Functions

What is function?

- Piece of code to accomplish certain operation.
- It has a name for identification
- They are of two types
  - Predefined functions
  - User defined functions

FUNCTION SYNTAX

```
#include<stdio.h>
    void fun();   //function prototype declaration
```

```
void main()
            {
              fun();      //function call
                          cout<<"i am main line!";
               }
```

```
void fun()  //function definition
{
cout<<"i m function body";
}
```

FUNCTION FOR ADDTION

```
#include<iostream.h>
#include<conio.h>
add(); //declaration

void main()
{
clrscr();
add(); //calling
getch();
}
```

```
add()  //definition
{
int a,b,c;
cout<<“enter two no. to add";
cin>>a>>b;
c=a+b;
cout<<c;
}
```

Benefits of function

- Modularization
- Easy to read
- Easy to debug
- Easy to modify
- Avoids rewriting of same code over and over
- Better memory utilization

FUNCTION FOR ADDTION

```
#include<stdio.h>
#include<conio.h>
add();
sub();
mul();
div();

void main()
{
clrscr();
add();
sub();
mul();
div();
getch();
}

add()
{
int a,b,c;
printf("Enter two no. for Addition");
scanf("%d%d",&a,&b);
c=a+b;
printf("Addition is : %d",c);
}
```

sub()

```
{
int a,b,c;
printf("Enter two no. for Substraction");
scanf("%d%d",&a,&b);
c=a-b;
printf("Substraction is : %d",c); }
```

mul()

```
{
int a,b,c;
printf("Enter two no for Multiplication");
scanf("%d%d",&a,&b);
c=a*b;
printf("Multiplication is : %d",c); }
```

div()

```
{
int a,b,c;
printf("Enter Two no for Division");
scanf("%d%d",&a,&b);
c=a/b;
printf("division is : %d",c);
}
```

Remember

- Program must have at least one function
- Function names must be unique
- Function is an operation, once defined can be used many times.
- One function in the program must be main
- Function consumes memory only when it is invoked and released from RAM as soon as it finishes its job.

FUNCTION CALL

Passing value between functions

- We can communicate between the calling and the called functions by passing the parameters.

There are two ways to pass parameters to a function:

1. **CALL BY VALUE:**
2. **CALL BY REFERENCE:**

Passing value between functions

CALL BY VALUE:

- **You actually pass a copy of the variable to the function modifies the copy the original remains unaltered.**

CALL BY REFERENCE:

- **Call by reference mechanism is used when you want functions to do the changes in called parameters and reflect those changes back to the calling function .**
- **In this case addresses (pointer) of the variable are passed to function so that function can work directly over the addresses.**

CALL BY VALUE

Example - Call by Value

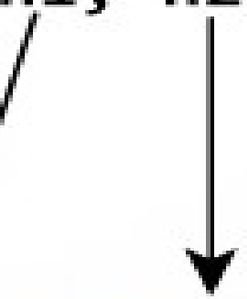
```
#include<iostream.h>
#include<conio.h>
int square(int x);
void main() //main function
{
int a,b;
clrscr();
cout<<"enter any value:\n"<<endl;
cin>>a;
b=square(a); //calling by value
cout<<"square of "<<a<<" = "<<b<<endl;
getch();
}
```

```
int square(int x)
{
int y;
y=x*x;
return(y);
}
```

PASSING ARGUMENTS TO FUNCTION

How to Pass arguments to a function ?

```c
#include <stdio.h>

int addNumbers(int a, int b);

int main()
{
    ... .. ...

    sum = addNumbers(n1, n2);

    ... .. ...
}

int addNumbers(int a, int b)
{
    ... .. ...
    ... .. ...
}
```



Return statement of a function

```c
#include <stdio.h>

int addNumbers(int a, int b);

int main()
{
    ... .. ...

    sum = addNumbers(n1, n2);

    ... .. ...
}

int addNumbers(int a, int b)
{
    ... .. ...
    return result;
}
```

Passing Arguments to a user defined function

```
#include<iostream.h>
#include<conio.h>
int addNumber(int a,int b); //function Declaration

void main()
{
clrscr();
int n1,n2,sum;
cout<<"Enter two number:"<<endl;
cin>>n1>>n2;
sum=addNumber(n1,n2); //function call
cout<<"sum="<<sum;
getch(); }
```

Return statement of a function

```
int addNumber(int a,int b) //function Definition
{
int result;
result=a+b;
return result;
}
```

CALL BY REFERENCE

Example - Call by Reference

```
#include<iostream.h>
#include<conio.h>
void swap(int*p1,int*p2); //function prototype
void main() //main function
{
int a=10;
int b=20;
cout<<"\n before value of a = "<<a <<" and value of b = "<<b<<endl;
swap(&a,&b); //calling by reference
cout<<"\n value of a(p1) = " <<a <<" and value of b(p2) = "<<b;
getch();
}
```

Example - Call by Reference

```
void swap(int*p1,int*p2)  //function / method
{
int c;
c=*p1;
*p1=*p2;
*p2=c;
cout<<"after swaping value of a(p1) = "<<*p1<<" and value of b(p2) =
    "<<*p2<<"\n";
}
```

GETC() & PUTC() FUNCTION

Getc( ) & putc( ) function

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
clrscr();
char c;
cout<<"Enter character:"<<endl;
c=getc(stdin);
cout<<"character entered:";
putc (c, stdout);
getch();
}
```

Getc( ) & putc( ) function

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
int main(void)
{
clrscr();
char msg[]="Hello";
int i=0;
while(msg[i])
putc(msg[i++],stdout);
getch();
}
```

THANKS