

C Programming Language

Tutor : Mr Phonpaserth SISAYKEO :]

Basic Code

```
/*Written by: Mr Phonepaserth SISAYKEO.
```

Comments

```
Date: 29/04/2019 */
```

```
#include<stdio.h>
```

Pcessor, Directive, Header file

```
int main () {
```

Function main()

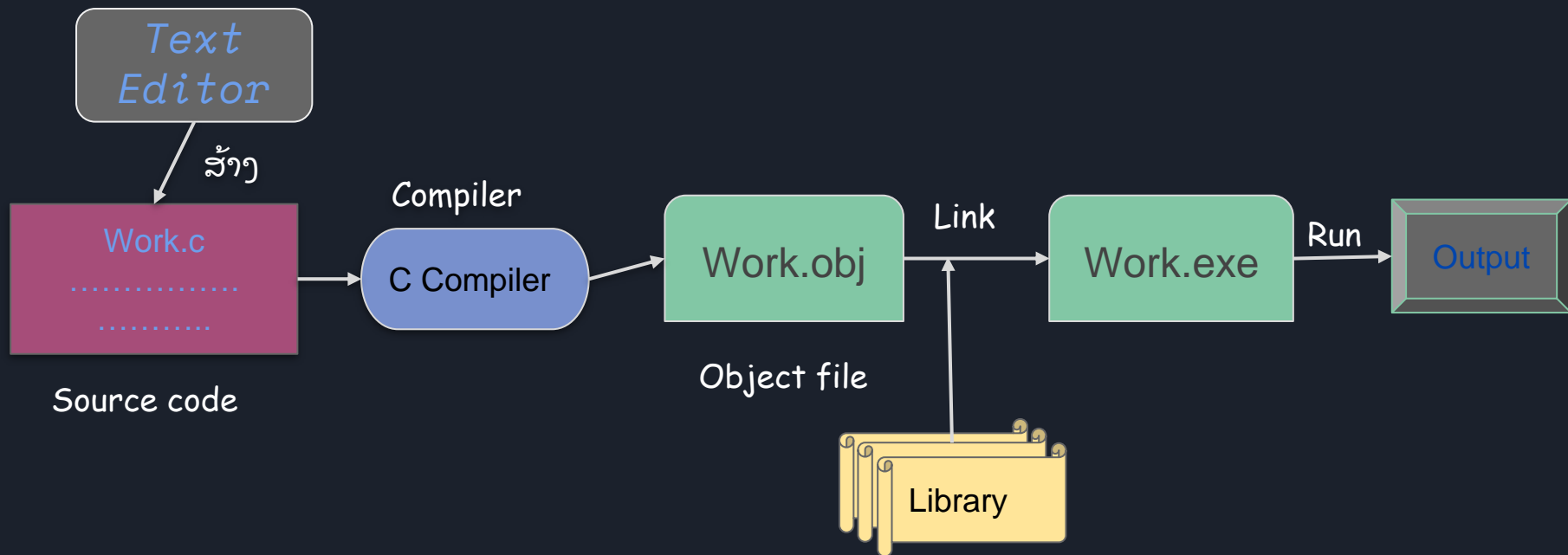
```
printf ("Hello my friends :) ");
```

Statements

```
return 0;
```

```
}
```

Don't
forget



Variable Vs Keyword(reserved word)

Variable

variable :ແມ່ນຊື່ທີ່ກຳນົດຂຶ້ນມາເພື່ອເກັບຂໍ້

Declaration: dataType

variableName:

Ex: **int** good;

Int better,best;

Declaration with value:

dataType variableName = value;

Ex: **int** go= 1;

Int ed= 6,omg= 2;

keyword

keyword:ແມ່ນຄຳສັບສະຫງວນ ທີ່ທຸກໂຕຕ້ອງ
ເປັນ lowercase ຫຼື ຕົວໃຈມນ້ອຍ.

auto, double, int struct, break, else, long,
switch, case, enum, register, typedef, char,
extern, return, union, const, float, short,
unsigned, continue, for, signed, void, default,
goto, sizeof, volatile, do, if, static, while.

ກົດເກນຕາມຫຼັກໄວຍາກອນຂອງພາສາໃນການຕັ້ງຊື່ມີດັ່ງນີ້:

- ຊື່ຕ້ອງຂຶ້ນຕົ້ນດ້ວຍຕົວອັກສອນ ຫຼື ສັນຍາລັກຂີດກອງ
underscore (_)
- ຊື່ບໍ່ສາມາດຂຶ້ນຕົ້ນດ້ວຍຕົວເລກ ສ່ວນຕົວຕໍ່ໄປສາມາດເປັນຕົວ
ເລກໄດ້
- ຊື່ສາມາດໃຊ້ມີໄດ້ຕົວອັກສອນ (A-Z, a-z, 0-9 ແລະ _)
- ຫ້າມໃຊ້ສັນຍາລັກພິເສດເຊັ່ນ: +, -, *, / ແລະ ອື່ນໆ
- ຊື່ເປັນແບບ case sensitive ເຊັ່ນ: a ແລະ A ຖືເປັນສອງຊື່ຕາງກັນ

Warm-Up

```
#include<stdio.h>

int main(){

printf("Name: Phonepaserth\n");

printf("Age: 18");

getch();

return 0 ;

}
```



Name: Phonepaserth
Age: 18

ລອງຂຽນເບິ່ງ
ຂຽນຕາມໃຈ (ຫ້າມຄຶກັນ)
ຂຽນມາ ຄົນລະ 3 ຢ່າງ
ໂດຍໃຊ້ ຄໍາສັ່ງ : printf("");

