



## LAYOUT OF C PROGRAM

ඕනෑම පරිගණක භාෂාවක් ගත් විට, එය උපයෝගී කරගෙන වැඩසටහනක් ලිවීම සඳහා එයටම ආවේනික උන විශේෂ වියුහයක් පවතී. අප වැඩසටහන ලිවිය යුතු වන්නේ එම වියුහයට අනුකූලවයි. එම වියුහය භාෂාවෙන් භාෂාවට වෙනස් වේ. අපි දැන් C භාෂාව සඳහා වන මෙම වියුහය මොන වගේද කියල පොඩ්ඩක් බලමු.

HEADER	#include <stdio.h>
MAIN( )	int main ( ) {
BODY	printf(" hello world!" );
RETURN	return 0; }

අපි දැන් බලමු මොනාද මේ HEADER, MAIN( ), BODY හා RETURN කියන්නේ කියල.

## HEADER

C වැඩසටහනක් ලිවීමේදී පළමු හා ජර්මානාත්මක කාර්යය වන්නේ සි වැඩසටහනකට ශීර්ෂ ලිපිගොනු ඇතුළත් කිරීමයි.

ශීර්ෂ ගොනුව යනු .h දිගුව සහිත ගොනුවකි, එහි C හි ශීර්ෂ සහ ජර්නව ගොනු කිහිපයක් අතර බෙදා ගත යුතු සාර්ව අර්ථ දැක්වීම් මෙහි අඩංගු වේ.

ශීර්ෂ ගොනු වලට උදාහරණ,

▣ stddef.h

▣ stdint.h

▣ stdio.h

▣ stdlib.h

▣ string.h

▣ math.h

header ගොනුව ඇතුළත් කිරීමේ syntax එක,

```
#include<"header ගොනුවේ නම.h">
```

EX:

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

```
#include<string.h>
```

```
#include<stddef.h>
```

#include<stdlib.h>

## main()

C වැඩසටහනක ඊළඟ කාර්යය වන්නේ main( ) ශ්‍රේණිය ජරකාශ කිරීමයි.

main() ශ්‍රේණිය ඇතුළත් කිරීමේ syntax එක,

```
int main()  
{  
  
  
}
```

## BODY

C වැඩසටහනක ක්‍රියාත්මක වන කේත බණ්ඩ ලිවිය යුතු වන්නේ **body** කොටස තුළය.

එය { }(සහල වරහන්) ඇතුළත ලිවිය යුතු වේ.

## RETURN

ඕනෑම C වැඩසටහනක අවසන් කොටස වන්නේ **return** ශ්‍රේණියයි.

**return** කියල කියන්නේ යම් ශ්‍රේණියකින් යම් අගයක් ලබා ගැනීම, කියල සරලව කියන්න පුළුවන් .තව ඉස්සරහට යද්දී **return** කියන්නේ මොකක්ද එකෙන් වෙන්වේ මොකක්ද කියල තේරුම් ගන්න පුළුවන්.

✓ ඔන්න අපි C වැඩසටහනක මූලික කොටස් ටික ඉගෙන ගත්ත.

- ✓ දැන් අපිට කියෙන්නේ මේ C භාෂාවේ එන මූලික දැනුම ලබා ගන්න එක තමයි.
- ✓ අපි එහෙන්න මුලින්ම C භාෂාවේ එන keywords හා identifiers හදුනා ගනිමු.

## Keywords

keywords කියලා කියන්නේ,පෙර අර්ථ දක්වන(pre defined) ලද හා වෙන්කරන ලද (reserved) වචන සමූහයකි.මේවා syntax වල කොටසක් වන අතර මෙම වචන identifier එකක් ලෙස භාවිත කල නොහැක.

C හි භාවිත වන keyword ටිකක් පහතින් පෙන්වන්නේ.

