Module-3
Functions

Why Functions

Programs studied/written in the UG Level are very small compared to when we write complex progs in real time applications. If the prog is very big following disadvantages can be there.

- It is very difficult for programmer to write Large Programs.
- Difficult to identify logical error's & debug them.
- Difficult to read, understand & analyse.
- Large progs are more prone to error's & so on.

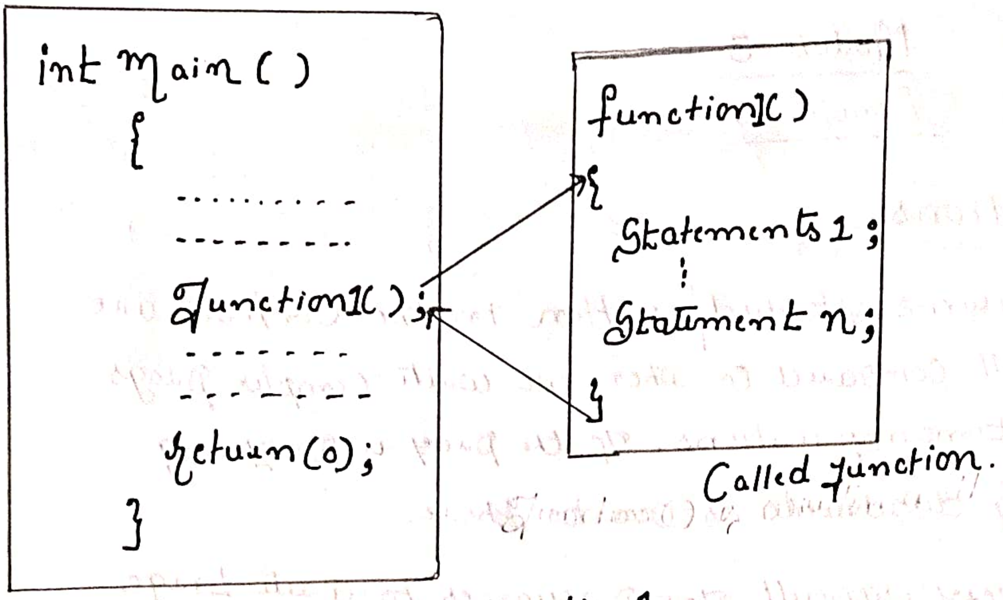
The above disadvantages can ~~be~~ overcome using Functions.

Function Definition:

'C' enables programmer to break Large Program into smaller segments. These small segments are known as functions.

Functions more or less they are independent to each other. Every function in a 'C' program performs well defined task.

In short
Functions are.
- Smaller segments
- Independent to each other.
- Performs well defined task
- There are two types
  - Library/predefined functions
  - user defined functions.



```
int main ( )
  {
      . . . . . . . . .
      . . . . . . . . .
      function1( );
      . . . . . . .
      . . . . . . .
    return(0);
  }
```

```
function1( )
{
  Statements 1;
     :
  Statement n;
}
```

Called function.

Main() calls function1

Main function → is known as Calling function → {A fn that uses another function

function1 → is known as Called function.

Now let us understand what main() does & is responsible for. Let us take an analogy of organisation.

Organisation.	main( ).
① In any organisation, owner is boss.	① In 'C' program, main() is the boss.
② Departments in the organisation are under the control of boss	② All functions in the progs are executed under the control of main().
③ Boss is responsible for conduction of smooth day-to-day activities.	③ Main() is responsible for smooth execution by calling appropriate functions.
④ Departments will do other activities themself with help of sub-ordinates.	④ Functions perform assigned activities themself with help of other sub functions if any.
⑥ once all the activities are performed by departments orgn job of the boss is done	⑤ All fn called by main() performed their activities, job of main() is completed.

Prog to demonstrate using Library function.

```
#include<stdio.h>
#include<math.h>
Void mainC)
{
    float n, s;
    n=36;
    s = sqrt(n);
    Printf ("Square root of n = %f", s);

}
```

<u>o/p</u>

```
s= 6.000000
square root of ~~36~~n
= 6.000000 .
```

Advantages of functions.

- → Reusability of Code.
- → Reduction in size of Code.
- → Easier debugging
- → Readability of the program can be Increased
- → function sharing.

Ref Pg 268 & 269 For "why fns are Needed".

Advantages of using Library fn

1) Since functions are readily available programmer's job is made easier.

2) usage of Library function reduces development time.

Ex 2: Prog to Check Even or odd using f^n

```
#include<stdio.h>
Void CheckoddEven(); //fn prototype.
Int main().
{
    CheckoddEven(); //Function call.
    return(0);
}

void checkOddEven().
{
   Int num;
   Printf("Enter the number");
   Scanf("%d", &num);
   if(num%2==0)
   {
     Printf("%d is Even", num);
   }
   Else.
   {
     Printf("%d is odd", num);
   }
}.
```

```
while(num%2==0)
{
 Pf("%d is Even", num);
 break;
}
while(num%2!=0)
{
 pf("%d is odd", num);
 break;
}
```