PROGRAMMING IN C

UNIT 3

Functions

CONTENTS

- Introduction
- Functions
- Types of Functions
- **Advantages of user-defined function**
- **3 Steps of Using Functions in C**
- **Function Prototype**

FUNCTION

- A function is a group of statements that together perform a task. Every C program has at least one function, which is **main().**
- C functions are basic building blocks in a program
- You can divide up your code into separate functions.
- A function **declaration** tells the compiler about a function's name, return type, and parameters.
- A function **definition** provides the actual body of the function

TYPES OF FUNCTIONS IN C PROGRAMMING

- There are two types of functions in C programming:
- Standard library functions (inbuilt functions): printf(), getch(), strlen(), clrscr().
- User defined functions

Advantages of user-defined function

- The program will be easier to understand, maintain and debug.
- Reusable codes that can be used in other programs
- A large program can be divided into smaller modules. Hence, a large project can be divided among many programmers.

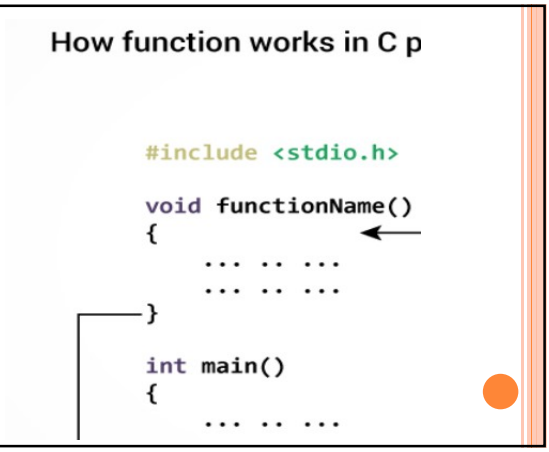
3 steps of using functions in C

- **Function declaration or prototype** – This informs compiler about the function name, function parameters and return value's data type.
- **Function call** – This calls the actual function
- **Function definition** – This contains all the statements to be executed.

Example

```
#include<stdio.h>
void square ( );
void main( )
{
printf ( "\n Program to find Square");
square ( ) ;
}
void square ( )
{
   int n, sq ;
   printf ( "\n Enter a Number\n");
   scanf ( "%d", &n ) ;
   sq=n*n;
   printf("square is =%d",sq);
}
```

How function works in C p



Function prototype

- A function prototype is the declaration of a function that specifies
  - function's name
  - Parameters (arguments)
  - return type.
- A function prototype gives information to the compiler.

Example: return_type function_name(parameter list);

```
int sum(int num1, int num2);
int square(int n);
float area(int r);
```

Function Arguments

- If a function is to use arguments, it must declare variables that accept the values of the arguments. These variables are called the **formal parameters**of the function.
- Formal parameters behave like other local variables inside the function and are created upon entry into the function and destroyed upon exit.
- While calling a function, there are two ways in which arguments can be passed to a function –

Sr.No.	Call Type & Description
1	Call by valueThis method copies the actual value of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument.
2	Call by referenceThis method copies the address of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call. This means that changes made to the parameter affect the argument.

References

1. Computer Fundamental and Programming in C, J B Dixit
2. LET US C, YASHWANT KANETKAR
3. Basics of C Programming, 2011, by J.B. Dixit
4. https://www.tutorialspoint.com/cprogramming/c_functions.htm
5. https://www.geeksforgeeks.org/functions-in-c/
6. https://fresh2refresh.com/c-programming/c-function/