



FUNCTIONS.

ඕනෑම පරිගණක භාෂාවක් භාවිතා කිරීමේදී ඉතාමත්ම වැදගත් දෙයක් නමයි මේ functions කියලා කියන්නේ. ජර්මාන වශයෙන් භාවිතා කරන function වර්ග දෙකක් තියෙනවා.

01-user-defined(පරිශීලක අර්ථ දැක්වන ලද) function

02-standard(සම්මත) library function

What is a functions?

යම් නිශ්චිත කාර්යයක් ඉටුකර ගැනීමේ අරමුණින් නිර්මාණය කරන ලද කේත සමූහයක් function එකක් ලෙස අර්ථ දැක්විය හැකි වේ.

උදාහරණයක් විදියට මෙහෙම හිතන්න, අපිට ඕනි වෙනවා යම් වස්තුවක වර්ගඵලය සොයන්න අපිට මේ වැඩේ කරගන්න වෙනම function එක නිර්මාණය කරගන්න පුළුවන්. එතකොට අපිට ඕනි වෙන හැම වෙලාවකම මේ function එක call කරලා අපිට ඕනි වැඩේ කරගන්න පුළුවන්.

Why We Use functions?

01-අපගේ ක්‍රමලේඛය කියවීම හා යාවත්කාලීන කිරීම පහසු කරයි.

02-එකම function එක නැවත නැවත භාවිතා කිරීමට හැකි වීම නිසා කාලය ඉතිරිකර ගැනීමට හැකිවීම.

03-ක්‍රමලේඛනයේ ඇති දෝෂ නිරාකරණය කිරීම පහසු කරවයි.

04-එකම කේතඛණ්ඩය පුනර්කරණය වීමක් සිදු නොවන නිසා ක්‍රමලේඛනයේ ධාරිතාවය අඩු වේ.

Standard (සම්මත) library function.

ක්රමලේඛන භාෂාව තුළම අර්ථ දක්වා ඇති functions මෙන්ම හඳුන්වයි. මෙම ශ්‍රේණියන් ශීර්ෂ ලිපි ගොනුව(header file) තුළ අර්ථ දක්වා තිබේ.

EX=අප බොහෝ සෙයින් භාවිතයට ගන්නා `printf()` ශ්‍රේණිය standard library function බාණ්ඩයට අයත් වේ. මෙම ශ්‍රේණිය `stdio.h` කියන header file එක තුළ අර්ථ දක්වා ඇත. තවද සංඛ්‍යාත්මක වර්ගමූලය සෙවීමට භාවිතා කරන `sqrt()` ශ්‍රේණියද `math.h` header file එක තුළ අර්ථ දක්වා ඇති standard library function වලට උදාහරණයකි.

Library Functions: are the functions which are declared in the C header files such as `scanf()`, `printf()`, `gets()`, `puts()`, `ceil()`, `floor()` etc.

User-defined(පරිශීලක අර්ථ දක්වන ලද) function.

පරිශීලකයාගේ අවශ්‍යතාවය අනුව , යම් නිශ්චිත කාර්යක් ඉටුකරගැනීමට සකස්කරගන්නා වූ ශ්‍රේණියක් පරිශීලක අර්ථ දක්වන ලද ශ්‍රේණියක් ලෙස අර්ථ දක්වනු ලැබේ.

HOW TO CREATE USER-DEFINED FUNCTION:

```
return_type function_name(argument){
//set of code to execute;
}
```

- return type: යම්කිසි function එකකින් return කරන අගයක දත්ත ජරාජය(data type) return type කියල සරලව කයන්න පුළුවන්. int, bouble , char, void, short වැනි ඕනෑම දත්ත ජරාජයක්(data type) return type එකක් වියහැකි වේ.

- **function name:**අපි අර්ථ දක්වන ලද ශීර්ෂයේ සත්ය නම මෙනමින් හඳුන්වනු ලැබේ.ශීර්ෂයකට නමක් ලබාදීමේදී එම ශීර්ෂය මගින් කෙරෙන කාර්යය හොඳින් පැහැදිලි වෙන ලෙස ලබාදීම මගින් කෙතකරණය පහසු කරගැනීමට හැකි වේ.
- **arguments:**මෙය සාමාන්‍යයෙන් භාවිතයට ගන්න විචල්‍යයන් වලට බොහෝ සෙයින් සමාන වේ.function එකක් තුල argument එකක් හෝ වැඩි ගණනක් පැවතිය හැකි වේ.මේවා ශීර්ෂයකට ලබාදෙන ආදානයන් යැයිද සරලව අර්ථ දැක්විය හැකිවේ .

