

C++ Programming Language

Chapter 01:

Variable and Data Type

1. Variable: ជាអញ្ញត្តិ ឬអថេរប្រើសំរាប់ផ្ទុកទិន្នន័យបណ្តោះអាសន្ន។ ដើម្បីបង្កើតឈ្មោះ Variable មួយ យើងត្រូវផ្សំឡើងដូចខាងក្រោម៖

- a - z, A - Z
- ឬផ្សំជាមួយតួរលេខពី 0 – 9
- ហើយក៏អាចប្រើនិមិត្តសញ្ញា Underscore (_)។

ឧទាហរណ៍៖

```
int age;  
long student_ID;  
float total24;
```

ក្រៅពី តួអក្សរ តួរលេខ ឬនិមិត្តសញ្ញាដែលបានរៀបរាប់ខាងលើ យើងមិនអាចយក មកបង្កើតជាឈ្មោះ Variable បានទេ។

ចំណាំ៖

- ឈ្មោះ Variable មិនអាចផ្ដើមឡើងដោយតួរលេខ ឬតួរលេខសុទ្ធសាធបានទេ។

ឧទាហរណ៍៖

```
int 2age; }  
float 2.5; } Error
```

- ឈ្មោះ Variable មិនអាចដូច Keywords ឬពាក្យមានស្រាប់នៅក្នុងកម្មវិធីបានទេ។

ឧទាហរណ៍៖

```
int if; }  
float else; } Error  
double class;
```

(Keyword មានពណ៌ស មិនអាចប្រើជា Variable បានទេ ហើយបើឃើញអក្សរ ពណ៌សកុំសរសេរបន្តទៀត បើចង់រក Keywords ទាំងអស់ ចុច Shift + F1)

2. Declaration of Variables

DataType VariableName;

ឧទាហរណ៍៖

```
float salary;  
long total_time;
```

3- DataType

DataType	Size	Range
- char ជាតួអក្សរ	1 byte	- 128 to 127
- short	2 bytes	- 2^{15} to 2^{15-1}
- long	4 bytes	- 2^{31} to 2^{31-1}
- int	short/ long	
- float	4 bytes	$3.4E10^{38}$ to $+/-3.4E10^{38}$
- double	8 bytes	$1.7E10^{-308}$ to $1.7E10^{308}$
- long double	10 bytes	$1.2E10^{-4932}$ to $1.2E10^{4932}$

4- Operators សំដៅលើនិមិត្តសញ្ញាក្នុង Computer

a. **Arithmetic Operators** ជានិមិត្តសញ្ញាប្រើសំរាប់គណនា (+, -, *, /, %, Modular)

ឧទាហរណ៍៖

```
int a;  
a = 4 + 3;  
int b;  
b = a % 2;  
// a = 7  
// b = 1
```

ឧទាហរណ៍៖

```
int a = 5;  
int b = 2;  
int c =;  
c = a / b;  
// c = ?
```

ឧទាហរណ៍៖

```
float c, x;  
x = a;  
c = x / b;  
c = ?
```

ឧទាហរណ៍៖

```
float c;  
c = (a * 1.0) / b;  
//c = ?
```

ឧទាហរណ៍៖

```
c = float (a; / b;)  
  
int / int = int  
int / float = float  
float / int = float  
float / float = float
```

b. Assignment Operators ជាសញ្ញាប្រើសំរាប់បញ្ជូនតំលៃ ឬឲ្យតំលៃ

(= , += , -= , *= , /= , %= , ++ , -- ,)

ឧទាហរណ៍៖

```
int a;  
a = 5;  
int b;  
b = a;  
// a = 5  
// b = 5
```

ឧទាហរណ៍៖

```
int x = 9;  
x += 3;  
// x = 12
```

ឧទាហរណ៍៖

```
int i = 6;
```

```
i ++;
```

```
// i = 7
```

c. Compansion Operators

- -, !=, >, <, >=, <=

d. Logical Operators

!, ||, &&

ឧទាហរណ៍៖

!T = F

!F = T

T||T = T

T||F = T

F||T = T

F||F = F

T&&T = T

T&&F = F

F&&F = F

F&&F = F

The End !

Chapter 02:

First Program with C⁺⁺

1. General Structure of C⁺⁺

```
//Header files including block
#include<header.h>
//Global variable
void main(){//main function
    //Local variable
    Statement(s);
}
```

2. Statements បែងចែងជាបីក្រុមធំៗគឺ:

2.1. Expression Statement : ប្រើសំរាប់កំណត់តំលៃ ឬឲ្យតំលៃ

ឧទាហរណ៍៖

```
int a;
int b;
a = 4;
b = a;
b = a + 20; } ← Expression Statements
```

2.2. Compound Statement : ប្រើសំរាប់ផ្សំ Statements ប្រើនបញ្ចូលគ្នាក្លាយទៅជា Statement តែមួយ

ឧទាហរណ៍៖

```
{
    int x;
    x = 8;
    cout<<"x = "<<x;
}
```

2.3. Control Statement : ប្រើសំរាប់ដាក់លក្ខខណ្ឌ

ឧទាហរណ៍៖

```
int a = 5;
if(a>0)
    cout<<"A is Positive";
else
    cout<<"A is Negative or 0";
```

3. Manipulator Functions : ជា Function មួយមានតួនាទីសំរាប់សំរួលការ Input និង Output របស់ Program ។ ដើម្បីប្រើ Functions ទាំងនោះបាន យើងចាំបាច់ត្រូវការ `#include<iomanip.h>` ។ ក្នុង Header file `iomanip.h` មាន Functions ជាច្រើន ដូចជា៖

- **endl** : ជាតំលៃថេរដែលមានអត្ថន័យស្មើនឹង `\n`

ឧទាហរណ៍៖

```
cout<<"Statement1"<<endl<<"Statement2";  
cout<<"Statement1\nStatement2";  
cout<<"Statement1"<<"\n"<<"Statement2";
```

សំគាល់៖ `endl` មិនត្រូវការប្រើក្នុងសញ្ញា “ ” ដូចជា “`\n`” ទេ “`endl`” ។

- **setbase()** : ជា Function មានតួនាទីប្តូរប្រព័ន្ធ Output ប្រព័ន្ធគោល10 ប្រព័ន្ធគោល16 និងប្រព័ន្ធគោល8។

ឧទាហរណ៍៖

```
int x = 12;  
cout<<"Decimal ="<<setbase(10)<<x<<endl;  
cout<<"Hexadecimal ="<<setbase(16)<<x<<endl;  
cout<<"Octal ="<<setbase(8)<<x<<endl;
```

- `dec`, `hex`, `oct` ជាតំលៃថេរដែលមានអត្ថន័យស្មើនឹង `setbase(10)` `setbase(16)` & `setbase(8)`។

ឧទាហរណ៍៖

```
cout<<"Decimal ="<<dec<<x<<endl;  
cout<<"Hexadecimal ="<<hex<<x<<endl;  
cout<<"Octal ="<<oct<<x<<endl;
```

- **setw()** : ជា Function មួយមានតួនាទីកំណត់ប្រវែងតិចបំផុតសំរាប់ Output។

ឧទាហរណ៍៖

```
cout<<"I'm"<<setw(10)<<"Dara"; //(setw បញ្ជាក់ចំនួនSpacesអោយទៅពាក់កណ្តាលថា Dara)  
//output: I'm      Dara.  
                _____  
                |-----|  
                6 Spaces
```

ឧទាហរណ៍៖

```
cout<<"I'm"<<setw(2)<<"Theara";  
//output: I'mTheara.
```

- **setfill()** : ជា Function មួយដែលមានតួនាទីបំពេញចន្លោះមិនបានប្រើដោយ `setw`។

ឧទាហរណ៍៖

```
cout<<setfill('*');
cout<<"I'm"<<setw(10)<<"Dara";
//output: I'm ***** Dara.
```

ឧទាហរណ៍៖

```
cout<<setfill('*');
cout<<"I'm"<<setw(2)<<"Theara";
//output: I'mTheara.
```

ឧទាហរណ៍៖

```
float m = 2.5;
cout<<setfill('*');
cout<<"Total"<<setw(10)<<m<<"$";
Total ***** 2.5 $
```

- **setprecision()** : ជា Function មួយ មានតួនាទីកំណត់ចំនួនខ្ទង់ក្រោយរង្វង់ស។

ឧទាហរណ៍៖

```
float r, a = 5, b = 3;
r = a / b;
cout<<setprecision(1);
cout<<"R ="<<r<<endl; //R = 1.7
cout<<setprecision(2);
cout<<"R ="<<r<<endl; //R = 1.67
```

The End !

Chapter 03:

Decision Making

1. If Statement

1.1. if Statement : ជា Statement មួយប្រើសំរាប់ដាក់លក្ខខណ្ឌឱ្យ Statement ផ្សេងទៀត បើ លក្ខខណ្ឌពិត Statement ទាំងអស់របស់ if ត្រូវបាន execute ។

ទម្រង់៖

```
if(condition){  
    Statement(s);    Statement អាចមានច្រើន  
}
```

ឧទាហរណ៍៖

```
int a = 5, b = 3;  
if(a>b){  
    cout<< a;    ចូលមួយ Tab ព្រោះនៅក្រោមលក្ខខណ្ឌ if  
}
```

ឧទាហរណ៍៖

```
int x = 8;  
if(x == 0){  
    cout<< "x is 0";  
}
```

1.2. if else Statement : ជា Statement ប្រើសំរាប់ដាក់លក្ខខណ្ឌឱ្យ Statement ផ្សេងទៀត បើ លក្ខខណ្ឌពិត Statement ទាំងអស់របស់ if ត្រូវបាន execute តែបើមិនពិត Statement ទាំងអស់របស់ else ត្រូវបាន execute វិញ។

ទម្រង់៖

```
if(condition){  
    Statement(s);    Statement អាចមានច្រើន  
}  
else{
```

```
        Statement(s);  
    }
```

ឧទាហរណ៍៖

```
int a = 5;  
int b = 3;  
if(a>b)  
    cout<<"Maximum is"<<"a";  
else  
    cout<<"Maximum is"<<"b";
```

1.3 if elseifelse Statement: ជា Statement ប្រើសំរាប់ដាក់លក្ខខណ្ឌមានចំនួនច្រើន។

ទម្រង់៖

```
if(condition1){  
    Statement(s);    Statement អាចមានច្រើន  
}  
else if(condition2){  
    Statement(s);  
}  
else if(condition3){  
    Statement(s);  
}  
.....  
else{  
    Statement(s);  
}
```

ឧទាហរណ៍៖

```
#include<iostream.h>  
void main(){  
    char ch = 'R';  
    if (ch == 'R')  
        cout<<"Red";  
    else if (ch == 'G')  
        cout<<"Green";  
    else if (ch == 'B')  
        cout<<"Blue";  
    else  
        cout<<"Unknown Color";  
}
```