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

## Identifiers

identifier එකක් කියලා කියන්නේ අප විසින් විචල්‍යකට (variable), ශ්‍රේණියකට (function), ව්‍යුහයකට (structure) ලබාදෙන නම වේ. මම කලින් කිව්ව විදියටම identifier එකක නමක් විදියට keyword එකක් යෙදීම කාරක රීති දෝෂයක් (syntax error) වේ.

identifier එකක් ලබාදීමේ syntax එක,

```
int emp_no;  
float number;  
char letter;
```

- I. වලංගු හඳුනාගැනීමක් සඳහා අක්ෂර , ඉලක්කම් හා යටි ඉරි තිබිය හැක.
- II. identifier එකේ මුල් අකුර, අක්ෂරයක් හෝ යටි ඉරක් විය හැක. ඉලක්කමකින් identifier එකක් ආරම්භ කළ නොහැක.
- III. identifier එකක් ලෙස keyword භාවිතා කළ නොහැක.
- IV. වචන කිහිපයකින් යුතු identifier එකක් නම්, එම වචන අතර space එකක් තිබිය නොහැක.
- V. යොදාගන්න identifier එක අකුරු 32 කට වඩා වැඩි වූ විට ගැටළු මතුවිය හැක.

Valid identifiers	Invalid identifiers
Count5=10	5count=10

<code>_campus=SLTC</code>	<code>Full name=0</code>
<code>full_name=0</code>	<code>lf =0</code>

- අනෙක් කාරණාව තමයි **identifier** එකකට නමක් ලබාදෙන සෑම අවස්ථාවකම එයට අර්ථවත් නමක් ලබාදෙන්න වග බලාගන්න.එක ඔයාලට **program** එක කරගෙන යද්දී ලොකු පහසුවක් වෙයි.
- උදාහරණයක් ලෙස , යමෙකුගේ මුල් නම ලබාගත යුතු අවස්ථාවක් සලකමු.නම රඳවා ගන්න **first\_name** යන **identifier** එක ලබා දෙන එක ,**fn** හරි වෙන මොකක් හරි කෙටි නමක් දෙන එකට වඩා හොඳයි කියලා මම හිතනවා.

## Variables

**Variables(විචල්‍යයක්)** යනු යම් දත්තයක් රඳවා තබා ගැනීමට යොදාගන්නා බහාලුමක්(ගබඩාවක්) වේ.සෑම විචල්‍යකටම තමාටම ආවේනික වූ නමක් (**identifier**) ලබාදිය යුතුය.විචල්‍යයක නම යනු මතක ස්ථානයක සංකේතාත්මක නිරූපනයකි.

Ex:

```
int price=100;
```

මෙතන **price** කියලා කියන්නේ **int** වර්ගයට අයත් විචල්‍යයක්,ඒ වගේම ඒ විචල්‍යයට **100** යන පූර්ණ අගය ලබා දීල තියෙනවා.

අනෙක් වැදගත්ම කාරණය, විචල්යයකට ලබා දුන් අගයන් වෙනස් කළ හැකි වීමයි.මෙයට විචල්යය යන නම ලැබී ඇත්තේද මේ ගුණය නිසාවෙනි.

```
int price=100;  
  
price=50;
```

විචල්යයන් භාවිත කිරීමේදී සැලකිලිමත් වියයුතු තවත් කාරණයක් වන්නේ,යම් විටෙක ඔබ විචල්ය වර්ගය ලබාදුන් විට එය නැවත වෙනස් කළ නොහැකි වීමයි.

```
int price=100; //integer Variable  
  
price=50.5; //error  
  
double price; //error
```

## Constant

විචල්යයක අගය නොවෙනස්ව පවත්වා ගතයුතු අවස්ථාවන් වලදී අපට **const** යන keyword එක භාවිත කළ හැකි වේ.

Ex

```
const float pi=3.14;
```

මෙහි pi යනු වෙනස්කළ නොහැකි නියතයකි.

Edited by RWD.J.Source:google.com1678cpp

