iostream>
using namespace std;
int main()
{
char source[] = "Computer Science";
char target[20]="target";
cout<<"\n String in Source Before Copied :"<<source;
cout<<"\n String in Target Before Copied :"<<target;
strcpy(target,source);
cout<<"\n String in Target After strcpy function Executed :"<<target;
return 0;
}
```

வெளியீடு:

String in Source Before Copied :Computer Science

String in Target Before Copied :target

String in Target After strcpy function Executed :Computer Science



## எடுத்துக்காட்டு நிரல் - 2

```
#include <string.h>
#include <iostream>
using namespace std;
int main()
{
char source[ ] = "Computer Science";
cout<<"\nGiven String is "<<source<<"its Length is "<<strlen(source);
return 0;
}
```

வெளியீடு:

Given String is Computer Science its Length is 16



## எடுத்துக்காட்டு நிரல் - 3

```
#include <string.h>
#include <iostream>
using namespace std;
int main()
{
char string1[] = "Computer";
char string2[] = "Science";
int result;
result = strcmp(string1,string2);
if(result==0)
{
cout<<"String1 : "<<string1<<" and String2 : "<<string2 <<"Are Equal";
}
if (result<0)
{
cout<<"String1 : "<<string1<<" and String2 : "<<string2 <<" Are Not Equal";
}
}
```

வெளியீடு: String1 : Computer and String2 : Science Are Not Equal



## எடுத்துக்காட்டு நிரல் - 4

```
#include <string.h>
#include <iostream>
using namespace std;
int main()
{
char target[50] = "Learning C++ is fun";
char source[50] = " , easy and Very useful";
strcat(target, source);
cout << target ;
return 0;
}
```

வெளியீடு: Learning C++ is fun , easy and Very useful



## கணித செயற்கூறுகள் (MATH.H)

- math.h என்ற தலைப்பு கோப்பு பெரும்பாலான அடிப்படை கணித செயற்கூறுகளை உள்ளடக்கி உள்ளது.
- **cos()** செயற்கூறு: cos() செயற்கூறு ஒரு செயலுருபுவின் மதிப்பு ரேடியன்ஸ் ஆக(Radians) எடுத்துக்கொள்ளும்.
- **sin()** செயற்கூறு: sin() செயற்கூறு ஒரே செயலுருபுவின் மதிப்பை ரேடியன்ஸில் ஏற்கும்.
- **sqrt() function:** sqrt() செயற்கூறு உள்ளீடப்பட்ட செயலுருபின் மதிப்பிற்கான வர்க்க மூலத்தைத் திருப்பி அனுப்பும்.
- **pow()** செயற்கூறு: pow() செயற்கூறு அடித்தள(base) செயலுருபின் மேல் அடுக்குக்குறி(exponent) மதிப்பைத் திருப்பி அனுப்பும்.



## எடுத்துகாட்டு நிரல் - 1

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
double x = 0.5, result;
result = cos(x);
cout << "COS("<<x<<" )= "<<result;
}
```

வெளியீடு:

COS(0.5)= 0.877583



## எடுத்துகாட்டு நிரல் - 2

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
double x = 625, result;
result = sqrt(x);
cout << "sqrt("<<x<<" ) = "<<result;
return 0;
}
```

வெளியீடு:

$\text{sqrt}(625) = 25$



# செயற்கூறை வரையறுத்தல்

- C++-ல், ஒரு செயற்கூறை நிரலில் ஏதேனும் ஒரு பகுதியில் அதைப் பயன்படுத்தும் முன் வரையறுக்கப்பட வேண்டும்

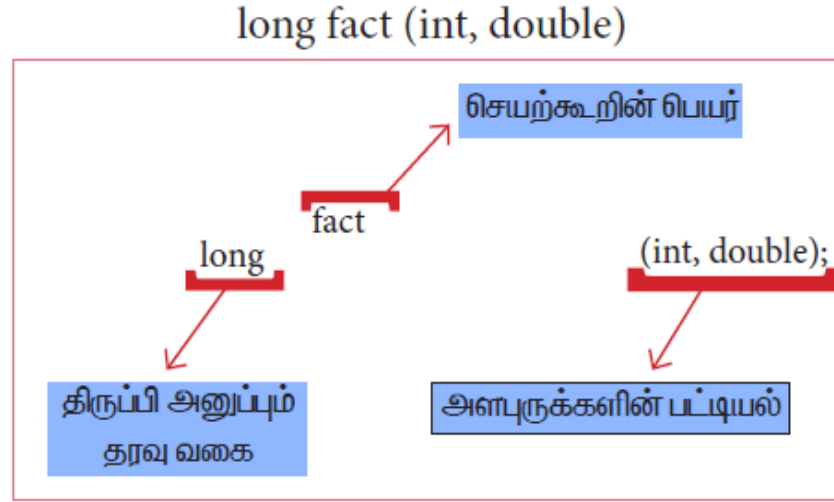
செயற்கூறு வரையறுப்பதற்கான தொடரியல்:

```
திருப்பு அனுப்பும் தரவு வகை செயற்கூறின்_பெயர்  
(அளபுருக்களின் பட்டியல்)  
{  
செயற்கூறின் உடற்பகுதி  
}
```



# செயற்கூறின் முன்வடிவு

எடுத்துக்காட்டு: long fact(int, double)



செயற்கூறின் முன்வடிவம் கீழே கொடுக்கப்பட்டுள்ள தகவல்களை நிரல்பெயர்ப்பிக்கு கொடுக்கும்.

- செயற்கூறு திருப்பி அனுப்பும் மதிப்பு long
- செயற்கூறின் பெயர் fact என்பதாகும்.
- இந்த செயற்கூறு இரண்டு செயலுருபுக்களை ஏற்கும்.



## VOID கட்டளையின் பயன்

void தரவினம் இரண்டு முக்கிய நோக்கங்களைக் கொண்டது:

- இந்த செயற்கூறு எந்த மதிப்பையும் திருப்பி அனுப்பாது என்பதைக் குறிக்க.
- பொது இனச் சுட்டியை (generic pointer) அறிவிக்க.



# செயற்கூற்றை செயல்படுத்துதல்

- பயனர் வரையறுத்த செயற்கூறை செயல்படுத்த அதன் பெயர் மற்றும் தேவையான செயலுருபுக்களை கொண்டு அழைக்க வேண்டும்.
- செயற்கூறை அழைக்கப்படும் போது நிரல்பெயர்ப்பி செயற்கூறின் முன்வடிவத்தை பயன்படுத்தி செயற்கூறு சரியாக அழைக்கப்பட்டுள்ளதா என்று சரிபார்க்கும்.
- முன்வடிவில் உள்ள செயலுருபின் தரவு வகையும் அழைப்புக் கூற்றில் உள்ள செயலுருபுக்களின் தரவு வகையும் பொருத்தமாக இல்லையெனில், தரவு வகை மாற்றம் தானாகவே செய்ய முடியும் என்றால் நிரல்பெயர்ப்பி அதை செய்யும், இல்லையெனில் நிரல்பெயர்ப்பி இதற்கு பிழை அறிக்கை அறிவிக்கும்.





முறையான அளபுருக்கள் (FORMAL PARAMETERS) மற்றும்  
மெய்யான அளபுருக்கள் (ACTUAL PARAMETERS) அல்லது  
செயலுருபுக்கள் (ARGUMENTS)

- செயலுருபுக்கள் அல்லது அளபுருக்கள் மூலமாக அழைக்கும் செயற்கூறிலிருந்து அழைக்கப்படும் செயற்கூறுக்கு மதிப்புகள் பரிமாற்றம் செய்யப்படும்.
- வரையறுக்கப்பட்ட செயற்கூறில் மாறிகளாக பயன்படுத்தப்பட்டுள்ள அளபுருக்கள் முறையான அளபுருக்கள் என்றழைக்கப்படும்.
- அழைப்பு செயற்கூறில் உள்ள மாறிலிகள் அல்லது மாறிகள் அல்லது கோவைகள் மெய்யான அளபுருக்கள் என்றழைக்கப்படும்.



# எடுத்துக்காட்டு