HOW TO WORK USER-DEFIND FUNCTIONS.

How function works in C programming?

```
#include <stdio.h>

void functionName()
{
    ... ..
    ... ..
}

int main()
{
    ... ..
    ... ..
    functionName();
}

... ..
... ..
}
```

EXAMPLE 1-(C38.c)

- පරිශීලක ලබාදෙන සංඛ්‍යා දෙකක එකතුව ලබා ගැනීම සඳහා ශ්‍රේතයක් නිර්මාණය කර ගනිමු.

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

```
int total(int a ,int b); // function prototype//
```

```
int main()
```

```
{
```

```
    int a,b,ans;
```

```
    printf("Enter Number1=");
```

```
    scanf("%d",&a);
```

```
    printf("Enter Number2=");
```

```
    scanf("%d",&b);
```

```
    ans=total(a,b); //call function//
```

```
    printf("Total is=%d",ans);
```

```
    return 0;
```

```
}
```

```
int total(int a1,int b2) //create function//
```

```
{
```

```
    int sum;
```

```

        sum=a1+b2;

        return sum;

    }
}

```

Output of above code:

```

"C:\Users\Jayashanka Deshan\Desktop\C39.exe"
Enter Number1=30
Enter Number2=20
Total is=50
Process returned 0 (0x0)   execution time : 9.255 s
Press any key to continue.

```

Explain Above Code:

මෙක ඉතා සරල උදාහරණයකි. මූලිකම අපි total කියල function එක නිර්මාණය කරගෙන තියෙනවා. function එක ඇතුලේ තමයි අපි සංඛ්‍යා දෙක එකතු කරන්න ඕනි කරන කේතය ලියල තියෙන්නේ. දැන් අපිට ඕනි වෙන වෙලාවක මේ ශ්‍රිතය කැඳවල ප්‍රතිදානයක් ලබාගන්න පුළුවන් වෙනවා.

ඊළගට අපි main function එක ඇතුලේ අපිට අවශ්‍ය වෙන අනිත් දෙවල් ටික සකස් කරගෙන තියෙනවා. ඊළගට තමයි අපි හදාගත්ත function එක call කරලා අපට අවශ්‍ය වෙන ක්‍රියාකාරකම් කරවගන්න යන්නේ (9 වන පේලිය). අපේ main function එක ඇතුලේ තියෙන 6 වන පේලිය බලන්න. එතැනදී අපි කරගෙන තියෙන්නේ ans කියන variable එකට අපි සකස්කරගත්ත function එකෙන් ලැබෙන අගය පවරා ගන්න එක. ඊළගට printf කියන සම්මත ශ්‍රිතය (standerd function) භාවිතයෙන් අප සකස්කර ගත්ත ශ්‍රිතයෙන් ලැබෙන අගය ප්‍රතිදානය කරනවා.

Argument and Functions.

argument එකක් කියන්නේ මොකක්ද කියල අපි කලින් කතා කරා.අපේ කලින් උදාහරණය බලන්න.එකේ a හා b කියන අගයන් තමයි function එකට pass කරලා තියෙන්නේ.

ඊළගට අපේ function එකේ තියෙන a1 හා b2 කියන parameters දෙක මගින් කලින් සඳහන් කරපු අගයන් දෙක ලබා ගන්නවා.පහත උදාහරණය බැලුවම ඔයාලට මේක හොඳින් තේරුම් ගන්න පුළුවන් වෙයි.

තව දෙයක් තියෙනවා.අපි function එකේ දීල තියෙන parameters වල දත්ත ප්‍රරූපයෙන්ම තමයි අපි function එක call කරනකොටත් අගයන් ලබාදෙන්න ඕනි.

උදාහරණයක් විදියට අපේ කලින් programme එක බලන්න .එකේ a1,b2, කියන parameters ලබාදීලා තියෙන්නේ int දත්ත ප්‍රරූපයෙන්.ඒවගේම a,b කියන අගයන්ද ලබාදීලා තියෙන්නේ int දත්ත ප්‍රරූපයෙන්මයි.

```
{
    int a,b,ans;

    printf("Enter Number1=");
    scanf("%d",&a);

    printf("Enter Number2=");
    scanf("%d",&b);

    ans=total(a,b); //call function//

    printf("Total is=%d",ans);

    return 0;
}
int total(int a1,int b2) //create function//
{
    int sum;
```