Homework

ចូរសរសេរ Program មួយ ដើម្បីគណនាកម្រៃប្រាក់ខែរបស់កម្មករក្នុងរោងចក្រមួយ។

- កម្មករត្រូវធ្វើការឲ្យបាន 100 ម៉ោង/ខែ នឹងទទួលប្រាក់ខែបាន 50 ដុល្លារ
- ក្នុងករណីដែលកម្មករបានធ្វើការច្រើនម៉ោងលើសការកំណត់ ចាប់ពី 1 ម៉ោងដល់ 20 ម៉ោង នោះទទួលប្រាក់បន្ថែមបាន 1.25 ដុល្លារ/ម៉ោង
- ក្នុងករណីដែលកម្មករបានធ្វើការច្រើនម៉ោងលើសការកំណត់ លើសពី 20 ម៉ោងឡើងទៅ នោះទទួលប្រាក់បន្ថែមបាន 1.5 ដុល្លារ/ម៉ោង
- ក្នុងករណីដែលកម្មករបានធ្វើការតិចជាងម៉ោងដែលបានកំណត់ នោះនឹងកាត់ប្រាក់ខែ 0.5 ដុល្លារ/ម៉ោង។

ចូរបង្ហាញរបាយការណ៍កម្មករ។

ដំណោះស្រាយ៖

ស្រង់សម្មតិកម្មដោយដៃ (Do Manually)៖

- 100 h → \$ 50
- [1 – 20] Bonus = \$ 1.25/h
- > 20 h Bonus = \$ 1.5/h
- < 100 h Lose = \$ 0.5/h

1. Enter Total Time

- Enter Total Time
- Basic Salary = \$50
-
- Total Salary = Basic Salary + Bonus Salary

2. Enter Total Time

- Enter Total Time = 117
- Extra time = 17
- Bonus rate = \$ 1.25/h
- Bonus Salary = Extra time * Bonus rate
-
- Total Salary = Total Salary + Bonus Salary

3. Enter Total Time

- Enter Total Time = 180
- Extra time = 80 h
- Extra Time1 = 20 h
- Bonus Rate1 = \$ 1.25/h
- Total Bonus Salary1 = Extra1 * 1.25
-

4. **Enter Total Time**

– Extra Time2 = 60 h
– Bonus Rate2 = \$ 1.5/h
– Total Bonus Salary2 = Extra Time2 * 1.5
–

5. **Total Salary**

– Total Salary = Basic Salary + Bonus Salary1 + Bonus Salary2
–

6. **Enter Total Time**

– Total Time = 80
– Lost Time = 20
– Lost Rate = \$ 0.5/h
– Lost Salary = Lose Rate * Lose Time
–
– Total Salary = Basic Salary * Lost Salary

7. **Enter Total Time**

– Total Time = - 70
– Output = Input Valid

ដំណោះស្រាយ

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    float total_t,extra_t,bonus_r,bonus_s,bonus_r2,bonus_s2,extra_t2,lose_t,lose_r,lose_s,total_s;
    cout<<"Enter Total Time=";<<cin>>total_t;
    if(total_t>0){
        if(total_t>=100){
            if(total_t ==100)
                cout<<"Total Salary=$50";
            else if(total_t<=120){
                extra_t=total_t-100;
                bonus_r=1.25;
```

```
bonus_s=bonus_r*extra_t;
total_s=50+bonus_s;
cout<<"Extra Time="<<extra_t<<"h";
cout<<"\nBonus Rate=$"<<bonus_r;
cout<<"\nBonus Salary=$"<<bonus_s;
cout<<"\n_____";
cout<<"\nTotal Salary =$"<<total_s;}

else if(total_t>120){
    extra_t=total_t-100;
    bonus_r=1.25;
    bonus_s=20*bonus_r;
    extra_t2=extra_t-20;
    bonus_r2=1.5;
    bonus_s2=bonus_r2*extra_t2;
    total_s=50+bonus_s+bonus_s2;
    cout<<"Extra Time="<<extra_t<<"h";
    cout<<"\nExtra 1=20h";
    cout<<"\nBonus Rate=$"<<bonus_r;
    cout<<"\nBonus Salary=$"<<bonus_s;
    cout<<"\n_____";
    cout<<"\nExtra t2="<<extra_t2<<"h";
    cout<<"\nBonus Rate2=$"<<bonus_r2;
    cout<<"\nBonus Salary=$"<<bonus_s2;
    cout<<"\n_____";
    cout<<"\nTotal Salary=$"<<total_s;
    }
}

else {
    lose_t=100-total_t;
    lose_r=0.5;
    lose_s=0.5*lose_t;
    total_s=50-lose_s;
    cout<<"Lose Time="<<lose_t;
    cout<<"\nLose Rate=$"<<lose_r;
```

```
        cout<<"\nLose Salary=$"<<lose_s;
        cout<<"\n_____";
        cout<<"\nTotal Salary=$"<<total_s;
    }
}

else

    cout<<"Invalid Input";

    getch();
}
```

2- Switch Statement

Switch statement ជា statement មួយប្រើសំរាប់ជ្រើសរើស Statement ផ្សេងៗទៀត ដើម្បី Execute ។

Form: ទំរង់

```
switch (expression){
    case value1:
        statement(s);
    case value2:
        statement(s);
    case value3:
        statement(s);
    default:
        statement(s);
}
```

ឧទាហរណ៍៖

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    char ch = 'R';
    switch (ch){
        case 'R':
            cout<<"Red";
        case 'G':
            cout<<"Green";
        case 'B':
            cout<<"Blue";
        default:
            cout<<"Unknown Color";
    }
    getch();
}
```

```
}
```

ចំណាំ៖

- ក្នុងករណី Case ទី១ពិត Switch execute ពី Case ទី១ពិត ចុះមកក្រោមលទ្ធផលរបស់វាគឺ៖ RedGreenBlueUnknownColor ។
- តែបើករណី Case ទី១មិនពិត Switch execute ចាប់ពី Case ទីពីរចុះមកក្រោមលទ្ធផលរបស់វាគឺ៖ GreenBlueUnknownColor ។
- វានៅតែ Execute ដូច្នេះរហូតដល់ Default ។
- Default ទៀងតែ Execute ទោះបីវាមិនពិតក៏ដោយ ព្រោះវាចុងក្រោយគេហើយ

ចំណាំ៖

- ទាំងក្នុង if-else statement និង switch statement បើយើងមិនចង់ឲ្យប្រកាន់អក្សរតូចនិងអក្សរធំ យើងត្រូវប្រកាសអក្សរទាំងពីរប្រភេទនេះ។

ឧទាហរណ៍៖ នៅក្នុង if-else statement

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    char ch = 'R';
    if (ch == 'R' || ch == 'r')
        cout<<"Red";
    else if (ch == 'G' || ch == 'g')
        cout<<"Green";
    else if (ch == 'B' || ch == 'b')
        cout<<"Blue";
    else
        cout<<"Unknown Color";
    getch();
}
```

ឧទាហរណ៍៖ នៅក្នុង switch statement

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    char ch = 'R';
    switch(ch){
        case 'R':
        case 'r':
```

```
        cout<<"Red";break;
    case 'G':
    case 'g':
        cout<<"Green";break;
    case 'B':
    case 'b':
        cout<<"Blue";break;
    default:
        cout<<"Unknown Color";
    }
    getch();
}
```

ឬម្យ៉ាងទៀត

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    char ch = 'R';
    switch (ch){
        case 'R':
            cout<<"Red";
        case 'r':
            cout<<"Red";
            break;
        case 'G':
            cout<<"Green";
        case 'g':
            cout<<"Green";
            break;
        case 'B':
            cout<<"Blue";
        case 'b':
            cout<<"Blue";
            break;
        default:
            cout<<"Unknown Color";
    }
    getch();
}
```

ចំណាំ៖

- 2 ត្រូវបានគេហៅថាជា Number អាចយកមកគណនាបាន
- '2' ត្រូវបានគេហៅថាជា Character អាចយកមកគណនាបានដែរ តែគេមិនដែលយកមកគណនាទេ ព្រោះ Character ក្នុង Single quote ជាកំណត់តែ១ មិនអាចជាកំណត់ច្រើនទេ '24'

- “2” ត្រូវបានគេហៅថាជា String អាចផ្ទុកស្ត្រីក៏បាន តែមិនអាចយកមកគណនាបានទេ ទោះបីសរសេរជាលេខក៏ដោយ

ចំណាំ៖

- Switch ជា Statement មួយខុសពី if-else statement ព្រោះវានឹង execute ចាប់ពីករណីពិតចុះមកក្រោម ហើយ Default តែងតែ Execute ទោះបីក្នុងករណីណាក៏ដោយ។
- Case មិនត្រូវការ Curly bracket { } ទេ។

ចំណាំ៖

- តែបើយើងត្រូវការឲ្យ Switch execute តែករណីពិត យើងត្រូវប្រើ break statement(s) គ្រប់ករណីទាំងអស់របស់ Switch ព្រោះ break statement មានតួនាទីចាកចេញពី Switch ។

ឧទាហរណ៍៖

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    char ch = 'R';
    switch (ch){
        case 'R':
            cout<<"Red";
            break;
        case 'G':
            cout<<"Green";
            break;
        case 'B':
            cout<<"Blue";
            break;
        default:
            cout<<"Unknown Color";
    }
    Getch();
}
```

Chapter 04:

Loop Statement

ប្រើសំរាប់ execute statement មួយ ឬច្រើន សារទៅសារមករហូតទល់ តែជួបលក្ខខណ្ឌពិត។

1- while loop :

ជា Loop ដែលចាំបាច់ត្រូវផ្ទៀងផ្ទាត់ជាមុនសិន ទើប execute statement របស់វា។

Form :

```
while(condition){  
    statement(s);  
}
```

ឧទាហរណ៍ទី១៖

```
int i = 1;  
while(i<=5){  
    cout<<i<<",";  
    i ++;  
}  
//output: 1,2,3,4,5,
```

សរសេរជាទម្រង់ពេញលេញ

```
#include<iostream.h>  
#include<conio.h>  
void main(){  
    clrscr();  
    int i = 1;  
    while(i<=5){  
        cout<<i<<",";  
        i ++;  
    }  
    getch();  
}
```

ឧទាហរណ៍ទី២៖

```
int i = 5;  
while(i>=1){  
    cout<<i<<",";  
    i - -;
```

```
    }  
    //output: 5,4,3,2,1,  
សរសេរជាទម្រង់ពេញលេញ  
#include<iostream.h>  
#include<conio.h>  
void main(){  
    clrscr();  
    int i = 5;  
    while(i>=1){  
        cout<<i<<" ";  
        i - -;  
    }  
}
```

សំគាល់ ៖

- ករណីគណនាតំលៃលេខកើន គឺប្រើ `i++` ហើយតំលៃចាប់ផ្តើមត្រូវតែតូចជាងតំលៃចុងក្រោយ។
 - `i++` កើនម្តងមួយលេខ
 - `i+=2` កើនម្តងពីរលេខ
 - `i+=3` កើនម្តងបីលេខ
 - `i+=4` កើនម្តងបួនលេខ
- ករណីគណនាតំលៃលេខថយ គឺប្រើ `i--` ហើយតំលៃចាប់ផ្តើមត្រូវតែធំជាងតំលៃចុងក្រោយ។
 - `i--` ថយម្តងមួយលេខ
 - `i-=2` ថយម្តងពីរលេខ
 - `i-=3` ថយម្តងបីលេខ
 - `i-=4` ថយម្តងបួនលេខ

Sample instances ឧទាហរណ៍គំរូ

```
#include<iostream.h>  
#include<conio.h>  
void main(){  
    clrscr();  
    int n;  
    long sum=0;  
    cout<<"Enter n=";  
    cin>>n;  
    int i = 1;  
    while(i<=n){
```

```
        sum = sum + 1;
        i ++;
    }
    cout<<"Sum ="<<sum;
    getch();
}
```

