SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 23ITT101 PROBLEM SOLVING AND C PROGRAMMING

I YEAR /I SEMESTER

Unit 4- Functions

Topic 2: Functions, Declaration, definition and user defined functions

Brain Storming

1. What is the purpose of the function?

Introduction

- A function is a group of statements that together perform a task.
- The function contains the set of programming statements enclosed by {}.
- The function is also known as *procedure* or *subroutine* in other programming languages.
- Every C program has at least one function, which is **main()**, and all the most trivial programs can define additional functions.

Advantage of functions in C

1. Reduce the source code

2. Easy to maintain and modify

3. It can be called any where in the program.

What is the purpose of a function prototype?

- Function prototype specifies the input/output interface to the function
- i.e. what to give to the function and what to expect from the function.

- It tells the return type of the data that the function will return.
- It tells the number of arguments passed to the function.
- It tells the data types of the each of the passed arguments.
- Also it tells the order in which the arguments are passed to the function.

Conti...

Return Type – A function may return a value. The **return_type** is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword **void**.

Function Name – This is the actual name of the function. The function name and the parameter list together constitute the function signature.

Conti...

Parameters:

- ✓ The parameter list refers to the type, order, and number of the parameters of a function.
- ✓ Parameters are optional; that is, a function may contain no parameters.
- ✓ Actual parameter – This is the argument which is used in function call.
- ✓ Formal parameter – This is the argument which is used in function definition.

- ✓ 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. The function body contains a collection of statements that define what the function does.
- ✓ Function call The function call statement invokes the function

```c
// C program to illustrate the function prototype
#include <stdio.h>

// Function prototype
float calculateRectangleArea(float length, float width);

void main()
{
    float length = 5.0;
    float width = 3.0;

    // Function call
    float area = calculateRectangleArea(length, width);

    printf("The area of the rectangle is: %.2f\n", area);
}

// Function definition
float calculateRectangleArea(float length, float width)
{
    return length * width;
}
```

Output:

The area of the rectangle is: 15.00

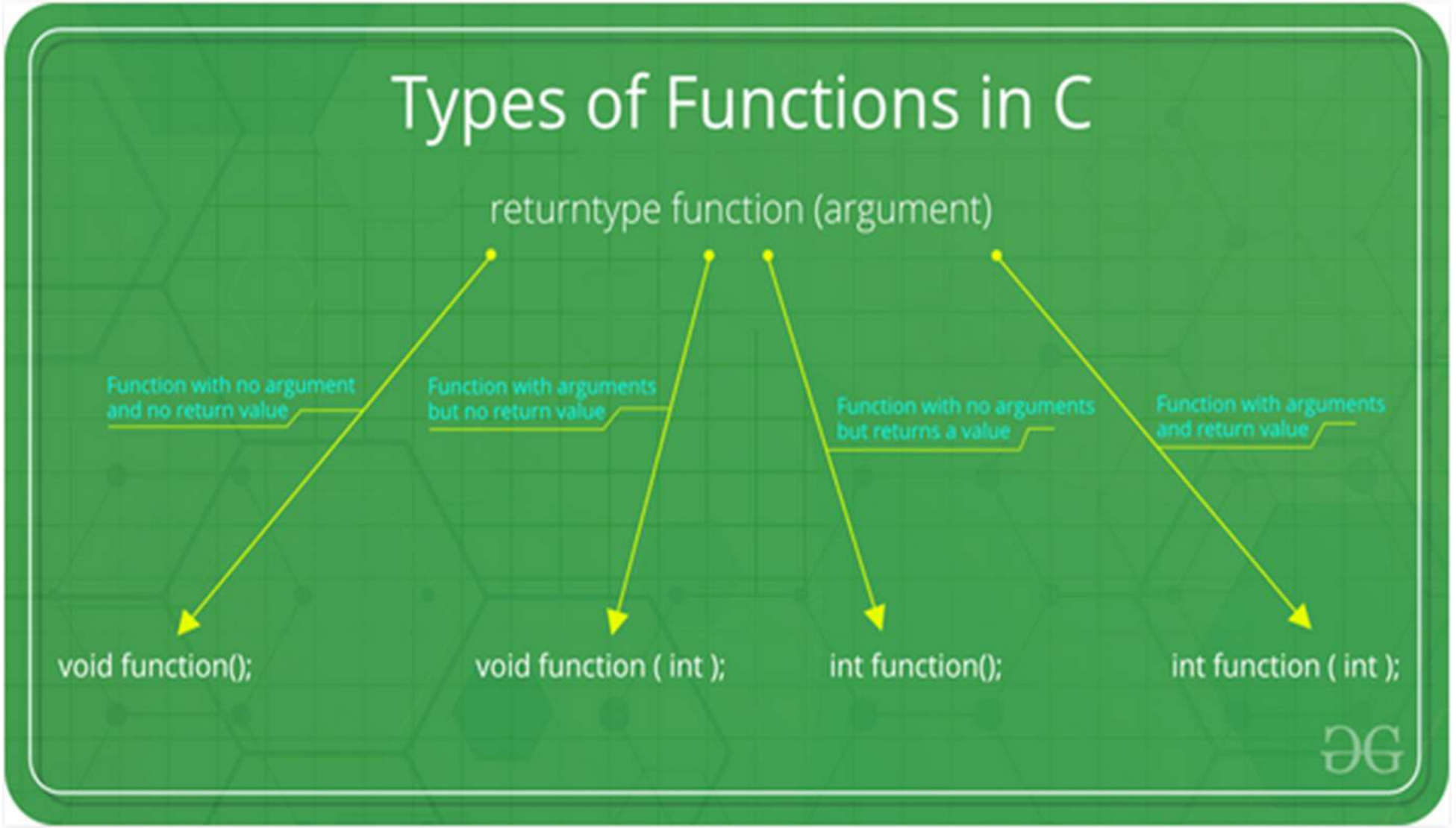
Function call

- The function call statement invokes the function. When a function is invoked the compiler jumps to the called function to execute the statements that are part of that function.
- There are four different aspects of function calls.

Types of function call

- function without arguments and without return value
- function without arguments and with return value
- function with arguments and without return value
- function with arguments and with return value

Conti...


Types of Functions in C
returntype function (argument)
Function with no argument and no return value
Function with arguments but no return value
Function with no arguments but returns a value
Function with arguments and return value
void function();
void function (int);
int function();
int function (int);

Example for Function with argument and with return value

```
#include<stdio.h>
int sum(int, int);
void main()
{
   int a,b,result;
   printf("\nGoing to calculate the sum of two numbers:");

   printf("\nEnter two numbers:");
   scanf("%d %d",&a,&b);
   result = sum(a,b);
   printf("\nThe sum is : %d",result);
}
int sum(int a, int b)
{
   return a+b;
}
```

Example:without argument and return value

```
#include<stdio.h>
void printName();
void main ()
{
   printf("Hello ");
   printName();
}
void printName()
{
   printf(“This is function call example");
}
```

Example for Function without argument and return value

```
#include <stdio.h>

void checkPrimeNumber();

void main() {
  checkPrimeNumber();    // argument is not passed
}
// return type is void meaning doesn't return any
value
void checkPrimeNumber() {
  int n, i, flag = 0;

  printf("Enter a positive integer: ");
  scanf("%d",&n);

  // 0 and 1 are not prime numbers
  if (n == 0 || n == 1)
    flag = 1;

  for(i = 2; i <= n/2; ++i) {
    if(n%i == 0) {
      flag = 1;
      break;
    }
  }

  if (flag == 1)
    printf("%d is not a prime number.", n);
  else
    printf("%d is a prime number.", n);
}
```

