DSA DAY 10

classmate
Date
Page

Function

It has a well defined task.

Structure of Function :-

return type   Function Name   Paeameters passing

```
int  addnumber (int a, int b){ // Function
                                  decleeation
        eetuen a+b; // Function body
}

int main (){
        int a,b;                      Attributes
        int ans = addnumbee (a,b); //
}                                   Function
                                    Calling
```

* IF Function does not eeturn anything, then void can be used.

Note:- Theee is possibility that theee are no input parameteed.

```
int main (){
           ↑ theee are no i/p parameteee.
}
```

```
int main(){        Note:-
                      Retuen 0 means that the
    eeturn 0;         main function has
                      been succesfully executed
}
```

• Retuen '0' means it's gives it to opeeating system.

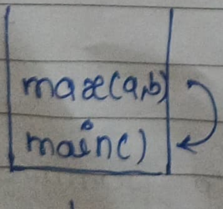
* Function Call Stack

∴ It work's like stack Datastructure & Follow LIFO ordering.

Function call stack show one after another which Function calls are happens. & which Function called another Function, local variables of Function & what the Function will return.

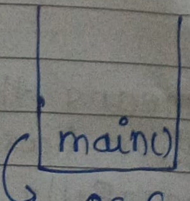
Ex. Finding maximum ~~of~~ From two Numbers

```
main() {                      int max( int a, int b) {
   int a=1;                       if (a>b)
   int b=2;                          return a;
   cout << max(a,b);              else
   return 0;                         return b;
}                              }
```



max Function return 2 as max to main() & gets out of Function call stack

↓



main Function return 0 to os & gets out of Function call stack

OS (return 0 to OS)

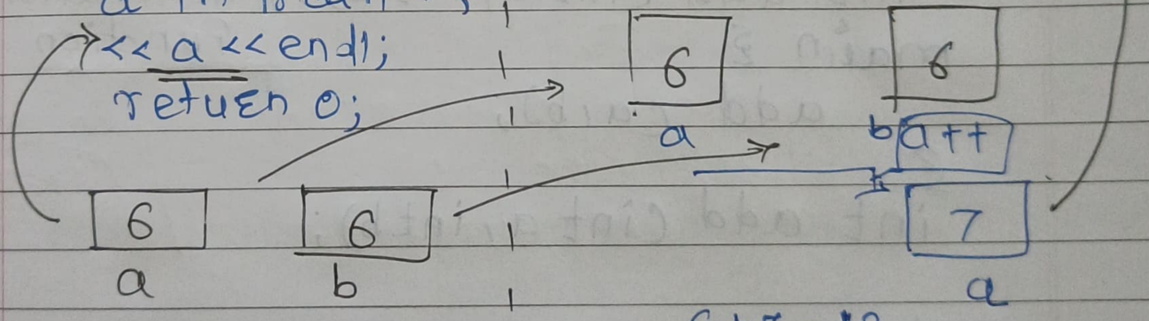
↓

Note:-
main() returns 0 to opeeating system as OS only staets the programe.

pass by value:-
Sepreate Copy will be created.

Ex:- Add a two numbees a & b.

```
main() {                      | int add(int a, int b) {
    int a=6;                  |       a++;
    int b=6;                  |   cout<<"value of a
                              |          in function"<<a;
    cout<<add(a,b);           |     return a+b;
    cout<<"value of           |   }
    a in main";               |
    <<a<<endl;                |
    retuEn 0;                 |
```



6+7=13

Value of sum :- 13

value of a befoee calling Function in main function is :- 6

In add Function a is :- 7

* How to See address of Variables?
```
int a=5;
cout << &a;  Print address of a
```
↑ These operator is an address operator.

* Invoking the Function
```
main() {
    add (a,b);      ← These give an error.
}
int add (int a, int b);
```
Soln to this
```
int add (int a, int b); // declaration of Function
main {
    add (a,b);
}
int add (int a, int b);
```
∴ Note

We have to make sure that atleast we have declared the Function before invoking the Function.

Q.2) Find max of 3 numbers

```
int maxOfThree( int a, int b, intc) {

    if (a>b && a>c){
        return a;
    } else if ( b>a && b>c) {
        return b;
    } else{
        return c;
    }
}.
```

Q.3) Counting From 1 to n

```
void Counting (int n)

    For( int i=1; i<=n; i++)
        cout << i <<endl;

}
```

Q.4) function to Students & grades.

```
~~char~~ printGrade (int marks){
Char
    if (marks >= 90) { retuen 'A' }
    else if (marks >= 80) { return 'B' }
    else if (marks >= 70) { return 'C' }
    else {
            return 'D'
    };
```

→ This Qn solve using an switch statment to more neatly.

```
swith (marks/10) {

    case 10 : return 'A'; break;
    case 9  : return 'A'; break;
    case 8  : return 'B'; break;
    case 7  : return 'C'; break;
    case 6  : return 'D'; break;
    default : return 'E';

}
```

Q Sum OF Even numbers Upto n.

```
int printEvenSum ( int n) {

    int sum = 0;
    For (int i = 2; i <= n; i++){     // or directl
        iF ( i % 2 == 0)                 i+=2
            sum = sum + i;
    }
    return sum;
}
```

Note:- Avoid % operator as it is heavy operation.