លំហាត់ទី១៖

ចូរគណនាតំលៃលេខ 15 ដោយរៀបពីលេខ $5 + 4 + 3 + 2 + 1 = 15$ ។ Enter 5 ដើម្បី

រកលទ្ធផល 15 ។

ដំណោះស្រាយ

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    long sum=0;
    cout<<"Enter n=";
    cin>>n;
    int i = n;
    while(i>=1){
        cout<<i<<"+";
        sum = sum + i;
        i --;
    }
    cout<<"\b ="<<sum;
    getch();
}
```

លំហាត់ទី២៖

ចូរគណនាតំលៃលេខ 15 ដោយរៀបពីលេខ $1 + 2 + 3 + 4 + 5 = 15$ ។ Enter 5 ដើម្បី

រកលទ្ធផល 15 ។

ដំណោះស្រាយ

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    long sum=0;
    cout<<"Enter n=";
    cin>>n;
    int i = n;
    while(i<=15){
        cout<<i<<"+";
        sum = sum + i;
        i ++;
    }
```

```
    }  
    cout<<"\b ="<<sum;  
    getch();  
}
```

2- Do While Loop:

ជា Loop ចាំបាច់ execute statements ជាមុន ទើបផ្ទៀងលក្ខខណ្ឌ ជាក្រោយ។

Form:

```
do{  
    statements;  
}while(condition);
```

e.g.:

```
int i = 1;  
do{  
    cout<<i<<" ";  
    i ++;  
}while(i<=5);  
//output: 1,2,3,4,5,
```

e.g.:

```
int i = 5;  
do{  
    cout<<i<<" ";  
    i - -;  
}while(i<=1);  
//output: 5,4,3,2,1,
```

Comparing While Loop and Do While Loop

e.g.:

```
int i = 5;  
while(i>=6){  
    cout<<i<<" ";  
    i ++;  
}
```

(Loop នេះដើរមិនរួចទេ ព្រោះលក្ខខណ្ឌមិនពិត)

e.g.:

```
int i = 5;  
do{  
    cout<<i<<" ";  
    i ++;  
}while(i>=6);
```

(Loop នេះដើររួច យ៉ាងហោចណាស់បានមួយជំងឺដែរ ព្រោះ execute statement រួចទើបផ្ទៀងផ្ទាត់លក្ខខណ្ឌជាក្រោយ)

3- for Loop:

ជា Loop មួយដែលមានលក្ខណៈពិសេសជាង Loop ដទៃ ព្រោះនៅក្នុង Expression របស់វាមាន 3 statements ដោយ statements ទាំង 3 នេះមានតួនាទីដូចជា៖

- Initialization: ជា Statement មួយប្រើសំរាប់កំណត់តម្លៃចាប់ផ្តើម Loop ហើយ statement នេះ execute តែម្តងគត់នៃដំណើរការនៃ Loop។
- Condition: ជា Statement មួយប្រើសំរាប់ដាក់លក្ខខណ្ឌឱ្យ Loop
- Step: ជា Statement ប្រើសំរាប់កំណត់ជំហានរបស់ Loop។

Form:

```
for(Initialization; condition; step){  
    statements;  
}
```

Comparing for Loop and While Loop

e.g.:

```
long sum=0;  
for(int i = 1; i<=5; i++){  
    cout<<i<<" ";  
    sum += i;  
}  
cout<<"\b ="<<sum;
```

e.g.:

```
int i = 1;  
long sum=0;  
while(i<=5){  
    cout<<i<<" ";  
    sum += i;  
    i++;  
}  
cout<<"\b ="<<sum;
```

ឧទាហរណ៍៖

```
#include<iostream.h>  
#include<conio.h>  
void main(){  
    clrscr();  
    long sum=1;  
    for(int i=1; i<=5, n!=i; i++){  
        cout<<i<<"*";  
        sum *=i;  
    }  
    cout<<"\b ="<<sum;  
    getch();  
}
```

Enter 5 ទៅ លទ្ធផលចេញដូចនេះ ៖ $5! = 1*2*3*4*5 = 120$

4- Nested Loop ៖ គឺសំដៅទៅលើ Loop នៅក្នុង Loop ។

ឧទាហរណ៍៖

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    cout<<"Enter n=";
    cin>>n;
    for(int i=1; i<=n; i++){
        for(int j=1; j<=i; j++){ ← Loop ខាងក្នុងដើរមុន
            cout<<"*";
            cout<<"\n";
        }
        getch();
    }
}
```

លទ្ធផលរបស់វាគឺ៖

```
Enter n=5
*
* *
* * *
* * * *
* * * * *
```

បើលទ្ធផលបែបនេះវិញ

```
Enter n=5
* * * * *
* * * *
* * *
* *
*
```

តើត្រូវសរសេរ Code ដូចម្តេច?

វាមានការសរសេរ Code ៣ របៀប៖

រូបថតទី ១៖

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    cout<<"Enter n=";
    cin>>n;
    for(int i=n; i>=1; i--){
        for(int j=1; j<=i; j++){ ← Loop ខាងក្នុងដើរមុន
            cout<<"*";
            cout<<"\n";
        }
        getch();
    }
}
```

រូបថតទី ២

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    cout<<"Enter n=";
    cin>>n;
    for(int i=1; i<=n; i++){
        for(int j=n; j>=i; j--){ ← Loop ខាងក្នុងដើរមុន
            cout<<"*";
            cout<<"\n";
        }
    }
}
```



```
    getch();  
}
```

រូបថតទី ៣

```
#include<iostream.h>  
#include<conio.h>  
void main(){  
    clrscr();  
    int n;  
    cout<<"Enter n=";  
    cin>>n;  
    for(int i=n; i>=1; i--){  
        for(int j=i; j>=1; j--){ ← Loop ខាងក្នុងដើរមុន  
            cout<<"*";  
            cout<<"\n";  
        }  
        getch();  
    }  
}
```

5- Jumping statement ៖ ជា statement មួយប្រើសំរាប់ execute រំលងទៅកាន់ statement ផ្សេង មួយទៀតដោយគ្មានលក្ខខណ្ឌ។

- i. **Break statement** ៖ ជា statement មួយប្រើសំរាប់ចាកចេញពី switch ឬពី Loop។

ឧទាហរណ៍៖

```
for(int i=1; i<=1000; i++){  
    cout<<i<<" ";  
    if(i == 5){  
        cout<<"Loop Aborted";  
        break;  
    }  
}
```

// output: 1, 2, 3, 4, 5, Loop Aborted

- ii. **Continue statement** ៖ ជា statement មួយប្រើសំរាប់ចាកចេញពីរំលង១ជុំនៃ Loop។

ឧទាហរណ៍៖

```
for(int i=1; i<=1000; i++){
    if(i == 5; || i == 9; || i == 10)
        continue;
    cout<<i<<" ";
}
// output: 1, 2, 3, 4, 6, 7, 8, 11, 12, ..., 1000,
// skip: 5, 9, 10,
(គេប្រើនសរសេរនៅក្នុងកម្មវិធីឆ្នោត Loto)
```

- iii. **Goto label statement** ៖ ជា statement មួយប្រើសំរាប់ execute ទៅកាន់ Label ដែល យើងបានកំណត់។

```
#include<iostream.h>
#include<conio.h>
void main(){
    BACK:
    clrscr();
    int a, b, sum;
    cout<<"Enter a="; cin>>a;
    cout<<"Enter b="; cin>>b;
    sum = a +b;
    cout<<"Sum ="<<sum;
    char ch;
    cout<<"Do you want to continue[Y/N]?"; cin>>ch;
    if(ch == 'Y' || ch == 'y')
        goto BACK;
}
```

The End!

Chapter 05:

Function

1. Introduction

ដើម្បីបង្កើតកម្មវិធីមួយ គេតែងតែបំបែកកម្មវិធីនោះទៅជាកម្មវិធីតូចៗ ក្នុងគោលបំណង បែងចែកការងារ ការពុះបំបែក Program គេឲ្យឈ្មោះថា Sub-program។ នៅក្នុងភាសា C++ ក៏ មាន Sub-program ដែរ ដែលគេឲ្យឈ្មោះថា Function។

ការសរសេរ Program ដោយប្រើ Function មានប្រយោជន៍ ដូចជា៖

- ងាយស្រួលក្នុងការសរសេរ Code
- ងាយស្រួលអាន និងរុករកកំហុស
- ងាយស្រួលកែតម្រូវ និងផ្លាស់ប្តូរ
- យើងសរសេរ Code តែម្តង អាចកោះហៅយកមកប្រើប្រាស់បានច្រើនដង និងច្រើន ទិសដៅ
- យើងអាចហៅ Function យកមកប្រើនៅក្នុង Function របស់ខ្លួនឯង

2. Function

2.1. Function has **return_type**:

Form:

```
Return_type FunctionName(Arguments){  
    Statement(s);  
    Return value;  
}
```

ឧទាហរណ៍៖

```
long Sum(int a, int b){  
    long S;  
    g = a + b;  
    return S;  
}
```

```
#include<iostream.h>
#include<conio.h>
long Sum(int a, int b){
    long S;
    return S;s
}
void main(){
    clrscr();
    int a = 5, b = 3;
    long S;
    S = Sum(a, b);
    cout<<"Sum ="<<S;
    getch();
}
```

តើហៅថា Sub-program ត្រូវនៅលើ

Main Program

Main Program

ដំណើរការគឺត្រូវចេញពី Main Program មុន រួចទើបមកកាន់ Sub-program ។

ចំណាំ៖

- return_type ជា data របស់ Function
- FunctionName ជាឈ្មោះរបស់ Function ដែលយើងបង្កើតដោយខ្លួនឯង
- Arguments ជា Variable ដែលត្រូវបានបង្កើតនៅក្នុង Expression របស់ Function មានតួនាទីរង់ចាំទទួលតម្លៃពី Parameter (Parameter ជាតម្លៃដែលត្រូវបានបញ្ជូនទៅឲ្យ Arguments នៅពេល Function នោះត្រូវបានកោះហៅ)។
- return_value ជា Statement មានតួនាទីបញ្ចប់ Function និងបញ្ជូនតម្លៃទៅឲ្យ Function តម្លៃដែលត្រូវបានបញ្ជូនចាំបាច់ត្រូវមានប្រភេទទិន្នន័យ ដូចទៅនឹង return_type របស់ function ។

```
#include<iostream.h>
#include<conio.h>
float Max(float a, float b){
```

```
        if(a>b)
            return a;
        else
            return b;
    }
void main(){
    clrscr();
    float a, b;
    cout<<"Enter a="; cin>>a;
    cout<<"Enter b="; cin>>b;
    float max;
    max = Max(a,b);
    cout<<"Maximum is"<<max;
    getch();
}
```

```
#include<iostream.h>
#include<conio.h>
float Max(float a, float b){
    if(a>b)
        return a;
    else
        return b;
}
void main(){
    clrscr();
    float a, b, c;
    cout<<"Enter a="; cin>>a;
    cout<<"Enter b="; cin>>b;
    cout<<"Enter c="; cin>>c;
    float max;
    max = Max(a,b);
    max = Max(max, c);
    cout<<"Maximum is"<<max;
    getch();
}
```

Enter width = 10

Enter height = 5

$\text{Parameter} = (10 + 5) * 2$

$\text{Area} = 10 * 5$