Data type

Fundamental Types

Integer

Floating

Character

Boolean

Derived Types

Array

Pointer

Structures

Enums

VS

Integer

ຈຳນວນຖ້ວນ ຫຼື ຈຳນວນເຕັມ ເຊິ່ງມີການຈອງຄວາມຈຳເທົ່າ 4 Byte

```
int name = value;
```

Example: ໃນ Memory.

```
int one;
```

```
int one = 1;
```

```
int two;
```

```
int one = 1, two = 2;
```

```
int three;
```

```
#include<stdio.h>
```

```
main() {
```

```
int number1, number2;
```

```
number 1 = 50000;
```

```
number 2 = -40000;
```

```
printf("Value of number1 = %d \n", number1);
```

```
printf("Value of number2 = %d \n", number2);
```

```
getch();
```

```
}
```

Floating Point Types.

ປະເພດຂອງ ຈຳນວນທົດສະນິຍົມ ຫຼື ຈຳນວນທີ່ມີເລກເສດ.

Float : ຂໍ້ມູນເລກເສດ ຈອງເນື້ອທີໄວ້ 4 Byte.
single-precision floating-point.

Double : ຂໍ້ມູນທີ່ມີເລກເສດຈອງເນື້ອທີໄວ້ 8 Byte.
double-precision floating-point.

Long Double : ຂໍ້ມູນທີ່ມີເລກເສດຈອງເນື້ອທີໄວ້ 10
Byte.
extended-precision floating-point.

Floating Point

ຈຳນວນທົດສະນິຍົມ ຫຼື ຈຳນວນທີ່ມີເລກເສດ.

float : ຂໍ້ມູນເລກເສດ ຈອງເນື້ອທີໄວ້ 4 Byte.
single-precision floating-point.

float name = value;

```
#include<stdio.h>
int main() {
```

```
    float height = 15.5;
    float base = 3.4;
    float area;
```

```
    area = 0.5*base*height;
```

```
    printf("Height: %f \n", height);
    printf("Base: %f \n", base);
    printf("Triangle area: %.2f \n", area); // %.f kum nod
```

0.000

```
    getch();
    return 0;
```

```
}
```

example:

```
float m = 1.2;
```

```
float i = 2.3;
```

```
float k = 3.4;
```

Or

```
float fiko = 0.9f;
```

Floating Point

ຈຳນວນທົດສະນິຍົມ ຫຼື ຈຳນວນທີ່ມີເລກເສດ.

double : ຂໍ້ມູນທີ່ມີເລກເສດຈອງເນື້ອທີໄວ້ 8 Byte.

double-precision floating-point.

```
#include<stdio.h>
```

```
main() {
```

```
double height = 5010.5;
```

```
double base = 500.25;
```

```
double area;
```

```
area = 0.5*height*base;
```

```
printf("This is height = %Lf \n", height);
```

```
printf("This is base = %Lf \n", base);
```

```
printf("Triangle area is = %Lf", area);
```

```
getch();
```

```
}
```

```
double name = value;
```

example:

```
double good-morning = 1.001;
```

```
double are = 111.001;
```

```
double Mr-Handsome = 0.111101;
```

Character

ຕົວອັບສອນ, ສາມາດເກັບຄ່າໄດ້ພຽງແຕ່ 1 ຕົວເທົ່ານັ້ນ ແລະຈອງເນື້ອທີໄວ້

1 BYTE

char name = value;

```
#include<stdio.h>
int main() {
```

```
char chTest;
chTest = 'A';
```

```
printf("Size of char = %d \n", sizeof(char));
printf("char A = %c \n", chTest);
printf("ASCII A = %d", chTest);
```

```
getch();
return 0;
}
```

example:

```
char ko ;
```

```
ko = 'o';
```

```
char f = 'i', k = 'o';
```

ຕ້ອງປະກາດຂະໜາດຂອງ
Array ໄວລາງໜ້າ

```
#include<stdio.h>
```

```
main() {
```

```
char name[15]="Phonepaserth";
```

```
int age=18;
```

```
printf("My name is %s \n", name);
```

```
printf("My age is %d \n", age);
```

```
printf("First character %c", name[0]);
```

```
getch();
```

```
}
```

Keyword “ (Constants) “

constant ແມ່ນ **identifier** ທີ່ກຳນົດໃຫ້ຕົວປ່ຽນໃດໜຶ່ງບໍ່ໃຫ້ມີການປ່ຽນແປງຄ່າ
ພາຍຫລັງ.

```
const DataType VariableName = value;
```

example:

```
const double sabaidee = 123.321;
```

```
const int hukder = 3000.01;
```

```
const char kk = 'p';
```

```
#include<stdio.h>
```

```
#include<math.h> //ໃຊ້ກັບ pow(x, y) ເທົ່າກັບ  
x^y
```

```
main() {
```

```
const double PI = 3.14159;
```

```
double radius = 50; double area;
```

```
area = PI* pow(radius, 2);
```

```
printf("Value of PI = %f \n", PI);
```

```
printf("Value of radius = %f \n", radius);
```

```
printf("Circle area is = %f", area);
```

```
getch();
```

```
}
```

Keyword “ (#define) “

ເປັນການການົດຄຳຄົງຄຳ ໂດຍການປະກາດໃຊ້ງານໄວໃນສ່ວນຂອງເຮດເຕີໄຟລ໌ (Header File) ໂດຍມີຮບແບບການປະກາດໃຊ້ງານຄຳຄົງຄຳດັ່ງນີ້:

`#define ConstantsName value ;`

ConstantsName ຄື: ຊື່ຂອງຄຳຄົງຄຳ

value ຄື: ຄ່າທີ່ຕ້ອງການການົດໃຫ້ຄຳຄົງຄຳ

example:

```
#define VAT 0.07
```

```
#define TEXT "Welcome to Laos"
```

```
#define NEWLINE '\n'
```

```
#define ONE 1
```

```
#include<stdio.h>
```

```
#include<math.h> //ໃຊ້ກັບ pow(x, y) ເທົ່າກັບ  
x^y
```

```
#define PI = 3.14159;
```

```
main() {
```

```
double radius = 50; double area;  
area = PI* pow(radius, 2);
```

```
printf("Value of PI = %f \n", PI);  
printf("Value of radius = %f \n", radius);  
printf("Circle area is = %f", area);
```

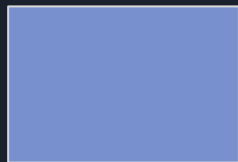
```
getch();
```

```
}
```

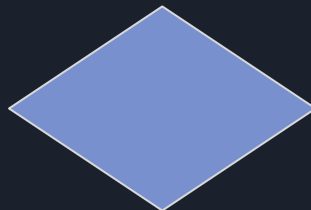
Flowchart



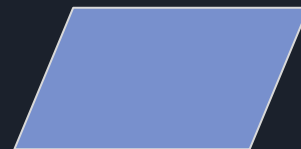
Start / Finish



Processing



Condition



Input / Output

1.ວິເຄາະບັນຫາ > 2.ອອກແບບ > 3.ລົງມືຂຽນ > 4. ທົດລອງ > 5. ເຮັດຄູ່ມື

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue parallelogram is in the foreground, and the green one is slightly behind it. They are positioned to the left of a blue horizontal bar.

Input-Output, Specifier, Operator

Output: ການສະແດງຜົນ

printf()

printf() ແມ່ນ function ພື້ນຖານໃນການສະແດງຜົນ
ໝາຍເຫດ: ການໃຊ້ printf() ໃນພາສາ C ແມ່ນສາມາດໃຊ້ໄດ້ແຕ່ຕົວອັກສອນ ພາສາ ອັງກິດ.


ປະເພດ

ບໍ່ມີຕົວປ່ຽນ

```
printf("message");
```

ມີຕົວປ່ຽນ

```
printf("%d", variableName);
```

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

main()
printf("I am so handsome.");
getch();
}
```

```
#include <stdio.h>
#include <conio.h>
main()
char n[] = "I am so handsome";
printf("%s",n);
getch();
}
```

Format Specifier

ແບບທາງການ

Specifier ແມ່ນຕົວທີ່ບົ່ງບອກເຖິງຂະໜາດ ແລະ ຊະນິດຂອງຄວາມຈໍາທີ່ຈັດໄວ້ສໍາລັບຕົວປ່ຽນ.

ແບບບ້ານງ

Specifier ແມ່ນຕົວທີ່ລະບຸຕໍາແໜ່ງ ແລະ ຊະນິດຂອງຂໍ້ມູນທີ່ສະແດງອອກ ແລະ ຮັບເຂົ້າ.

```
#include <stdio.h>
```

```
main(){  
    char a[] = "I am so handsome";  
    printf("%s",a);  
    getch();  
}
```

```
#include <stdio.h>
```

```
main(){  
    double wrong = 44.44;  
    printf("wrong : %s",wrong);  
    getch();  
}
```

```
#include <stdio.h>
```

```
main(){  
    int a = 13;  
    printf("I like the number %d",a);  
    getch();  
}
```



Specifier Table

Data type (ຊະນິດຂໍ້ມູນ)	Format	Memory size (ຂະໜາດຄວາມຈໍາ)	Range (ຂອບເຂດ)
unsigned char	%c	1 Byte	0 ຫາ 255
char	%c	1 Byte	-128 ຫາ 127
int	%d	2 Bytes	-32768 ຫາ 32767
unsigned int	%u	2 Bytes	0 ຫາ 65535
long int	%ld	4 Bytes	-2147483648 ຫາ 2147483648
unsigned long int	%lu	4 Bytes	0 ຫາ 4294967295
float	%f	4 Bytes	$3.4 \times (10^{-38})$ ຫາ $3.4 \times (10^{38})$
double	%lf	8 Bytes	$1.7 \times (10^{-308})$ ຫາ $1.7 \times (10^{308})$
long double	%Lf	10 Bytes	$3.4 \times (10^{-4932})$ ຫາ $1.1 \times (10^{4932})$
char[] (string)	%s		

Format Specifier

Data Type	Specifier	Usage
Integer	%d	ໃຊ້ກັບຈຳນວນຖ້ວນ
float, double	%f	ໃຊ້ກັບຈຳນວນທົດສະນິຍົມ
character	%c	ໃຊ້ກັບຕົວອັກສອນ
String	%s	ໃຊ້ກັບຂໍ້ຄວາມ

ສຳຫລັບ %f: ການບອກຈຳນວນຫລັງຈຸດທົດສະນິຍົມ

%nf : n ຈຳນວນຫລັງຈຸດທົດສະນິຍົມ

Escape Character

Code	ຜົນໄດ້ຮັບ
\n	ລົງແຖວ
\t	ຍະຫວ່າງ tab 1 ຫຼື ຍະຫວ່າງ 7 ເທື່ອ
\r	ກຳນົດ cursqr ໄປຢູ່ຕົ້ນແຖວ ຫຼື ຖ້າຜິມຕໍ່ຈາກ \r ມັນຈະເອົາຂໍ້ມູນໄປທັບໂຕທາງໜ້າ
\b	ລຶບຕົວອັກສອນສຸດທ້າຍອອກ 1 ຕົວ
\a	ເຮັດໃຫ້ມີສຽງດັງຂຶ້ນ
\0	Null string ຫຼືວ່າຄຳເປົ້າຫວ່າງ
\\	ສະແດງເຄື່ອງໝາຍ Backslash “ \ ”
\'	ສະແດງເຄື່ອງໝາຍ ‘
\”	ສະແດງເຄື່ອງໝາຍ “

Input

scanf()

scanf() ແມ່ນ function ທີ່ໃຊ້ເພື່ອຮັບຄ່າຂ້າມາທາງ Keyboard

ໝາຍເຫດ: ການໃຊ້ scanf() ໃນພາສາ C ແມ່ນສາມາດໃຊ້ໄດ້ແຕ່ຕົວອັກສອນ ພາສາ ອັງກິດ.

scanf("format",&variableName);