```
#include <iostream>
```

```
using namespace std;
```

```
int sum (int x, int y)
```

செயற்கூறு

```
{  
}  
}
```

```
return (x + y);
```

முறையான அளபுருக்கள்

```
int main ()
```

Main செயற்கூறு

```
{  
}  
}
```

```
int a,b ;
```

```
cout<<"\n Enter Number 1:";
```

```
cin>>a;
```

```
cout<<"\n Enter Number 2:";
```

```
cin >>b;
```

```
cout<<"\n The sum = "<<sum (a,b);
```

மெய்யான அளபுருக்கள்



## முன்னியல்புச் செயலுருபுக்கள் (DEFAULT ARGUMENT)

- C++ மொழியில் ஒரு செயற்கூற்றின் முன்வடிவில் உள்ள முறையான அளபுருக்களில் முன்னியல்பு மதிப்புகளை இருத்தி வைக்க முடியும்.
- செயற்கூற்றை அழைக்கும் போது முன்னியல்பு செயலுருபு சில மதிப்புகளைத் தவிர்க்க வழிவகுக்கும்.
- செயற்கூற்றை அழைக்கும் போது ஏதேனும் செயலுருபுகளுக்கு மதிப்பு கொடுக்காவிடில் நிரல்பெயர்ப்பி முன்னியல்பு செயலுருபுக்களின் மதிப்பைகளை அழைக்கப்பட்ட செயற்கூற்றிற்கு ஏற்கும்.

எடுத்துக்காட்டு:

```
void defaultvalue(int n1=10, n2=100);
```



## செயற்கூற்றை அழைப்பதற்கான வழிமுறைகள்

### மதிப்பு மூலம் அழைத்தல் முறை:

- மதிப்பு மூலம் அழைத்தல் முறையில் மெய்யான அளபுருவின் மதிப்பை முறையான அளபுருவில் நகலெடுக்கப்படும்.
- முறையான அளபுருவின் மதிப்பில் ஏதேனும் மாற்றங்கள் செய்தால் அது மெய்யான அளபுருவின் மதிப்பில் பிரதிபலிப்பதில்லை.



## எடுத்துக்காட்டு நிரல் (மதிப்பு மூலம் அழைத்தல் முறை)

```
#include<iostream>
using namespace std;
void display(int x)
{
int a=x*x;
cout<<"\n\nThe Value inside display function (a * a):"<<a;
}
int main()
{
int a,b;
cout<<"\nExample : Function call by value:";
cout<<"\n\nEnter the Value for A :";
cin>>a;
display(a);
cout<<"\n\nThe Value inside main function "<<a;
return(0);
}
```

**வெளியீடு** : Example : Function call by value

Enter the Value for A : 5

The Value inside display function (a \* a) : 25

The Value inside main function 5



## செயற்கூற்றை அழைப்பதற்கான வழிமுறைகள்

### குறிப்பு மூலம் அழைத்தல் முறை:

- இந்த முறையில் மெய்யான அளபுருவின் குறிப்பை அல்லது முகவரியை முறையான அளபுருவில் நகலெடுக்கும்.
- முகவரியின் மூலம் அழைப்பதால் முறையான அளபுருவின் மதிப்பில் ஏதேனும் மாற்றம் செய்தால் மெய்யான அளபுருவில் அந்த மாற்றம் பிரதிபலிக்கும்.



## எடுத்துக்காட்டு நிரல் (குறிப்பு மூலம் அழைத்தல் முறை)