```
#include<iostream.h>
#include<conio.h>
float Parameter (float width, float height){
    float Para = (w * h) * 2;
    return Para;
}
float Area (float w, float h){
    float A = w * h;
    return a;
}
void main(){
    clrscr();
    float w, h;
    cout<<"Enter width="; cin>>w;
    cout<<"Enter height="; cin>>h;
    cout<<"Parameter ="; <<Parameter (w, h);
    cout<<"\nArea ="<<Area(w, h);
    getch();
}
```

Enter R = 10

$\text{Parameter} = 2 * 3.14 * R$

$\text{Area} = 3.14 * R * R$

```
#include<iostream.h>
#include<conio.h>
float Parameter (float R){
    float Para = 2 * 3.14 * R;
```

```
        return R;
    }
float Area (float R){
    float A = 3.14 * R *R;
    return R;
}
void main(){
    clrscr();
    float R;
    cout<<"Enter R="; cin>>R;
    cout<<"Parameter ="<<Parameter(R);
    cout<<"\nArea ="<<Area(R);
    getch();
}
```

2.2. Function has not return Values

Form:

```
Void FunctionName(Argument){
    Statement(s);
}
```

```
#include<iostream.h>
#include<conio.h>
void star(int n){
    for(int i=1; i<=n; i++){
        for(int j=1; j<=n; j++)
            cout<<"*";
        cout<<"\n";
    }
}
void main(){
    clrscr();
    int n;
    cout<<"Enter n="; cin>>n;
    Star(s);
    getch();
}
```

ចំណាំ៖

Functions មាន 2 ប្រភេទគឺ៖

- Function មាន Return Value: គឺជា Function ដែលមាន Data type ជា char, short, int, long, float, double, long double និងមាន Return Statement Function ប្រភេទនេះ គេតែងតែប្រើសំរាប់គណនា ឬរកកត្តាផ្សេងៗ។
- Function គ្មាន Return Value: គឺជា Function ដែលមាន Return type ជា Void និងគ្មាន Return Statement ទេ។ Function ប្រភេទនេះ គេតែងតែប្រើសំរាប់ Output ឬបង្ហាញទិន្នន័យផ្សេងៗ។

```
#include<iostream.h>
#include<conio.h>
Void Display(int n){
    For(int i=1; i<=n; i++){
        cout<<i<<" ";
        cout<<"\b";
    }
}
float sum(int n){
    float s = 0;
    for(int i=1; i<=n; i++)
        s = s + i;
    return s;
}

void main(){
    clrscr();
    int n;
    cout<<"Enter n="; cin>>n;
    Dislay(n);
    cout<<"="<<sum(n);
    getch();
}
```

2.3. Value and Reference Parameter

ការបញ្ជូនតម្លៃពី Parameter ទៅ Arguments យើងអាចបញ្ចូលបានពីរយ៉ាងគឺ Value Parameter ឬ Reference Parameter។

- Value Parameter ជា Parameter ដែលបញ្ជូនតម្លៃទៅឲ្យ Arguments ប្រសិនបើ Arguments ប្រែប្រួលតម្លៃ នោះតម្លៃរបស់ Parameter មិនប្រែប្រួលតាមតម្លៃរបស់ Arguments ទេ។

```
#include<iostream.h>
#include<conio.h>
void change(int a, int b){
    a = a + 5;
    b = b * 2;
}

void main(){
```



```
clrscr();
int a = 4, b = 3;
change(a, b);
cout<<"A ="<<a;
cout<<"B ="<<b;
getch();
}
```

- Reference Parameter ជា Parameter ដែលបញ្ជូន address ទៅឲ្យ Arguments ប្រសិនបើ Arguments ប្រែប្រួលតម្លៃ នោះតម្លៃរបស់ Parameters នឹងប្រែប្រួលតាមតម្លៃរបស់ Arguments ដែរ។

```
#include<iostream.h>
#include<conio.h>
void change(int&a, int&b){
    a = a + 5;
    b = b * 2;
}
void main(){
    clrscr();
    int a = 4, b = 3;
    change(a, b);
    cout<<"A ="<<a;
    cout<<"B ="<<b;
    getch();
}
```

ចំណាំ៖

- Value Parameter ជា Parameter ដែលគេប្រើសំរាប់បញ្ជូនតម្លៃទៅឲ្យ Arguments ដើម្បីគណនារកអ្វីមួយដោយមិនចាំបាច់យកតម្លៃរបស់ Arguments ត្រលប់មកវិញទេ។
- Reference Parameter ជា Parameter ដែលគេប្រើសំរាប់បញ្ជូនតម្លៃ address ទៅឲ្យ Arguments ដើម្បីគណនា ឬរកតម្លៃថ្មីរបស់ Arguments ហើយគេពិតជាត្រូវការតម្លៃរបស់ Arguments ត្រលប់មកវិញ។

```
#include<iostream.h>
#include<conio.h>
void Input(int&a, int&b, int&c){
    cout<<"Enter a="; cin>>a;
    cout<<"Enter b="; cin>>b;
    cout<<"Enter c="; cin>>c;
}
int Max(int a, int b){
    if(a>b)
        return a;
    else
        return b;
}
```

```
}
void main() {
    clrscr();
    int a, b, c, max;
    Input(a, b, c);
    max = Max(a, b);
    max = Max(max, c);
    cout<<"Maximum is"<<max;
    getch();
}
#include<iostream.h>
#include<conio.h>
void Input(float&w, float&h){
    cout<<"Enter width="; cin>>w;
    cout<<"Enter height="; cin>>h;
}
void ParameterArea(float w, float h, float&P, float&A){
    P = (w + h) * 2;
    A = w * h;
}
void main() {
    clrscr();
    float w, h, P, A;
    Input(w, h);
    ParameterArea(w, h, P, A);
    cout<<"Parameter ="<<P;
    cout<<"\nArea ="<<A;
    getch();
}
```

```
C:\TC\INCLUDE\Input.h
void main(int&a, int&b, int&c){
    cout<<"Enter a="; cin>>a;
    cout<<"Enter b="; cin>>b;
    cout<<"Enter c="; cin>>c;
}
```

```
C:\TC\INCLUDE\Max.h
int Max(int a, int b){
    if(a>b)
        return a;
    else
        return b;
}
```

```
C:\TC\BIN\Test.cpp
#include<iostream.h>
#include<conio.h>
#include<Input.h>
#include<Max.h>
```

```
C:\TC\INCLUDE\Sub.h
void Input(float&w, float&h){
    cout<<"Enter width="; cin>>w;
    cout<<"Enter height="; cin>>h;
}
void ParaArea(float w, float h, float&P, float&A){
    P = (w + h) * 2;
    A = w * h;
}
#include<iostream.h>
#include<conio.h>
#include<Sub.h>
void main(){
    clrscr();
    float w, h, P, A;
    Input(w, h);
    ParaArea(w, h, P, A);
    cout<<"Parameter = "<<P;
    cout<<"\nArea = "<<A;
    getch();
}
```

C:\TC\INCLUDE\Sub.h

```

cout<<"\n";
cout<<"\n";
cout<<"C++\n";
cout<<"\n";
cout<<"BE\n";
cout<<"\n";
cout<<"INT\n";
cout<<"\n";
cout<<"Mgt\n";
cout<<"\n";
cout<<"App\n";
cout<<"\n";
cout<<"Avg\n";
cout<<"\n";
}
void Input(float&S, int x, int y){
    gotoxy(x, y);
    cin>>S;
    while(S<0 || S>100){
        gotoxy(x, y);
        cout<<"Invalid Input";
        getch();
        gotoxy(x, y);
        cout<<"_____";
        gotoxy(x, y);
        cin>>S;
    }
}

```

C:\TC\INCLUDE\Edit.h

```

void ShowEdit(){
    gotoxy(20,2);
    cout<<"1.C++";
    gotoxy(20,3);
    cout<<"2.Be";
    gotoxy(20,4);
    cout<<"3.Int";
    gotoxy(20,5);
    cout<<"4.Mgt";
    gotoxy(20,6);
    cout<<"5.Ap";
    gotoxy(20,7);
    cout<<"6.Result";
}

```

```

void Edit(float&cpp, float&Be, float&Int, float&Mgt, float&Ap, float&AVG){
    LbEdit:
    ShowEdit();
    gotoxy(20,8);
    cout<<"Press key...";
}

```

```
char ch;
ch = getch();
if(ch == '1'){
    gotoxy(10,6);
    cout<<"_____";
    Input(cpp,10,6);
    gotoLbEdit;
}else if(ch == '2'){
    gotoxy(10,8);
    cout<<"_____";
    Input(Be,10,8);
    goto LbEdit;
}else if(ch == '3'){
    gotoxy(10,10);
    cout<<"_____";
    Input(Int,10,10);
    goto LbEdit;
}else if(ch == '4'){
    gotoxy(10,12);
    cout<<"_____";
    Input(Mgt,10,12);
    goto LbEdit;
}else if(ch == '5'){
    gotoxy(10,14);
    cout<<"_____";
    Input(Ap,10,14);
    goto LbEdit;
}else if(ch == '6'){
    gotoxy(10,16);
    cout<<"_____";
    AVG = (cpp + Be + Int + Mgt + Ap)/ 5;
    cout<<AVG;
}else
    goto LbEdit;
}
```

```
#include<iostream.h>
#include<conio.h>
#include<Sub.h>
#include<Edit.h>
void main(){
```

```
clrscr();
Table();
float cpp, be, Int, Mgt, Ap, AVG;
Input(cpp, 10, 6);
Input(be, 10, 8);
Input(Int, 10, 10);
Input(Mgt, 10, 12);
Input(Ap, 10, 14);
AVG=(cpp+be+Int+Mgt+Ap)/ 5;
Gotoxy(10, 16);
cout<<Avg;
getch();
}
```

2.4. Defaults Values In Arguments

ជាការឲ្យតំលៃទៅ Arguments ធ្វើឲ្យ Arguments នោះ អាចទទួលតំលៃរបស់ Parameter ក៏បាន ឬមិនទទួលក៏បាន។

ឧទាហរណ៍៖

```
float Max(float x, float b=0){
    if(a>b)
        return a;
    else
        return b;
}
void main(){
    int a = 5, b = 8;
    cout<<"Max = "<<Max(a,b);
    cout<<"\nMax = "<<Max (a);
}
```

ចំណាំ៖

Default Value In Arguments យើងអាចបង្កើតលើតែ Arguments ណាដែលនៅចុងក្រោយដែលនៅចុងក្រោយគេ ឬក៏ Arguments ដែលនៅពីមុខ Defaults Value In Arguments ដូចគ្នា។

```
void main(int a, int b = 0, int c){
    cout<<"a = "<<a;
    cout<<"b = "<<b;
    cout<<"c = "<<c;
}
void main(){
    int a=1, b=2, c=3;
    Show(a, b, c);
}
```

2.5. Overloaded Functions:

ជាការបង្កើត Functions ច្រើនដែលមានឈ្មោះដូចគ្នា ហើយវាចាំបាច់ត្រូវខុសគ្នា
ដោយ Data type របស់ Arguments ឬក៏ចំនួន Arguments។