```
#include <stdio.h>
```

```
main(){  
    int test ;  
    printf("input your number: ");  
    scanf("%d",&test);  
    getch();  
}
```

ໝາຍເຫດ: ສຳລັບ String ບໍ່ຈຳເປັນຕ້ອງມີ & ໄວ້ຫນ້າ
ໂຕປ່ຽນ

Operator : ຕົວດຳເນີນການ

Operator ການຄຳນວນທາງ
ຄະນິດສາດ

Operator
ທາງຕັກກະສາດ

Operator
ການປຸງບທຽບ

Operator ການຄຳນວນທາງ
ຄະນິດສາດ

ເຄື່ອງໝາຍ	ໜ້າທີ່	ຕົວຢ່າງ	ອະທິບາຍ
+	ບວກ	$x=a+b$	x ເທົ່າກັບຜົນບວກຄ່າຂອງ a ບວກ b
-	ລົບ	$x=a-b$	x ເທົ່າກັບຜົນລົບຄ່າຂອງ a ລົບ b
*	ຄູນ	$x=a*b$	x ເທົ່າກັບຜົນຄູນຄ່າຂອງ a ຄູນ b
/	ຫານ	$x=a/b$	x ເທົ່າກັບຜົນຫານຄ່າຂອງ a ຫານ b
%	ຫານເອົາຕົວເສດ (Modulo)	$x=a\%b$	x ເທົ່າກັບຕົວເສດຈາກການຫານຄ່າຂອງ a ຫານ b
++	ເພີ່ມຂຶ້ນເທື່ອລະ 1	$x=++a$	ບວກຄ່າຂອງ a ຂຶ້ນ 1 ຄ່າແລ້ວຈຶ່ງເກັບໄວ້ x
		$x=a++$	ເກັບຄ່າ a ໄວ້ໃນ x ກ່ອນແລ້ວຈຶ່ງບວກຂຶ້ນ
--	ລົບລົງເທື່ອລະ 1	$x=--a$	ລົບຄ່າຂອງ a ລົງ 1 ຄ່າແລ້ວຈຶ່ງເກັບໄວ້ x
		$x=a--$	ເກັບຄ່າ a ໄວ້ໃນ x ກ່ອນແລ້ວຈຶ່ງລົບລົງ
+=	ບວກ (ແບບລັດ)	$a+=b \Rightarrow a=a+b$	a ເທົ່າກັບ a ບວກ b

Operator ທາງຕັກກະສາດ

ເຄື່ອງໝາຍ	ຊື່
&&	ແລະ
	ຫຼື
!	ປະຕິເສດ

Operator ການປຽບທຽບ

ເຄື່ອງໝາຍ	ຊື່	ຕົວຢ່າງ
==	ເທົ່າກັບ	$x == y$
!=	ບໍ່ເທົ່າກັບ	$x != y$
>	ໃຫຍ່ກວ່າ	$x > y$
>=	ໃຫຍ່ກວ່າເທົ່າກັບ	$x >= y$
<	ນ້ອຍກວ່າ	$x < y$
<=	ນ້ອຍກວ່າເທົ່າກັບ	$x <= y$

Decision Control Statement ປະໂຫຍກຄວບຄຸມການຕັດສິນໃຈ

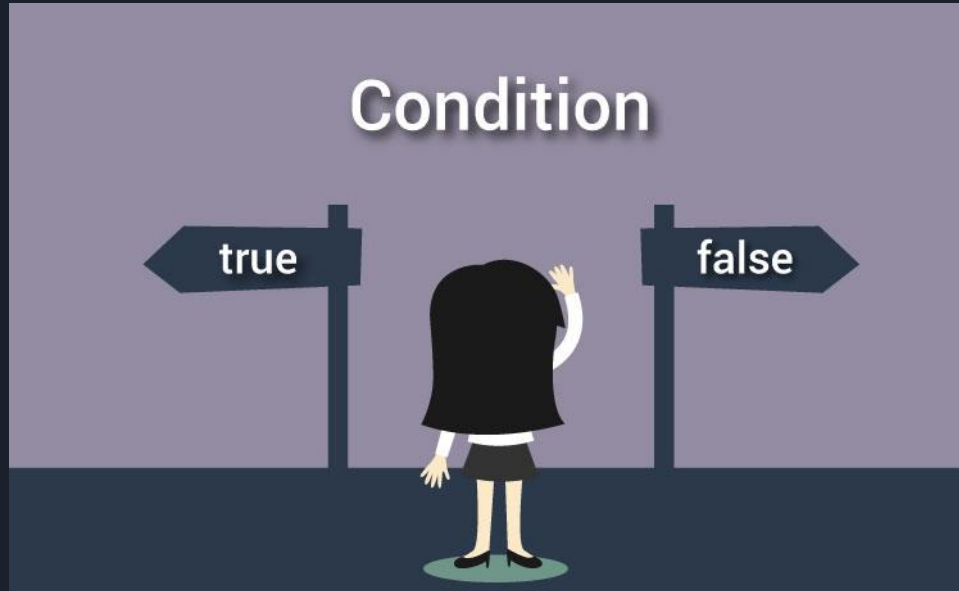
Decision Control Statement

Conditional
Statement
ປະໂຫຍກເງື່ອນໄຂ

Loop
ການວົນຊ້າ

Conditional Statement

1. ການຂຽນ Conditional Statement ແມ່ນຫຍັງ?
ແມ່ນການຂຽນເງື່ອນໄຂເພື່ອໃຫ້ໂປຣແກຼມສາມາດກວດ
ສອບວ່າ ຄວນຈະເຮັດວຽກຕາມເງື່ອນໄຂໃດຈຶ່ງຈະຖືກຕ້ອງ.





Decision Control Statement