```
#include<iostream>
using namespace std;
void display(int &x) //passing address of a//
{
x=x*x;
cout<<"\n\nThe Value inside display function (n1 x n1) : "<<x ;
}
int main()
{
int n1;
cout<<"\nEnter the Value for N1 :";
cin>>n1;
cout<<"\nThe Value of N1 is inside main function Before passing : "<< n1;
display(n1);cout<<"\nThe Value of N1 is inside main function After passing (n1 x
n1) : "<< n1; return(o);
}
```

வெளியீடு : Enter the Value for N1 :45

The Value of N1 is inside main function Before passing : 45

The Value inside display function (n1 x n1) :2025

The Value of N1 is inside main function After passing (n1 x n1) : 2025

## INLINE செயற்கூறு

- inline செயற்கூறு மூல நிரலில் சாதாரணச் செயற்கூறு போன்றே தோற்றமளிக்கும். ஆனால் செயற்கூறின் கட்டளைகள் முழுமையும் அழைப்புக்கூற்றுக்குப் பதிலாக அப்படியே நிரலில் செருகப்பட்டுவிடும்.
- Inline சிறப்புச் சொல்லை செயற்கூறின் தலைப்பில் இணைத்து அந்த செயற்கூறை inline செயற்கூற்றாக மாற்ற முடியும்.

### தொடரியல்:

```
inline returntype functionname  
(datatypeparametername1,... datatype parameternameN)
```

### inline செயற்கூற்றுகளின் நன்மைகள்:

- inline செயற்கூறுகள் வேகமாக செயல்படும். ஆனால் அதிக நினைவக இடத்தை எடுத்துக் கொள்ளும்.
- அடுக்கங்களை பயன்படுத்தும் போது உள்ள சிக்கல்பாட்டினைக் குறைக்கிறது.





# எடுத்துக்காட்டு

## நிரல்:

```
#include <iostream>
using namespace std;
inline float simpleinterest(float p1,float n1, float r1)
{
float si1=(p1*n1*r1)/100;
return(si1);
}
int main ()
{
float si,p,n,r;
cout<<"\nEnter the Principle Amount Rs. :";
cin>>p;
cout<<"\nEnter the Number of Years :";
cin>>n;
cout<<"\nEnter the Rate of Interest :";
cin>>r;
si=simpleinterest(p,n,r);
cout << "\nThe Simple Interest = Rs."<<si;
return 0;
}
```

## வெளியீடு:

```
Enter the Principle Amount Rs. :60000
Enter the Number of Years :10
Enter the Rate of Interest :5
The Simple Interest = Rs.30000
```



## செயற்கூறு மதிப்பை திருப்பி அனுப்பும் பல்வேறு வடிவங்கள்

திருப்பி அனுப்பும் மதிப்பு மற்றும் அளபுருக்களை ஏற்காத  
செயற்கூறு:

- `display()` என்ற செயற்கூறின் திருப்பி அனுப்பும் தரவினம் `int` மற்றும் இந்த செயற்கூறு அளபுருவையும் ஏற்காது.
- `return` செயற்கூறு அழைப்பு செயற்கூறுக்கு மதிப்பைத் திருப்பி அனுப்பும் மற்றும் நிரலின் கட்டுப்பாட்டை மீண்டும் அழைப்புக் கூற்றுக்கு திருப்பி அனுப்பும்.



# திருப்பி அனுப்பும் மதிப்பு மற்றும் அளபுருக்களை ஏற்காத செயற்கூறு:

## எடுத்துக்காட்டு:

```
#include<iostream>
using namespace std;
int display()
{
int a, b, s;
cout<<"Enter 2 numbers: ";
cin>>a>>b;
s=a+b;
return s;
}
int main()
{
int m=display();
cout<<"\nThe Sum="<<m;
return(o);
}
```

## வெளியீடு:

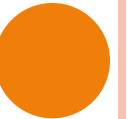
```
Enter 2 numbers: 10 30
The Sum=40
```



# செயற்கூறு மதிப்பை திருப்பி அனுப்பும் பல்வேறு வடிவங்கள்

மதிப்பை திருப்பி அனுப்பும் மற்றும் அளபுருவை ஏற்கும்  
செயற்கூறு:

- `display()`, என்ற செயற்கூறு `int` என்ற மதிப்பைத் திருப்பி அனுப்பும். மேலும் `x` மற்றும் `y` என்ற இரண்டு அளபுருக்கள் அல்லது செயலுருபுக்களில் மதிப்புகளை ஏற்கும்.
- `return` கூற்று கட்டுப்பாட்டை அழைப்பு கூற்றிக்குத் திருப்பி அனுப்பும்.



# மதிப்பை திருப்பி அனுப்பும் மற்றும் அளபருவை ஏற்கும் செயற்கூறு

## எடுத்துக்காட்டு:

```
#include<iostream>
using namespace std;
int display(int x, int y)
{
int s=x+y;
return s;
}
int main()
{
int a,b;
cout<<"\nEnter the First Number :";
cin>>a;
cout<<"\nEnter the Second Number :";
cin>>b;
int s=display(a,b);
cout<<"\nThe Sum of Passed Values: "<<s;
return(o);
}
```

## வெளியீடு:

```
Enter the First Number :45
Enter the Second Number :67
The Sum of Passed Values: 112
```



## கட்டுப்பாட்டை செயற்கூறிலிருந்து திருப்பி அனுப்புதல்

- return கூற்றை பயன்படுத்தி கட்டுப்பாட்டை செயற்கூறிலிருந்து திரும்பப் பெறலாம்.
- return கூற்று செயற்கூறின் இயக்கத்தை நிறுத்தி கட்டுப்பாட்டை அழைத்த செயற்கூறுக்கு திருப்பி அனுப்பும்.
- செயற்கூற்றின் பகுதியில் எங்கு return கூற்று இயக்கப்படுகிறதோ, உடனே அந்த இடத்தில் செயற்கூறின் செயல்பாடு நிறுத்தப்பட்டு, கட்டுப்பாடு அழைப்பு கூற்றுக்குத் திரும்பும்.



## தற்சுழற்சி செயற்கூறு (RECURSIVE FUNCTION)

- ஒரு செயற்கூறு தன்னைத் தானே அழைத்துக் கொண்டால் அதை தற்சுழற்சி செயற்கூறு என்கிறோம்.

தற்சுழற்சி முறையில் ஒரு எண்ணின் காரணியை கணக்கிடுதல்.

```
#include <iostream>
using namespace std;
int factorial(int); // Function prototype //
int main()
{
int no;
cout<<"\nEnter a number to find its factorial: ";
cin >> no;
cout << "\nFactorial of Number " << no <<" = " << factorial(no);
return 0;
}
int factorial(int m)
{
if (m > 1)
{ return m*factorial(m-1); }
else
{ return 1;}
}
```

வெளியீடு:

Enter a number to find its factorial: 5  
Factorial of Number 5 = 120



## மாறிகளின் வரையெல்லை (SCOPE RULES OF VARIABLES)

வரையெல்லை என்பது ஒரு மாறியின் அணுகியல்பைக் குறிக்கிறது. C++ மொழியில் நான்கு வகையான வரையெல்லைகள் உள்ளன . அவை,

- உள்ளமை வரையெல்லை (Local scope): ஒரு தொகுதிக்குள் அறிவிக்கும் போது அவற்றை உள்ளமை மாறிகள் என்கிறோம்.
- செயற்கூறு வரையெல்லை (Function scope): செயல்கூறின் உள்ளே அறிவித்தால் அவை செயல்கூறு மாறிகள் என்றழைக்கப்படும்.
- கோப்பு வரையெல்லை (File scope): எல்லா செயற்கூறுக்கும் வெளியே அறிவித்தால், அவற்றை பொதுமையான/ முழுதாளவிய (Global) மாறிகள் என்கிறோம். கோப்பு வரையெல்லை மாறி முழுதாளவிய மாறிகள் எனப்படும்.
- இனக்குழு வரையெல்லை (Class scope): இனக்குழுவில் உள்ளே அறிவித்தால் அவை இனக்குழு மாறிகள் அல்லது தரவு உறுப்புகள்(data members) என்று அழைக்கப்படும் .





## எடுத்துக்காட்டு:

```
#include<iostream>
using namespace std;
int sum; → கோப்பு வரையெல்லை
void add(int x, int y)
{
int z=30 → செயற்கூறு வரையெல்லை
sum=x+y+z;
}
int main()
{
int a=10;
{
int b=20; → உள்ளமை வரையெல்லை
add(a,b);
}
cout<<sum;
}
```



# வரையெல்லை தெளிவுபடுத்தும் செயற்குறி (::)

- ஒரு மாறியின் மறைந்து கிடக்கும் வரையெல்லையை வெளிக்கொணர வரையெல்லை தெளிவுபடுத்தும் செயற்குறி (::) பயன்படுகிறது.
- பொதுமையான மாறியும் உள்ளமைமாறியும் ஒரே பெயரை கொண்டிருந்தால், பொதுமையான மாறியை இயக்க வரையெல்லை தெளிவுபடுத்தும் செயற்குறி பயன்படுத்தும் ஒரு எடுத்துக்காட்டு.

## எடுத்துக்காட்டு நிரல்:

```
#include<iostream>
using namespace std;
int x=45; // Global Variable x
int main()
{
int x = 10; // Local Variable x
cout << "\nValue of global x is " << ::x;
cout << "\n\nValue of local x is " << x;
return 0;
}
```

வெளியீடு: Value of global x is 45  
Value of local x is 10



# முக்கிய வினாக்கள்

1. செயற்கூறுகள் - வரையறு
2. `strlen()` செயற்கூறைபற்றி எழுதுக
3. `void` தரவு வகையின் முக்கியத்துவங்கள் என்ன?
4. அளபுரு என்றால் என்ன ? அதன் வகைகளை பட்டியலிடுக
5. உள்ளமைவரையெல்லைபற்றி சிறுகுறிப்பு வரைக.
6. உள்ளிணைந்த செயற்கூறுகள் என்றால் என்ன ?
7. `isupper()` மற்றும் `toupper()` செயற்கூறுகளின் வேறுபாடுகள் யாவை?
8. `strcmp()` செயற்கூறு பற்றி குறிப்பு வரைக.
9. C++ மொழியில் உள்ள `pow()` செயற்கூறு பற்றி சிறுகுறிப்பு வரைக.
10. செயற்கூறு முன்வடிவம் நிரல்பெயர்ப்பிக்கு எந்த தகவலை வழங்கும்?
11. முன்னிலைப்பு செயலுருபுக்கள் என்றால் என்ன ? எடுத்துக்காட்டு தருக.
12. மதிப்பு மூலம் அழைத்தல் முறையை தகுந்த எடுத்துக்காட்டுடன் விளக்குக.
13. மாறியின் வரையெல்லைவிதிமுறைகளை எடுத்துக்காட்டுடன் விளக்குக.

# நன்றி!



கல்வி என்பது கடல்.  
அதை கற்றுக் கொடுப்பது  
தொழில் அல்ல தவம்.  
விருப்பம் பல கொண்டு  
விரைவுடன் நீ படித்தால்  
வாழ்வில் மேன்மை  
பெறலாம்..

**ஜெ. கவிதா** B.Sc, B.Ed, M.C.A, M.Phil.,  
கணினி பயிற்றுநர் நிலை - I  
அரசு மேல்நிலைப்பள்ளி,  
சர்க்கார்சாமக்குளம்,  
கோயம்புத்தூர் - 641107.

## வாழ்த்துக்கள்.