```
int div(int a, int b){
    return a/b;
}
int div(float a, float b){
    return a/b;
}
float div(float a, int b){
    return a/b;
}
float div(int a, float b){
    return a/b;
}
void main(){
    int a=5, b=2;
    float x=5, y=2;
    cout<<"Int div Int = "<<div(a,b);
    cout<<"Float div Float = "<<div(x,y);
    cout<<"Float div Int = "<<div(x,b);
    cout<<"Int div Float = "<<div(a,y);
}
```

```
float Max(float a, float b){
    if(a>b)
        return a;
    else
        return b;
}
float Max(float a, float b, float c){
    float max;
    max = Max(a,b);
    max = Max(max, c);
    return max;
}
void main(){
    float a=5, b=3,
    float x=10, y=12, z=6;
    cout<<"Max2 = "<<Max(a,b);
    cout<<"Max3 = "<<Max(x,y,z);
}
```

```
void Show(int a, int b){
```

```
        cout<<"\nA = "<<a;
        cout<<"\nB = "<<b;
    }
    void Show(int x, int y, int z = 0){
        cout<<"\nX = "<<x;
        cout<<"\nY = "<<y;
        cout<<"\nZ = "<<z;
    }
    void main(){
        int a = 5, b = 5;
        Show(a, b);
        int x = 10; y = 10, z = 10;
        Show(x, y, z);
    }
```

Run ទៅបាន Error

ចំណាំ៖

ប្រសិនបើយើងបង្កើត Overload នៃ Function យើងមិនគួរណាប្រើ Default Value In Argument ទេ ព្រោះវាអាចបណ្តាលឱ្យ Argument របស់ Overload នៃ Functions មានលក្ខណៈដូចគ្នា។

2.6. Recursive Function

ជា Function មួយដែលកោះហៅខ្លួនឯងមកប្រើ ដើម្បីដោះស្រាយបញ្ហា វានឹងកោះហៅរហូតទាល់តែបញ្ហានោះត្រូវបានដោះស្រាយ។

```
long factorial(int i){
    if(a<=1)
        return 1;
    else
        return n * factorial (n-1);
}
long factorial (int i){
    long f = 1;
    for(int i=1; i<=n; i++){
        f*=i;
    }
    return f;
}
void main(){
    int n = 5;
    cout<<n<<"! = "<<factorial(n);
}
```

សំគាល់៖

Factorial គឺជាការគុណតួគ្នា។

ឧទាហរណ៍៖

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$

```
cout<<factorial(5);
    5 * factorial(4);
        4 * factorial(3);
            3 * factorial(2);
                2 * factorial(1);
                    1
```

Enter $1 + 2 + 3 + 4 + 5 \dots n$

```
long Sum(int n){
    if(n<=1)
        return 1;
    else
        return n + Sum(n-1);
}
```

Enter $2 + 4 + 6 + 8 + 10 \dots 2n$

```
long Sum(int n){
    if(n<=2)
        return 2;
    else
        return 2 * n + Sum(n-1);
}
```

Enter $1 + 3 + 5 + 7 + 9 \dots 2n-1$

```
long Sum(int n){
    if(n<=1)
        return 1;
    else
        return 2 * n - 1 + Sum(n-1);
}
```

Enter $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots \frac{1}{n}$

```
float Sum(float n){
    if(n<=1)
        return 1;
    else
        return 1.0/n + Sum(n-1);
}
```

Enter $\frac{1}{1} + \frac{1}{3} + \frac{1}{5} + \dots \frac{1}{2n-1}$

```
float Sum(float n){
    if(n<=1)
```

```
        return 1;
    else
        return (1.0/(2*n-1)) + Sum(n-1);
}
```

Enter $\frac{2}{1} + \frac{4}{3} + \frac{6}{5} + \frac{8}{7} + \dots + \frac{2n}{2n-1}$

```
Float Sum(float n){
    If(n<=2)
        return 2;
    else
        return (2.0*n/(2*n-1)) + Sum(n-1);
}
```

26-11-2010

$1+2+3+4+5+\dots+n$

```
void display(int n){
    if(n<=1)
        cout<<"1 = ";
    else{
        cout<<" + ";
        display(n);
    }
}
long Sum(int n){
    if(n<=1)
        return 1;
    else
        return n+Sum(n-1)
}
void main(){
    clrscr();
    int n;
    cout<<"Enter n = ";
    cin>>n;
    display(n);
    cout<<Sum(n);
    getch();
}
```

$1!+2!+3!+4!+\dots+n!$

```
void display(int n){
    if(n<=1)
        cout<<"1!=";
    else{
        cout<<n<<"! + ";
        display(n-1);
    }
}
long factorial(int n){
    if(n<=1)
        return 1;
    else
        return n+factorial(n-1);
}
double Sum(int n){
    if(n<=1)
        return factorial(1);
    else{
        return factorial(n)+Sum(n-1);
    }
}
void main(){
    clrscr();
    int n;
    cout<<"Enter n=";
    cin>>n;
    display(n);
    cout<<Sum(n);
    getch();
}
```

2.7. Prototyping Function

ជាការប្រកាស Function ជាមុន ទើបបង្កើត Function នៅពេលក្រោយ។

```
#include<iostream.h>
#include<conio.h>
void odd(int);
void even(int);
void main(){
    clrscr();
    int n;
    do{
        cout<<"Type 0 to Exit: ";
        cin>>n;
    }while(n);
}
void odd(int n){
    if(n%2!=0)
        cout<<"This is odd number.";
    else
        even(n);
}
```

```
}  
void even(int n){  
    if(n%2==0)  
        cout<<"This is even number.";  
    else  
        odd(n);  
}
```

2.8. Variable Scope

សំដៅទៅលើដែនឥទ្ធិពលរបស់ Variable ។ Variable Scope ចែកជា២គឺ៖ Local Variable និង Global Variable។

2.8.1. Local Variable

ជា Variable ដែលត្រូវបានបង្កើតឡើងក្នុងតំបន់ណាមួយ ហើយ Variable នោះមានឥទ្ធិពលតែនៅក្នុងតំបន់នោះទេ។

```
#include<iostream.h>  
#include<conio.h>  
void Show();  
void main(){  
    int n;  
    n=s;  
    cout<<"\nN1 = "<<n;  
    Show();  
}  
void main(){  
    n +=3;  
    cout<<"\nN2 = "<<n;  
}
```

Output: Error

```
#include<iostream.h>  
#include<conio.h>  
void main(){  
    int x;  
    x=s;  
    {  
        int y=2;  
        x +=y;  
    }  
    cout<<"\nX = "<<x;  
    cout<<"\nY = "<<y;  
}
```

Output: Error

2.8.2. Global Variable ៖ ជា Variable ដែល Declare នៅក្រៅ Function ហើយ Variable នោះ មានឥទ្ធិពលទាំងក្នុង Main Function និងគ្រប់ Functions ទាំងអស់របស់ Main Programs ។

```
#include<iostream.h>
#include<conio.h>
int n;
void Show();
void main(){
    n=5;
    cout<<"\nN1 = "<<n;
    Show();
}
void Show(){
    n +=3;
    cout<<"\nN2 = "<<n;
}
```

Output: N1 = 5
N2 = 8

One more:

```
#include<iostream.h>
#include<conio.h>
int n;
#include<Sub.h>
void main(){
    n=5;
    cout<<"\nN1 = "<<n;
    n +=3;
    Show1();
    n +=4;
    Show2();
    cout<<"\nN4 = ";
}
Sub.h
Void Show1(){
    n +=4;
    cout<<"\nN2 = "<<n;
}
void Show2(){
    cout<<"\nN3 = "<<n;
    n +=6;
}
```

Output: N1 =5
N2 = 12
N3 = 16

N4 = 22

ដោយសារ n តែមួយគឺបូកបញ្ចូលគ្នាហួត

សំនួរ៖

1. តើការសរសេរ Program ដោយប្រើ Function មានសារៈសំខាន់អ្វីខ្លះ?
ការសរសេរ Program ដោយប្រើ Function មានសារៈសំខាន់ដូចខាងក្រោម៖
 - ស្រួលសរសេរ Code
 - ស្រួលអាន និងរុករកកំហុស
 - ស្រួលកែតម្រូវ និងផ្លាស់ប្តូរ
 - យើងសរសេរ Code តែម្តង អាចកោះហៅយកមកប្រើប្រាស់បានច្រើនដង និងច្រើនទិសដៅ
 - យើងអាចហៅ Function យកមកប្រើនៅក្នុង Function របស់ខ្លួនឯង
2. តើ Function ចែកចេញជាប៉ុន្មានប្រភេទ? អ្វីខ្លះ?
Function ចែកចេញជា២ប្រភេទគឺ៖
 - Function មាន Return type
 - Function គ្មាន Return type
3. តើយើងសំគាល់ Function ដែលមាន Return type ត្រង់ចំណុចណាខ្លះ? ប្រើសំរាប់ធ្វើអ្វី?
 - យើងសំគាល់ Function ដែលមាន Return type ត្រង់នៅពីមុខ Function name មាន Data type និងនៅក្នុង Function មាន Return Value ។
 - គេប្រើ Function ដែលមាន Return type សំរាប់គណនា ឬរុករកតម្លៃអ្វីមួយ។
4. តើយើងសំគាល់ Function ដែលគ្មាន Return type ត្រង់ចំណុចណាខ្លះ? ប្រើសំរាប់ធ្វើអ្វី?
 - យើងសំគាល់ Function ដែលគ្មាន Return type ត្រង់នៅពីមុខ Function name មាន Keyword void ហើយវាគ្មាន Return Value ទេ។
 - គេប្រើវាសំរាប់ Output ឬ Display
5. តើការបញ្ជូនទិន្នន័យពី Parameter ទៅ Argument បានប៉ុន្មានយ៉ាង? អ្វីខ្លះ?
ការបញ្ជូនទិន្នន័យពី Parameter ទៅ Argument បាន២យ៉ាង៖
 - Value Parameter
 - Reference Parameter
6. តើយើងសំគាល់ Value Parameter ត្រង់ចំណុចណា? ប្រើធ្វើអ្វី?
 - យើងសំគាល់ Value Parameter ត្រង់គ្មានសញ្ញាអាសយដ្ឋាន (&)។ គេប្រើវាសំរាប់បញ្ជូនត្រឹមតែតម្លៃទៅឲ្យ Argument ហើយមិនត្រូវការតម្លៃពី Argument ត្រលប់មកវិញទេ ទោះបី Argument ប្រែប្រួលតម្លៃក៏ដោយ។
7. តើយើងសំគាល់ Reference Parameter ត្រង់ចំណុចណា? ប្រើធ្វើអ្វី?

- យើងសំគាល់ Reference Parameter ត្រង់មានសញ្ញាអាសយដ្ឋាន (&) ។ គេប្រើវាសំរាប់បញ្ជូនតម្លៃ ឬអាសយដ្ឋាន របស់ Parameter ទៅឲ្យ Argument ហើយគេត្រូវការតម្លៃត្រលប់មកវិញ។
- 8. ចូរឲ្យនិយមន័យ Default Value In Argument
 - Default Value In Argument ជាការកំណត់តម្លៃទៅឲ្យ Argument ធ្វើឲ្យ Argument ត្រូវការតម្លៃក៏បាន មិនត្រូវការក៏បាន។
- 9. ចូរឲ្យនិយមន័យ Overloaded Function។ ប្រើដើម្បីអ្វី?
 - Overloaded Function គឺជាការបង្កើត Function ច្រើនដូចគ្នា តែខុសគ្នាត្រង់ចំនួន ឬ Data type នៃ Argument។ គេបង្កើតវាដើម្បីឲ្យយើងគិតថា Function មានតែមួយ ហើយមានតួនាទីច្រើនយ៉ាង។
- 10. ហេតុអ្វីបានជាគេមិនតម្រូវឲ្យប្រើ Overloaded Function ជាមួយ Default Value in Argument?
 - ព្រោះវាឲ្យ Function ទាំងអស់មានលក្ខណៈដូចគ្នា ដោយសារតែ Default Value in Argument អាចបណ្តាលឲ្យចំនួន Argument ស្មើគ្នា។
- 11. ដូចម្តេចដែលហៅថា Variable Scope? តើមានប៉ុន្មានប្រភេទ?
 - Variable Scope គឺសំដៅទៅលើដែនឥទ្ធិពលរបស់ Variable។
 - Variable Scope មាន២ ប្រភេទ៖
 1. Local Variable
 2. Global Variable
- 12. ចូរឲ្យនិយមន័យ Local Variable។ តើវាមានប្រយោជន៍អ្វី?
 - Local Variable គឺជាអញ្ញាតដែលត្រូវបានបង្កើតនៅក្នុងតំបន់ណាមួយ ហើយវាមានឥទ្ធិពលតែនៅក្នុងតំបន់នោះតែប៉ុណ្ណោះ។
 - វាមានប្រយោជន៍ត្រង់ចំណេញទំហំ Memory ព្រោះ Variable នឹងរលាយបាត់ពី Memory នៅពេលតំបន់របស់វាត្រូវបាន execute ចប់។
- 13. ចូរឲ្យនិយមន័យ Global Variable។ តើវាមានប្រយោជន៍អ្វី?
 - Global Variable ជា Variable ដែលត្រូវបានបង្កើតឡើងនៅក្រៅ Function។ ហើយ Variable នោះមានឥទ្ធិពលនៅគ្រប់ទីកន្លែងនៃ Program។
 - វាពិតជាមានប្រយោជន៍ ព្រោះវាមិនចាំបាច់ប្រើ Parameter និង Argument

14. ចូរពន្យល់ Code ខាងក្រោម៖