```
graph TD; A[Decision Control Statement] --> B[if]; A --> C["if - else"]; A --> D["Nested if"]; A --> E[switch];
```

if

if - else

Nested
if

switch

Conditional Statement

if : ຖ້າ

ຄໍາສັ່ງ if ແມ່ນຄໍາສັ່ງໃຊ້ເພື່ອກວດສອບຕາມເງື່ອນໄຂທີ່ເຮົາ
ກຳນົດໄວ້,

- ເຮັດວຽກເມື່ອ true
- ບໍ່ເຮັດວຽກເມື່ອ false

```
if (Condition)
{

    // statements

}
```

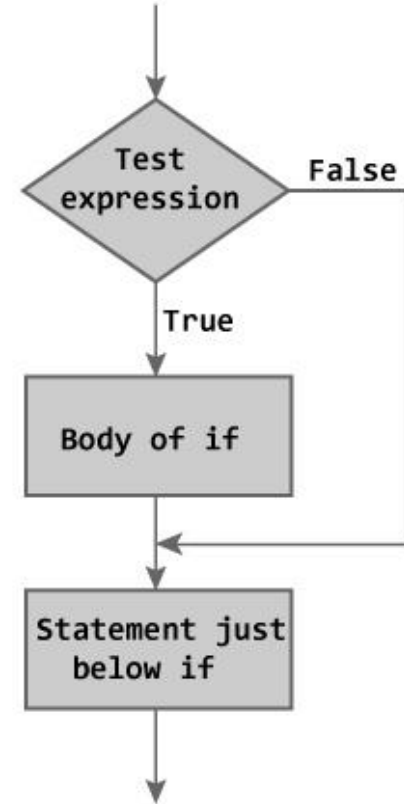


Figure: Flowchart of if Statement



```
#include<stdio.h>
#include<math.h>
```

```
void main()
{
    int A, B;
    A = 2;
    B = 3;

    if(A + B == 5){
        printf("This work!!!");
    }
}
```

Conditional Statement

If - else

ຄໍາສັ່ງ if-else ເປັນຄໍາສັ່ງໃນການເຮັດວຽກບົນພື້ນຖານຄວາມເປັນຈິງ:

- ເມື່ອເງື່ອນໄຂຖືກຕາມ if ກໍຈະເຮັດວຽກໃນ if
- ເມື່ອບໍ່ຖືກຕາມເງື່ອນໄຂ if ກໍຈະເຮັດວຽກໃນ else

```
if (Condition) {  
    // body of if  
}  
else {  
    // the body of else  
}
```

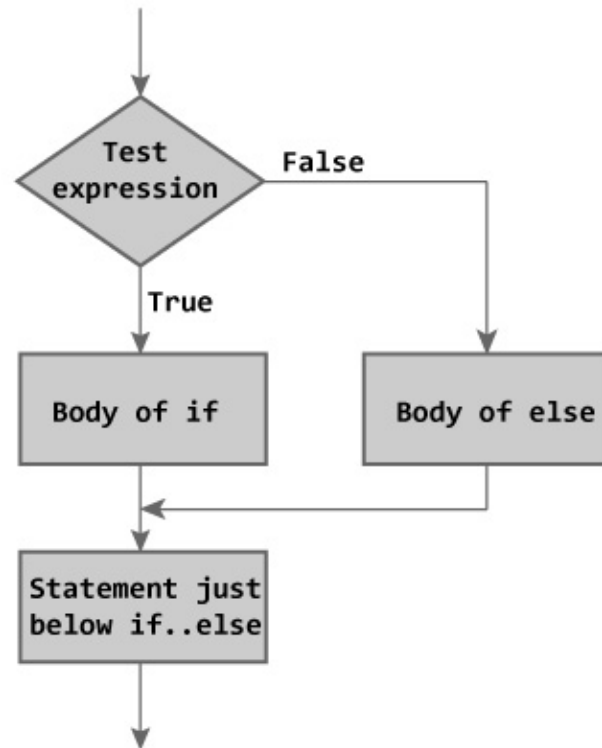



Figure: Flowchart of if...else Statement