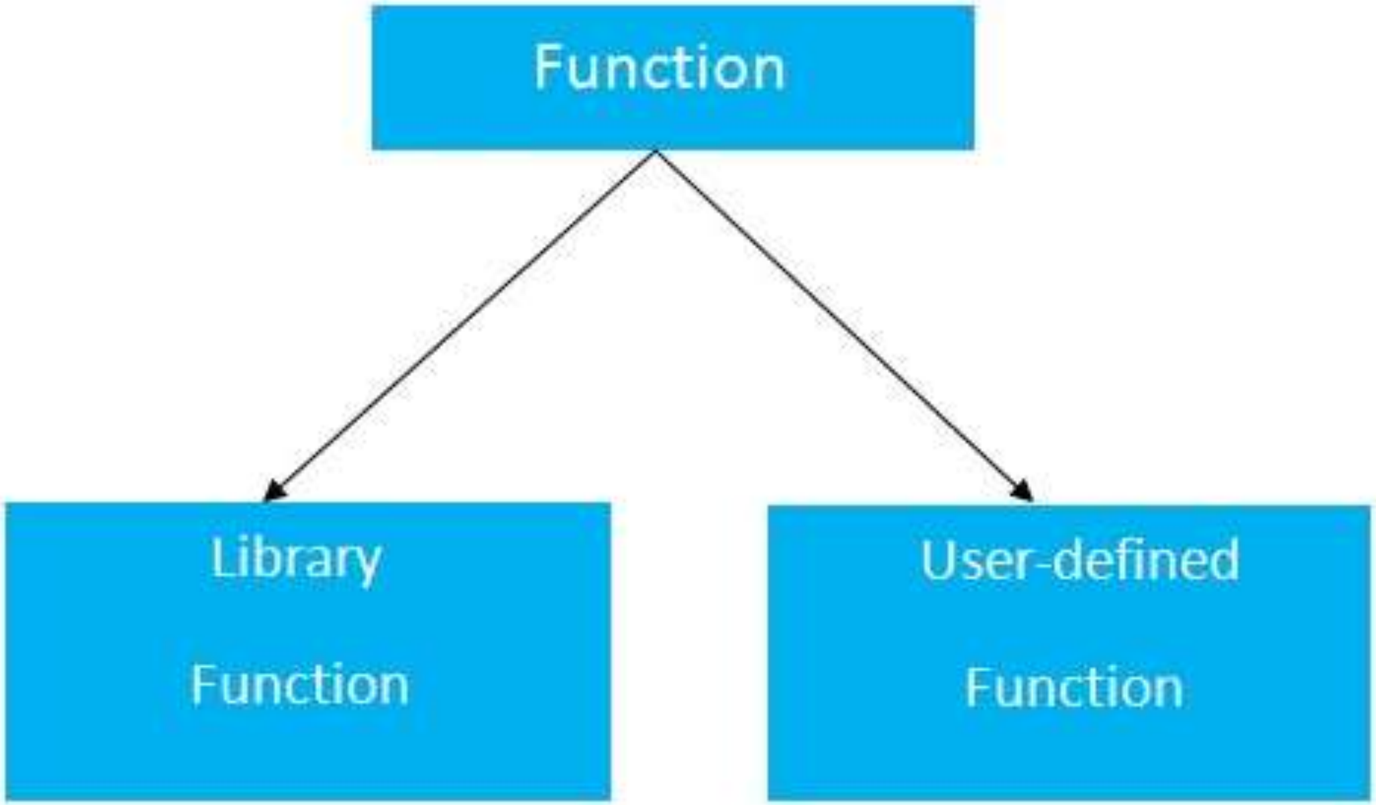
Example for Function without argument and with return value

```
#include<stdio.h>
int sum();
void main()
{
   int result;
   printf("\nGoing to calculate the sum of two numbers:");
   result = sum();
   printf("%d",result);
}
int sum()
{
   int a,b;
   printf("\nEnter two numbers");
   scanf("%d %d",&a,&b);
   return a+b;
}
```

Example for Function with argument and without return value

```
#include<stdio.h>
void sum(int, int);
void main()
{
   int a,b,result;
   printf("\nGoing to calculate the sum of two numbers:");
   printf("\nEnter two numbers:");
   scanf("%d %d",&a,&b);
   sum(a,b);
}
void sum(int a, int b)
{
   printf("\nThe sum is %d",a+b);
}
```

Types of Functions



Types of Functions

S.No.	User-Defined Functions	Library Functions
1	These functions are not predefined in the Compiler.	These functions are predefined in the compiler of C language.
2	These functions are created by users as per their own requirements.	These functions are not created by users as their own.
3	User-defined functions are not stored in library files.	Library Functions are stored in a special library file.
4	There is no such kind of requirement to add a particular library.	If the user wants to use a particular library function then the user has to add the particular library of that function in the header file of the program.
5	Execution of the program begins from the user-define function.	Execution of the program does not begin from the library function.
6	**Example:** sum(), fact(),…etc.	**Example:** printf(), scanf(), sqrt(),…etc.

Conti...

- There are two types of functions in C programming:
- **Library Functions**: are the functions which are declared in the C header files such as scanf(), printf(), gets(), puts(), ceil(), floor() etc.
- **User-defined functions**: are the functions which are created by the C programmer, so that he/she can use it many times. It reduces the complexity of a big program and optimizes the code.

```c
//User defined functions:
// Function declaration
void myFunction(int x, int y);

// The main method
int main() {
  int result = myFunction(5, 3); // call the function
  printf("Result is = %d", result);

}

// Function definition
int myFunction(int x, int y) {
  return x + y;
}
```

```c
//Built-in functions
#include <string.h>
void main () {
  char str1[12] = "Hello";
  char str2[12] = "World";
  char str3[12];
  int  len ;
        /* copy str1 into str3 */
  strcpy(str3, str1);            //  Function call
  printf("strcpy( str3, str1) :  %s\n", str3 ); //  Function call
        /* concatenates str1 and str2 */
  strcat( str1, str2); //  Function call
  printf("strcat( str1, str2):   %s\n", str1 );
        /* total lenghth of str1 after concatenation */
  len = strlen(str1);
  printf("strlen(str1) :  %d\n", len );
}
```

Assessment 1

1. What is function?

Ans : _____

2. Write about function prototype?

Ans : _____

References

1. Reema Thareja, “Programming in C”, Oxford University Press, Second Edition, 2016

Thank You