```
#include<iostream.h>
#include<conio.h>
int n;
#include<Sub.h>
void main(){
    n = 5;
    cout<<"\nN1 = "<<n;
    {
        int n = 2;
        n +=1;
        Show1();
        cout<<"\nN3 = "<<n;
    }
}
void Show1(){
    n +=2;
    cout<<"\nN2 = "<<n;
}
void main(){
    cout<<"\nN4 = "<<n;
    n +=3;
}
```

ដោយបង្ហាញលទ្ធផលដូចខាងក្រោម៖

```
N1 = 5
N2 = 7
N3 = 3
N4 = 11
N5 = 14
```


Chapter 06:

Array

1. What is array?

Array ជាបណ្តុំ Variables ដែលអាចផ្ទុកទិន្នន័យបានច្រើន ហើយវាមានឈ្មោះដូចគ្នា និង Data type ដូចគ្នា ហើយពួកវាខុសគ្នាទៅដោយ Index ។

2. Declaration of array

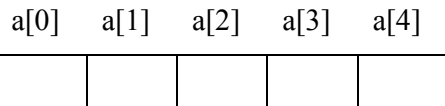
Form:

Data type ArrayName[Elements];

e.g.: int a[5];

ហៅថាចំនួនធាតុរបស់ Array

ហៅថា Index



ចំណាំ:

- Element ជាចំនួនធាតុរបស់ Array ។ យើងមិនអាចប្រើ Variable សំរាប់កំណត់ចំនួនធាតុរបស់ Array បានទេ ចំពោះចំនួនធាតុរបស់ Array ចាំបាច់ត្រូវតែជាតម្លៃថេរ។
- Index ជាលេខរៀងនៃធាតុនីមួយៗរបស់ Array ។
- នៅក្នុងភាសាកម្មវិធី C++ Element មិនអាចជាអថេរបានទេ តែនៅក្នុងភាសាកម្មវិធី Java វិញ អាចប្រើជាអថេរបាន។
- Index មិនមែនជាការកំណត់ចំនួនរបស់ Array ទេ ។

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
void main(){
```

```
    clrscr();
```

```
    int n;
```

```
    int a[100];
```

```
    cout<<"Enter number of elements = ";
```

```
    cin>>n;
```

```
cout<<"\nNow, you can enter "<<n<<" values";
for(int i=0; i<n; i++){
    cout<<"\nEnter a["<<i<<"] = ";
    cin>>a[i];
}
cout<<"\nNow, you can Display "<<n<<" values";
for(i=0; i<n; i++)
    cout<<"a["<<i<<"] = "<<a[i]<<endl;
getch();
}
```

MAXIMUM

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    int a[100];
    cout<<"Enter number of elements = ";
    cin>>n;
    cout<<"\nNow, you can enter "<<n<<" values";
    for(int i=0; i<n; i++){
        cout<<"\nEnter a["<<i<<"] = ";
        cin>>a[i];
    }
    int max;
    max = a[0];
    for(i=0; i<n; i++)
        if(max<a[i]) /* Find the bigger number */
            max=a[i];
    cout<<"Maximum is "<<max;
    getch();
}
```

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    int a[100];
    cout<<"Enter number of elements = ";
    cin>>n;
    cout<<"\nNow, you can enter "<n<<" values";
    for(int i=0; i<n; i++){
        cout<<"\nEnter a["<i<<" = ";
        cin>>a[i];
    }
    int max;
    max = a[0];
    for(i=0; i<n; i++)
        if(max>a[i]) /* Find the lesser number */
            max=a[i];
    cout<<"Minimum is "<max;
    getch();
}
```

a[0]>a[1]
a[1]>a[2]
a[2]>a[3]
a[3]>a[4]

a[0]>a[1]
a[1]>a[2]
a[2]>a[3]

a[0]>a[1]
a[1]>a[2]

a[0]>a[1]

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    int a[100];
    cout<<"Enter number of elements = ";
    cin>>n;
    cout<<"\nNow, you can enter "<<n<<" values";
    for(int i=0; i<n; i++){
        cout<<"\nEnter a["<<i<<"] = ";
        cin>>a[i];
    }
    cout<<"\nNow, you can Display "<<n<<" loop values";
    for(i=1; i<n; i++)
        for(int j=0; j<n-i; j++)
            if(a[j]>a[j+1]){
                int temp;
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
    for(i=0; i<n; i++)
        cout<<"a["<<i<<"] = "<<a[i]<<endl;
    getch();
}

#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    int a[100];
    cout<<"Enter number of elements = ";
    cin>>n;
```

```
cout<<"\nNow, you can enter "<<n<<" values";
for(int i=0; i<n; i++){
    cout<<"\nEnter a["<<i<<"] = ";
    cin>>a[i];
}
cout<<"\nList all category number";
int negative = 0, smaller9 = 0, smaller19 = 0, smaller29 = 0, over30 = 0;
for(i=0; i<n; i++){
    if(a[i]<0)
        negative++;
    else if(a[i]<=9)
        smaller9++;
    else if(a[i]<=19)
        smaller19++;
    else if(a[i]<=29)
        smaller29++;
    else
        over30++;
}
cout<<"Negative = "<<negative;
cout<<"\nSmaller9 = "<<smaller9;
cout<<"\nSmaller19 = "<<smaller19;
cout<<"\nSmaller29 = "<<smaller29;
cout<<"\nOver30 = "<<over30;
getch();
}
```

Enter number of elements = 5

A[0] =

A[1] =

A[2] =

A[3] =

A[4] =

Enter number you need to Search =

Index ... found in Array

Enter number you need to Search =

Index not found in Array

Writing code:

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    int a[100];
    cout<<"Enter number of elements = ";
    cin>>n;
    cout<<"\nNow, you can enter "<n<<" values ";
    for(int i=0; i<n; i++){
        cout<<"\nEnter a["<i<<" = ";
        cin>>a[i];
    }
    int search;
    cout<<"\nEnter number you need to Search = ";
    cin>>search;
    cout<<"Index";
    int Boolean=0;
    for(i=0; i<n; i++){
        if(a[i]==search){
            boolean=1;
            cout<<" , ";
        }
    }
    if(boolean==1)
        cout<<"found in Array";
    else
        cout<<"Not found in Array";
    }
    getch();
}
```

ដំណើរការរកការពិតនៅទីនេះ

How to write code having existed number

Enter number of elements = 5

Enter a[0] = 43

Enter a[1] = 43

Number already exist

Enter a[1] = 23

Enter a[2] = 43

Number already exist

a[0] = 43

a[1] = 23

a[2] = 24

a[3] = 1

a[4] = 5

Writing code:

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int n;
    int a[100];
    cout<<"Enter number of elements = ";
    cin>>n;
    cout<<"\nNow, you can enter "<<n<<"values but not duplicate";
    for(int i=0; i<n; i++){
        if(i==0){
            cout<<"\nEnter a[0] = ";
            cin>>a[0];
        }else{
            cout<<"\nEnter a["<<i<<"] = ";
            int Boolean=0;
            int value;
            cin>>value;
            for(int j=0; j<i; j++)
```

```
        if(a[j]==value){
            cout<<"\nNumber already exist.";
            Boolean=1;
            break;
        }
        if(Boolean==0)
            a[i]=value;
    }
}

cout<<"\nNow, you can display " << n << " values but not duplicate.";
for(i=0; i<n; i++)
    cout<<"\na[" << i << "] = " << a[i] << endl;
getch();
}
```

3. **Multidimensional Array:** សំដៅលើ Array ច្រើនវិមាត្រ។ ជាទូទៅ Multidimensional Array គេច្រើនប្រើ២វិមាត្រយ៉ាងច្រើន៖

Form:

DataType ArrayName[element1][element2];

ឧទាហរណ៍ទី១៖

int a[2][3];

↖ ជា Row
↗ ជា Column

a[0][0]	a[0][1]	a[0][2]
a[1][0]	a[1][1]	a[1][2]

```
#include<iostream.h>
#include<conio.h>
#include<dos.h>
void main(){
```



```
clrscr();
int r, c;
int a[100][100];
cout<<"Enter rows = ";
cin>>r;
cout<<"Enter columns = ";
cin>>c;
cout<<"\nNow, you can enter " <<r*c<<"values\n";
for(int i=0; i<r; i++){
    for(int j=0; j<c; j++){
        cout<<"Enter a["<<i<<"]["<<j<<"] = ";
        cin>>a[i][j];
    }
    cout<<"\n";
}
cout<<"Display Table " <<r<<"Rows " <<c<<" Columns \n";
for(i=0; i<r; i++){
    for(int j=0; j<c; j++){
        delay(1000);
        cout<<a[i][j]<<"\t";
    }
    cout<<endl;
}
getch();
}
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
#include<dos.h>
```