```
#include<stdio.h>
void main(){
    int A = 5;
    if(A == 4){
        printf("I am the Dude");
    }
    else {
        printf("I am number 5\n");
        printf("You Know!!!");
    }
}
```

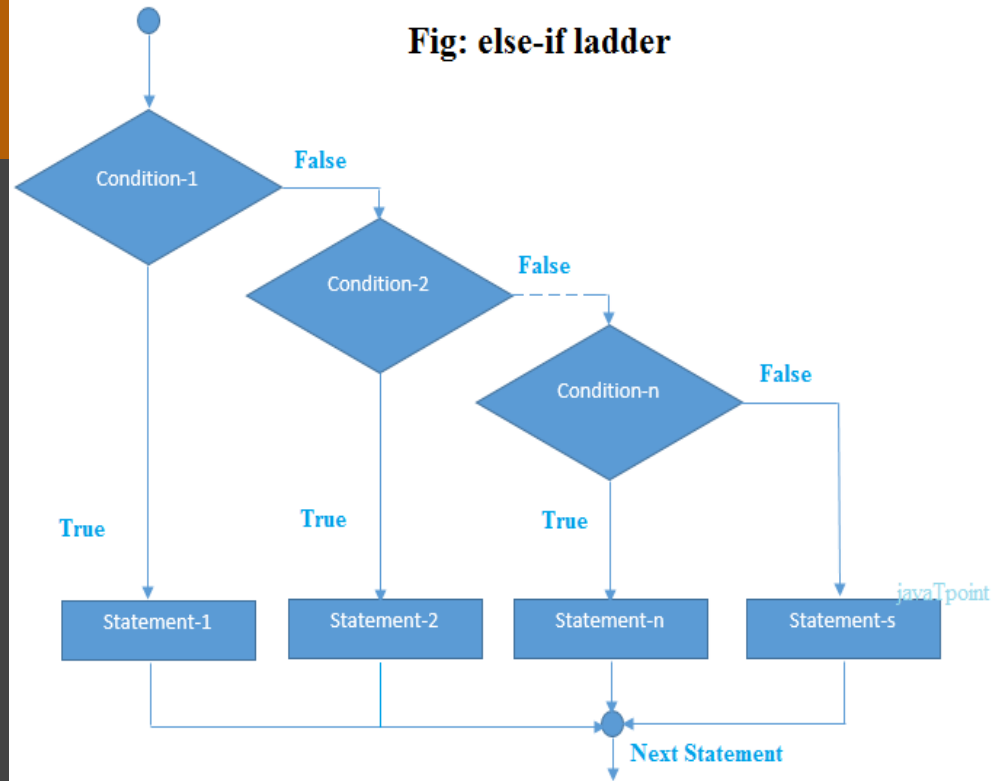
Conditional Statement

Nested if

- ແມ່ນຄຳສັ່ງທີ່ມີ if, else if ແລະ else
ເມື່ອເງື່ອນໄຂຖືກຕາມ if ກໍຈະເຮັດວຽກໃນ if
- ເມື່ອບໍ່ຖືກຕາມເງື່ອນໄຂ if ກໍຈະເຮັດວຽກໃນ else if ຕໍ່ໄປ
ເລື້ອຍໆ
ຕົວຢ່າງ ການໃຊ້ ກໍຈະເຮັດວຽກໃນ else

```
if (testExpression1){  
    //  
}  
else if(testExpression2){  
    //  
}  
else if (testExpression 3){  
    //  
}  
.  
.  
else{  
    //  
}
```

Fig: else-if ladder



```
#include<stdio.h>
```

```
main()  
{
```

```
    int grade = 100;
```

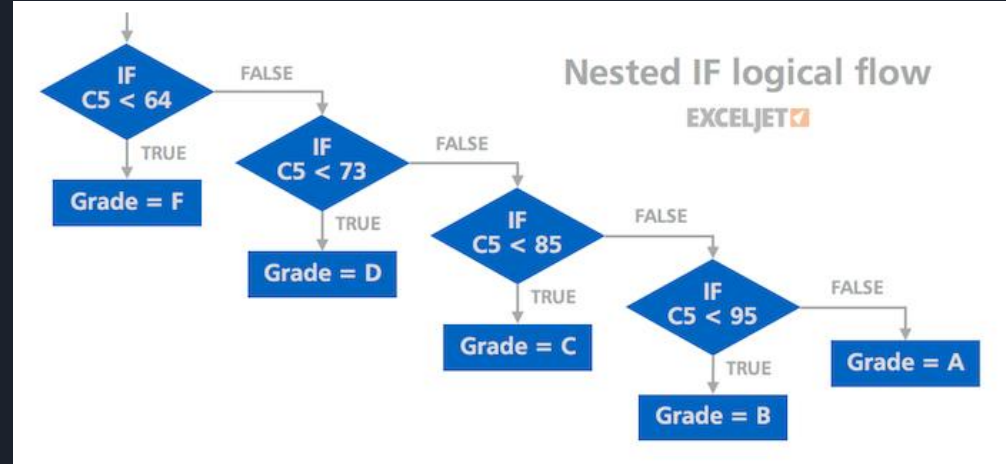
```
    if(grade < 75){  
        printf("Your grade is C");  
    }
```