```
void main(){
```

```
    clrscr();
```

```
    int r, c;
```

```
    int a[100][100];
```

```
    cout<<"Rows = ";
```

```
cin>>r;
cout<<"Columns = ";
cin>>c;
cout<<"\nNow, you can enter "<<r*c<<" values\n";
for(int i=0; i<r; i++){
    for(int j=0; j<c; j++){
        cout<<"\nEnter a["<<i<<"]["<<j<<"] = ";
        cin>>a[i][j];
    }
    cout<<"\n";
}
cout<<"Display Table"<<r<<"Rows"<<c<<"Columns and its calculation by row\n";
int cal=0;
for(i=0; i<r; i++){
    for(int j=0; j<c; j++){
        delay(1000);
        cout<<a[i][j]<<"\t";
        cal+=a[i][j];
    }
    cout<<cal;
    cout<<endl;
}
getch();
}
```

```
#include<iostream.h>
#include<conio.h>
#include<dos.h>
void main(){
    clrscr();
    int r, c;
    int a[100][100];
    cout<<"Rows = ";
    cin>>r;
```

```
cout<<"Columns = ";
cin>>c;
cout<<"\nNow, you can enter " <<r*c<<" values\n";
for(int i=0; i<r; i++){
    for(int j=0; j<c; j++){
        cout<<"\nEnter a["<<i<<"]["<<j<<"] = ";
        cin>>a[i][j];
    }
    cout<<endl;
}
cout<<"Display Table " <<r<<"Rows " <<c<<"Columns and its calculation \n";
int cal=0;
for(i=0; i<r; i++){
    for(int j=0; j<c; j++){
        delay(1000);
        cout<<a[i][j]<<"\t";
        cal+=a[i][j];
    }
    cout<<cal;
    cout<<endl;
}
int total=0;
for(int j=0; j<c; j++){
    int cal=0
    for(i=0; i<r; i++){
        cal+=a[i][j];
        total+=cal;
        cout<<cal<<"\t";
    }
}
cout<<total;
getch();
}
```

```
#include<iostream.h>
```

```
#include<conio.h>

void main(){
    clrscr();
    int r, c;
    int a[100][100];
    cout<<"Row = ";
    cin>>r;
    cout<<"Columns = ";
    cin>>c;
    cout<<"\nNow, you can enter " <<r*c<<" values\n";
    for(int i=0; i<r; i++){
        for(int k=0; k<c; k++){
            cout<<"\nEnter a["<<i<<"]["<<j<<"] = ";
            cin>>a[i][k];
        }
        cout<<"\n";
    }
    cout<<"Display Table " <<r<<"Rows " <<c<<" Columns\n";
    int cal=0;
    for(i=0; i<r; i++){
        int=0;
        for(int k=0; k<c; k++){
            cal+=a[i][k];
            cout<<a[i][k]<<"\t";
        }
        cout<<endl;
        cout<<endl;
    }
    int calcul[100];
    for(int k=0; k<c; k++){
        calcul[k]=0;
        for(int i=0; i<r; i++)
            calcul[k]+=a[i][k];
    }
```

```
int cal=0;
for(k=0; k<c; k++){
    cout<<calcu[k]<< "\t";
    cal+=calcu[k];
}
cout<<cal;
getch();
}
```

4. Character Array

យើងអាចបង្កើតទិន្នន័យជា String បាន ដោយយើងចាំបាច់ត្រូវតែប្រើ Character Array ។

ឧទាហរណ៍៖

```
char str1[100]= "C++";
char str2[]="C++";
char str3[100]={ 'C', '+', '+', '\0' };
char str3[]={ 'C', '+', '+', '\0' };
//ជាលេខសូន្យ
```

🚧 ចំណាំ៖

យើងអាច Initialize ទិន្នន័យជា String ឲ្យទៅ Character Array បាន ឬមួយក៏ប្រើ cin ដើម្បីបញ្ចូលទិន្នន័យពី Keyboard ឲ្យទៅ Character Array ក៏បានដែរ តែយើងមិនអាច Assign ទិន្នន័យទៅឲ្យ Character បានទេ។ ចំពោះដំណោះស្រាយគឺយើងត្រូវបញ្ចូល Header file មួយឈ្មោះថា #include<string.h> ចំពោះ Header file នេះមានតួនាទីជួយសំរួលរាល់ការ execute ទិន្នន័យដែលមានប្រភេទជា String។

- void strcpy(destination, source): ជា function មានតួនាទី copy ទិន្នន័យរបស់ String source ឲ្យទៅ String Destination។

ឧទាហរណ៍ទី១៖

```
char men[100];
strcpy(men, "very good");
//men=very good
```

ឧទាហរណ៍ទី២៖

```
char str1[100]= "C++";
char str2[100];
str2=str1;
strcpy(str2, str1);
```

```
//str2=str1
```

- `int strlen(String)` ជា function មានតួនាទីរាប់ចំនួន Character របស់ String ណាមួយ។

ឧទាហរណ៍៖

```
char str[100] = "C++ Programming";  
int len;  
len=strlen(str);  
//len=15
```

- `int strcmp(String1, String2)` ជា function មានតួនាទីប្រៀបធៀប String ចំនួន២ បើ៖

1. `return = 0` នៅ៖ `String1 = String2`
2. `return > 0` នៅ៖ `String1 > String2`
3. `return < 0` នៅ៖ `String1 < String2`

ឧទាហរណ៍៖

```
char str1[100] = "C++";  
char str2[100] = "Java";  
int cmp = strcmp (str1, str2);  
if(cmp==0)  
    cout<<"str1==str2";  
else if(cmp>0)  
    cout<<"str1>str2";  
else if(cmp<0)  
    cout<<"str1<str2";
```

ឧទាហរណ៍៖

```
#include<iostream.h>  
#include<conio.h>  
void main(){  
    clrscr();  
    char ch;  
    do{  
        cout<< "\nPress key = ";  
        ch=getch();  
        cout<< "\n"<<ch<< "\t" <<int(ch);
```

```
    }while(ch!=13);  
    getch();  
}
```

✚ getch មានន័យថា ទទួល 1 character ពី keyboard ។

✚ Code របស់គូអក្សរហៅថា SCII Code ។

✚ String គឺត្រូវបាន assign តាមរយៈ Index ។

✚ '\0' មានតួនាទីបញ្ចប់ String ។

```
#include<iostream.h>  
#include<conio.h>  
#include<string.h>  
void main(){  
    clrscr();  
    char pass[100], passIn[100]= "3x4";  
    char ch; int i=0;  
    cout<< "\Press key = ";  
    do{  
        ch=getch();  
        if(ch==8){  
            if(i>0){  
                cout<< "\b\b";  
                i--;  
            }else{  
                pass[i]=ch;  
                cout<< "*";  
                i++;  
            }  
        }  
    }while(ch!=13);  
    cout<< "\b\b";  
    pass[i-1]= '\0';  
    int cmp;  
    cmp=strcmp(pass, passIn);  
    if(cmp==0){
```

```
        cout<< "\nCorrect Password";
    }else
        cout<< "\nIncorrect Password";
    getch();
}
#include<iostream.h>
#include<conio.h>
#include<string.h>
void main(){
    clrscr();
    char pass[100], passIn[100]= "3x4";
    char ch; int i=0;
    cout<< "\Press key = ";
    do{
        ch=getch();
        if(ch==8){
            if(i>0){
                cout<< "\b\b";
                i--;
            }else{
                if(i<8){
                    pass[i]=ch;
                    cout<< "*";
                    i++;
                }
            }
        }
    } while(ch!=13);
    if(i==8){
        pass[i]= '\0';
    }else{
        cout<< "\b";
        pass[i-1]= '\0';
    }
}
```



```
        getch();
    }

#include<iostream.h>
#include<conio.h>
#include<string.h>
void main(){
    clrscr();
    label:
    char pass[100], passIn[100]= "3x4";
    char ch; int i=0;
    cout<< "\Press key = ";
    do{
        ch=getch();
        if(ch==8){
            if(i>0){
                cout<< "\b\b";
                i--;
            }else{
                pass[i]=ch;
                cout<< "*";
                i++;
            }
        }
    }while(ch!=13);
    cout<< "\b\b";
    pass[i-1]= '\0';
    int cmp;
    cmp=strcmp(pass, passIn);
    if(cmp==0){
        cout<< "\nCorrect Password";
    }
```

```
    }else
        cout<< "\nIncorrect Password";
        goto label;
    getch();
}
```

```
TC\INCLUDE\sh5pass.h
void password(char pass[]){
    char ch; int i=0;
    do{
        ch=getch();
        if(ch==8){
            if(i>0){
                cout<< "\b \b";
                i--;
            }
        }else{
            cout<< "*";
            pass[i]=ch;
            i++;
        }
    }while(ch!=13);
    cout<< "\b";
    pass[i-1]= '\0';
}

#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<sh5pass.h>
void main(){
    clrscr();
    char usr[100], usrIn[100]=<< "Sok Rotha";
    char pass[100], passIn[100]= "3x4";
```

```
cout<< "Enter UserName = ";
password(pass);
int cmp1, cmp2;
cmp1=strcmp(usr, usrIn);
cmp2=strcmp(pass, passIn);
if(cmp1==0&&cmp2==0){
    cout<< "\nCorrect Account";
}else{
    cout<< "\nIncorrect UserName or Password";
}
getch();
}
```

- char* strlwr(string); ជា function មួយមានតួនាទីសំរាប់ Convert ទិន្នន័យនៅក្នុង String ទៅជាតួអក្សរតូចទាំងអស់។

ឧទាហរណ៍៖

```
char st[100]= "Welcome to My Program";
cout<< "st = "<<strlwr(st);
//st = welcome to my program
```

ឬម្យ៉ាងទៀត

```
char st[100]= "Welcome to My Program";
strlwr(st);
cout<< "st = "<<st;
//st = welcome to my program
```

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<sh5pass.h>
void main(){
    clrscr();
    char usr[100], usrIn[100]=<< "Sok Rotha";
```

```
char pass[100], passIn[100]= "3x4";
cout<< "Enter UserName = ";
password(pass);
int cmp1, cmp2;
cmp1=strcmp(strlwr(usr), usrIn);
cmp2=strcmp(pass, passIn);
if(cmp1==0&&cmp2==0){
    cout<< "\nCorrect Account";
} else {
    cout<< "\nIncorrect UserName or Password";
}
getch();
}
```

- char*strupr(string); ជា function មួយមានតួនាទីសំរាប់ Convert ទិន្នន័យនៅក្នុង String ទៅជាតួអក្សរធំទាំងអស់។

ឧទាហរណ៍៖

```
char st[100]= "Welcome to My Program";
cout<< "st = "<<strupr(st);
//st = WELCOME TO MY PROGRAM
```

ឬម្យ៉ាងទៀត

```
char st[100]= "Welcome to My Program";
strupr(st);
cout<< "st = "<<st;
//st = WELCOME TO MY PROGRAM
```

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<sh5pass.h>
void main(){
    clrscr();
```

```
char usr[100], usrIn[100]=<< "Sok Rotha";
char pass[100], passIn[100]= "3x4";
cout<< "Enter UserName = ";
password(pass);
int cmp1, cmp2;
cmp1=strcmp(strupr(usr), usrIn);
cmp2=strcmp(pass, passIn);
if(cmp1==0&&cmp2==0){
    cout<< "\nCorrect Account";
} else{
    cout<< "\nIncorrect UserName or Password";
}
getch();
}
```

- stdlib.h ជា header file ដែលត្រូវបានគេហៅថាជា library។ វាមាន command មួយចំនួនដូចខាងក្រោម៖
 - void exit(0); មានតួនាទីសំរាប់ចាកចេញពីកម្មវិធី។
 - int atoi(string); ជា function មួយមានតួនាទីសំរាប់ Convert ទិន្នន័យមកជា Integer។ atoi ពាក្យពេញរបស់វាគឺ autointeger។

ឧទាហរណ៍៖

```
char st[100]= "123";
int num;
num atoi(st);
cout<< "Number is "<<num;
num=num+2;
cout<< "\nNumber is "<<num;
```

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
```

```
void main(){
    clrscr();
    char st[100];
    cout<< "Enter string number = ";
    cin>>st;
    int num;
    num=atoi(st);
    cout<< "\nNumber is "<<num;
    getch();
}
```

TC\INCLUDE\bestnum.h

```
long bestNum(int len=3){
    char ch; int i=0;
    char st[100];
    do{
        ch=getch();
        if(ch==8){
            if(i>0){
                cout<< "\b\b";
                i--;
            }
        }else if(i<len)
            if(ch=='0' || ch == '1' ch == '2' || ch == '3' ch == '4' || ch== '5'
            || ch == '6' || ch == '7' ch == '8' || ch == '9'){
                cout<<ch;
                st[i]=ch;
                i++;
            }
    }while(ch!=13||i==0)← ទាល់តែបញ្ចូលទិន្នន័យរួចហើយចុច Enter បាន
    st[i]= '\0';
    return atoll(st);
}
```

```
}

#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#include<bestnum.h>
void main(){
    clrscr();
    int num;
    cout<< "Enter String Number = ";
    num=bestnum();
    int b[100], i=0;
    cout<< "\nBinary = ";
    do{
        int n;
        n=num%2;
        b[i]=n;
        i++;
    }while(num=num/2);
    for(int j=i-1; j>=0; j--)
        cout<< b[j];
    getch();
}

- long atoll(string);
- double atof(string);
```

🚩 Random numbers ជាលេខដែលកើតឡើងដោយចៃដន្យ ដើម្បីបង្កើត Random Numbers បាន យើងចាំបាច់ត្រូវតែ Include Header File មួយឈ្មោះថា <stdlib.h> ជាមុនសិន។

- void randomize(): ជា Function មួយមានតួនាទីប្រាប់ទៅ System ថា យើងត្រូវការ Random Numbers ។
- int rand(): ជា Function មួយមានតួនាទីបង្កើត Random Numbers ។

ឧទាហរណ៍៖

```
randomize();
```

```
int n;
n=rand();
cout<< "Number is "<<n;
// Number is ?????

#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
    clrscr();
    randomize();
    int n;
    n=rand();
    cout<< "Number is "<<n;
    getch();
}
```

- int random(int elements): ជា function មួយមានតួនាទីបង្កើត random numbers តែយើងអាចកំណត់ចំនួនដែលយើងត្រូវការបាន។

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
    clrscr();
    randomize();
    int n;
    n=random(100); (វា random ពី 0 ទៅ 99)
    cout<< "Number is "<<n;
    getch();
}
```

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
```



```
        clrscr();
        randomize();
        int n;
        n=random(90)+1; (random ကို 1 မှ 90)
        cout<< "Number is "<<n;
        getch();
    }
```

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
    clrscr();
    randomize();
    int usr;
    int sys;
    cout<< "\nSmall=0    Big=1\n";
    cin>>usr;
    sys=random(2);
    if(usr==sys)
        cout<< "You win"<<sys;
    else
        cout<< "You lose"<<sys;
    getch();
}
```

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
    clrscr();
    randomize();
    int usr;
    int sys;
```

```
do{
    cout<< "\nSmall=0   Big=1\n";
    cin>>usr;
    sys=random(2);
    if(usr==sys)
        cout<< "You win"<<sys;
    else
        cout<< "You lose"<<sys;
} while(usr==1||usr==0);
getch();
}
```



```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
    clrscr();
    randomize();
    int usr;
    int sys;
    do{
        cout<< "\nSmall=0   Big=1\n";
        cin>>usr;
        sys=random(2);
        if(usr==sys){
            sys=random(2);
            if(usr==sys)
                cout<< "You win"<<sys;
            else
                cout<< "You lose"<<sys;
        }
        else
            cout<< "You lose"<<sys;
    } while(usr==1||usr==0);
    getch();
}
```

}

Random(1000)

0 – 49

50 – 149

150 – 299

300 – 599

600 – 999

1	2	3	4	5
5%	10%	15%	30%	40%
6	3.5	3	2.3	2

Enter Number = 1

System = 1

You win!

The End!

Chapter 07:

Pointer

1. What is pointer?

Pointer គឺជា variable ដែលមាន address ប្រែប្រួល។

2. Declaration of pointer

Pointer ជាអថេរដែលប្រែប្រួល (address)។

Form:

```
DataType *pointerName;
```

e.g.:

```
int *age;
```

```
float *amount;
```

ចំណាំ:

- ពាក្យថា *age, *amount ជាតម្លៃដែលត្រូវបានរក្សាទុកនៅក្នុង Pointer ។
- ពាក្យថា age, amount ជា address របស់ pointer ដែលស្ថិតនៅលើ Memory ។

3. Pointer can refer with other variable

e.g1.:

```
int a;  
a=10;  
int *p;  
p=&a; // Reference variable with pointer  
*p+=2;  
cout<<a; // a=12;
```

e.g2.:

```
Data obj1;  
obj1.a=10;  
obj1.b=10;  
Data *p;  
p=&obj1; // Reference object with pointer  
(*p).a+=2;  
(*p).b+=2;
```

```
cout<< "obj1"<<obj1.a<< " "<<obj1.b;
```

- ✚ Address គឺសំដៅទៅលើការសិក្សាលើផ្ទៃ RAM គឺតាមប្រព័ន្ធគោល 16 ។
- ✚ Pointer ចង្អុលលើ Address ។
- ✚ *p ទាញតម្លៃ។

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int a, b;
    int *p;
    a=10;
    b=20;
    cout<< "Address A = "<<&a<<endl;
    cout<< "Address B = "<<&b<<endl;
    cout<< "Address P = "<<p<<endl<<endl;

    p=&a; // Assign Address "a" to pointer "p"
    *p+=5;
    cout<< "Address A = "<<&a<<endl;
    cout<< "Address B = "<<&b<<endl;
    cout<< "Address P = "<<p<<endl;
    cout<< "Value A = "<<a<<endl<<endl;

    p=&b; // Assign Address "b" to pointer "p"
    *p+=6;
    cout<< "Address A = "<<&a<<endl;
    cout<< "Address B = "<<&b<<endl;
    cout<< "Address P = "<<p<<endl;
    cout<< "Value B = "<<b<<endl<<endl;
    getch();
}
```

✚ .free(pointer); ជា function ដែលអាច free address របស់ pointer ដើម្បីប្រើ function នោះបាន យើងចាំបាច់ត្រូវបញ្ចូល #include<stdlib.h> ។

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
    clrscr();
    int a;
    a=10;
    int *p;
    p=&a;
    *p+=5;
    cout<< "Value A = "<<a<<endl;
    cout<< "Value *p = "<<*p<<endl<<endl;

    free(p); // Free address "a" and pointer "p"
    // Not recommend to use "a" and "p"
    cout<< "Value A = "<<a<<endl;
    cout<< "Value *p = "<<*p<<endl<<endl;
    getch();
}
```

4. Using pointer with Array

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    int a[]={1, 2, 3, 4, 5};
    int *p;
    p=&a; // Assign Address array "a" to pointer "p"
    for(int i=0; i<n; i++){
        cout<< "*p = "<<*p<<endl;
```

```
        p++;  
    }  
    getch();  
}
```

5. Using pointer as Array N elements

យើងអាចប្រើ pointer ជំនួស array បាន ហើយថែមទាំងអាចបង្កើត array N ធាតុទៀតផង ដើម្បីបង្កើត pointer ជា array N ធាតុបាន យើងត្រូវបញ្ចូល function មួយឈ្មោះថា malloc ជាចាំបាច់ដែលស្ថិតនៅក្នុង #include<stdlib.h>។

Function malloc មានតួនាទីស្វែងរក free address សំរាប់ pointer ។

```
#include<iostream.h>  
#include<conio.h>  
#include<stdlib.h>  
void main(){  
    clrscr();  
    int n;  
    cout<< "Enter n = ";  
    cin>>n;  
    int *p, *first;  
  
    p=(int *) malloc(n);  
    first=p; // Set first address of pointer  
    for(int i=0; i<n; i++){  
        cout<< "*p = ";  
        cin>>*p;  
        p++;  
    }  
  
    p=first; // Move pointer to first address  
    cout<< "\n\nDisplay all values\n";  
    for(i=0; i<n; i++){  
        cout<< "*p = "<<*p<<endl;
```

```
        p++;
    }
    free(p); // Free all address of pointer
    getch();
}
```

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
    clrscr();
    int n;
    cout<< "Enter n = ";
    cin>>n;
    int *p;
    p=(int *)malloc(n);
    for(int i=0; i<n; i++){
        cout<< "p[<<i<<"] = ";
        cin>>p[i];
    }

    cout<< "\n\nDisplay all values\n";
    for(i=0; i<n; i++){
        cout<< "p[<<i<<"] = "<<p[i]<<endl;
    }

    free(p); // Free all address of pointer
    getch();
}
```

ចំណាំ៖

គុណសម្បត្តិ៖

- Pointer អាច reference គ្រប់ variable និង objects ទាំងអស់ដែលមានប្រភេទទិន្នន័យដូចគ្នា។

```
#include<iostream.h>
#include<conio.h>
class Data{
    public:
    int a, b;
};
void main(){
    clrscr();
    int a;
    int *p;
    p=&a; // Pointer reference variable
    *p = 10;
    a = 10; // The same use *p or a

    Data obj;
    Data *pobj;
    pobj=&obj; // Pointer reference object
    obj.b = 10;
    (*pobj).b = 10; // The same use *pobj or obj

    int arr[] = {,1 2, 3, 4, 5};
    int *pArr;
    pArr = arr; // Pointer reference array

    arr[0] = 30;
    pArr[0] = 30; // The same use pArr or arr

    getch();
}
```

- ការប្រើ Pointer សន្សំសំចៃ Memory ព្រោះ Pointer អាចបង្កើត Array N ធាតុបាន។

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void main(){
    clrscr();
    int n;
    cout<< "Enter n = ";
    cin>>n;
    int *p;
    p=(int *)malloc(n); // Create N elements
    for(int i=0; i<n; i++){
        cout<< "p[<<i<<"] = ";
        cin>>p[i];
    }
    getch();
}
```

❖ គុណវិបត្តិនៃការប្រើប្រាស់ Pointer

- ទិន្នន័យដែលរក្សាទុកនៅក្នុង Pointer គ្មានសុវត្ថិភាពទេ។ ដូចនេះ ការប្រើ Pointer មិនធានាសុវត្ថិភាពឡើយ។

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class Data{
    private:
        int a;
    public:
        int b;
        Data(){
```

```
        a = 20;
        b = 40;
    }
};

void main(){
    clrscr();
    data obj;
    int *p;
    p = &obj.b;
    p--;
    cout << "\nA = "<<*p;
    *p+=4;
    cout << "\nA = "<<*p;
    cout<< endl;
    cout << "\nA = "<<obj.getA();
    getch();
}
```

The End !!!