```
    else if(grade < 85){  
        printf("Your grade is B");  
    }
```

```
    else if(grade < 90){  
        printf("Your grade is A");  
    }
```

```
    else{  
        printf("Your grade is A++");  
    }
```

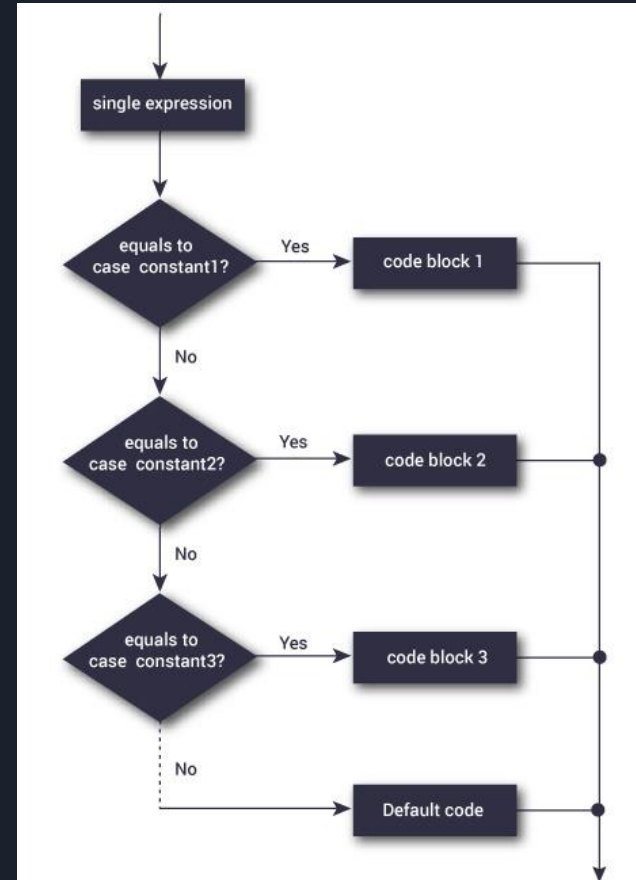
```
}
```



Conditional Statement

switch case

- ຄໍາສັ່ງ switch ຫຼື switch-case ແມ່ນຄ້າຍຄືກັນກັບຄໍາສັ່ງ if-else if-else. ແຕ່ການເຮັດວຽກຂອງ switch-case ຈະເຮັດວຽກໄດ້ສະດວກກວ່າເພາະວ່າມັນເປັນຄໍາສັ່ງຄ້າຍຄືກັບລາຍການ (Menu), ມີຫຼາຍໆ case, ແລ້ວໂປຣແກຣມກະຈະເຮັດວຽກຕາມ case ທີ່ຖືກຕ້ອງ



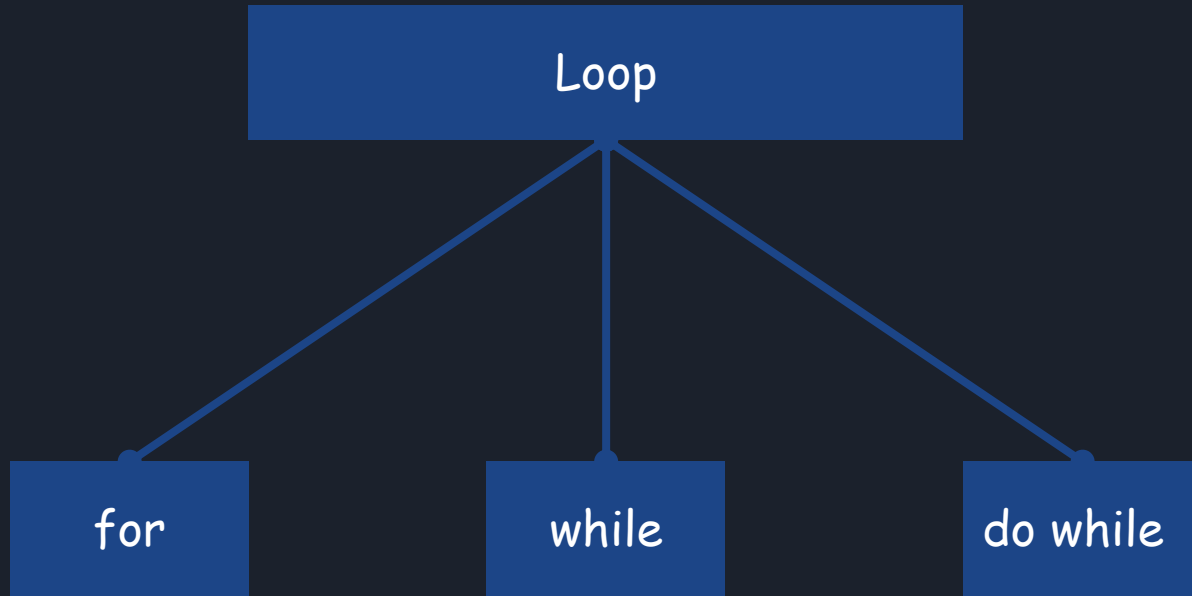
```
switch ( variableName ){  
  
    case value1:  
        statement;  
        break;  
  
    case value2:  
        statement;  
        break;  
  
    case value3:  
        statement;  
        break;  
    .  
    .  
default: statement;  
}
```

```
#include <stdio.h>  
#include <conio.h>  
main()  
{  
  
    int A;  
    printf("i = ");  
    scanf("%d",&A);  
    switch ( A )  
    {  
        case 1 :  
            printf("Yes");  
            break;  
        case 2 :  
            printf("No");  
            break;  
        default :  
            printf("Error");  
    }  
    getch();  
}
```



Loop

ແມ່ນຄໍາສັ່ງເຮັດວົນໄປວົນມາ ຫຼື ເຮັດຊໍ້າ
ເພື່ອໃຊ້ຄວບຄຸມໂປຣແກມ



Loop : ການວົນລຳ




while

ຄໍາສັ່ງ while ການເຮັດວຽກວົນລຳໂດຍທີ່ມີການເຮັດວຽກຕາມເງື່ອນໄຂທີ່ເປັນ True ຈົນກວ່າຖືກເງື່ອນໄຂທີ່ False ແລ້ວໂປຣແກມຈະອອກຈາກ while Loop

```
while(Condition)
{

    // statements

}
```

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

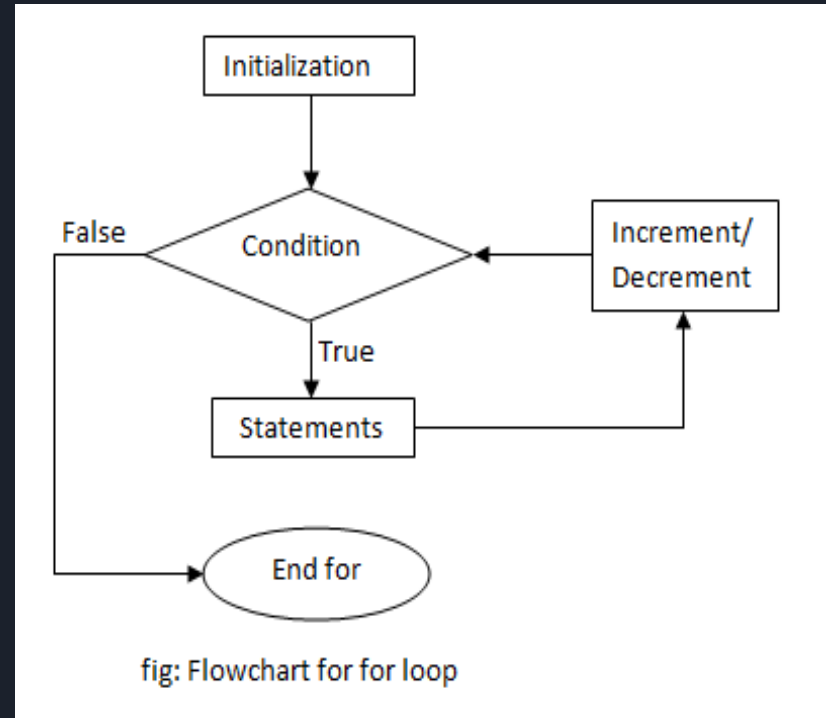
void main(){
    int i;
    i=1;
    while(i<10){
        printf("\n%d",i);
        i++;
    }
    getch();
}
```


Loop : ການວົນລຳ

"for" loop

ຄໍາສັ່ງ for ແມ່ນການເຮັດວຽກວົນລຳໂດຍທີ່ເຮົາມີການກຳນົດເງື່ອນໄຂໃຫ້ມັນເຮັດວຽກວ່າຈະມີການວົນລຳຮອດໃສ່ ສິນສຸດຢູ່ໃສ່. ເຊິ່ງສໍາລັບ Loop for ນີ້ແມ່ນການເຮັດວຽກຂຶ້ນກັບຄ່າເລີ່ມຕົ້ນທີ່ເຮົາວາງໄວ້, ເງື່ອນໄຂ ແລະ ການເພີ່ມ ຫຼື ຫຼຸດຄ່າ.

```
for(initializer;condition;updater)
{
    // statements
}
```





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

main() {
int i;
    for(i=0;i<20;i++){

        printf("\n%d",i++);
    }
getch();
}
```

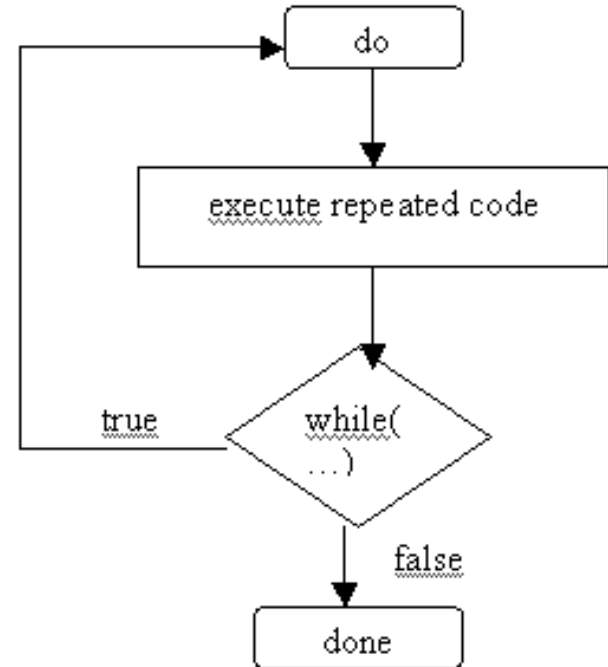
Loop : ການວົນລຳ

" do while " loop

ຄ້າຍຄືກັບ Loop while ແຕ່ວ່າການເຮັດວຽກຂອງມັນຄືຈະເຮັດໃນສ່ວນ Loop ກ່ອນຈາກນັ້ນຈຶ່ງມາກວດສອບເງື່ອນໄຂຕາມຫຼັງວ່າຖືກຕ້ອງ ຫຼື ບໍ່. ຖ້າວ່າເງື່ອນໄຂເປັນຈິງກໍຈະກັບມາເຮັດວຽກໃນສ່ວນ Loop ຄືນແຕ່ຖ້າບໍ່ກໍຈະອອກຈາກ Loop ໄປເຮັດຄໍາສັ່ງອັນຕໍ່ໄປ

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

Flow Diagram of do .. while LOOP





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

```
main(){
    int i;
    i=1;
    Do
    {
        printf("\n%d",i);
        i++;
    } while(i<=10);
    getch();
}
```