2nd YEAR COMPUTER

REVISION LECTURE NOTES

CHAPTER NO.13 (FUNCTIONS IN C)

PUNJAB BOARD

LECTURE NO.1 **TIME: 30 MIN.**

PREVIEW OF TODAY'S TOPICS

A. Functions (Overview), Importance of function, Structure & Un-structured Programming

Name: ________________________

Roll No: ________________________

Date: ________________________

FUNCTIONS (OVERVIEW), IMPORTANCE OF FUNCTION, STRUCRUE & UN-STRUCTURED PROGRAMMING

EXTENSIVE QUESTION

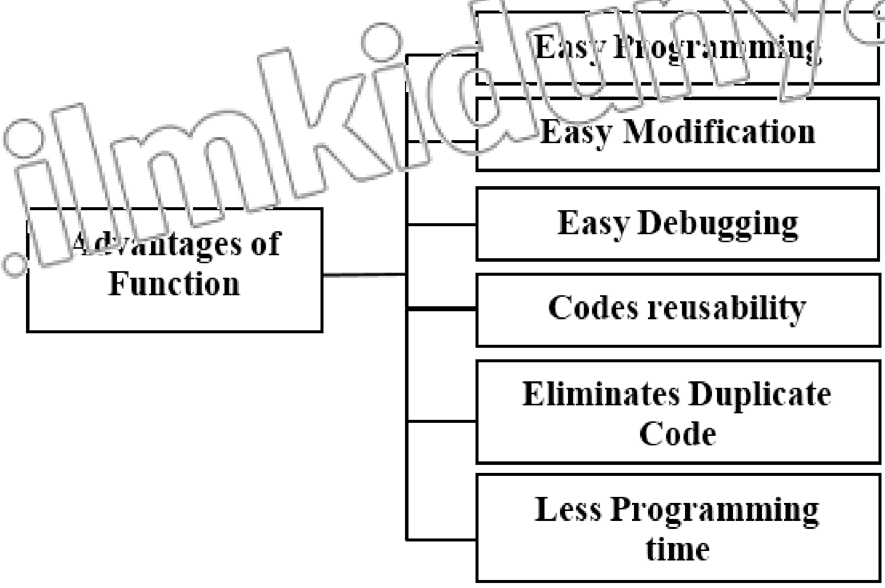
Q.1 What is function? Write the importance of function.

SHORT QUESTIONS

(i) Define Function. **(BWP 2022) (MTN 2021)(K.B)**
(ii) Write about unstructured programming languages. **(BWP 2022)(U.B)**
(iii) Describe structure programming. **(LHR2021)(GRW 2017)(K.B)**
(iv) Write a note on unstructured programming. **(RWP2022)(U.B)**
(v) Write any two benefits of functions. **(U.B)**

FUNCTION
Def:

"A function is a self-contained piece of code that perform a specific operation." Functions are used to accomplish the similar kinds of tasks again and again without writing the same code into the program. Functions perform tasks that may need to be repeated many times.



ADVANTAGES OF FUNCTION/IMPORTANCE OF FUNCTION:

Function is a piece of code designed to perform specific task. There are many advantages of using functions. These advantages are described below.

- **Easy Programming:**

Every program is developed to solve some specific problem. If problem is large the number of instructions in a program increases. It is difficult to write a large program as a single unit. By dividing a large program into functions, programming becomes easy.

- **Easy Modification:**

  If a large program becomes simple and if it is divided into functions, it is easy to understand the logic of program. After development if we want to make a modification, it becomes easy. Instead of changing entire program, we can modify relevant function.

- **Easy Debugging:**

  The process of finding and removing errors in a program is called debugging. Debugging becomes easy if our program consists of different functions. Instead of checking the whole program we can only check the function not working properly.

- **Reusability:**

  A function written for one program can be used in another program. Other programmer can use a function written by one. This is called reusability. It reduces the program development time.

- **Eliminates Duplicate Code:**

  A function reduces program length. To perform same task in a program at different places, we write a function to perform that task. We call the function at desired places in the program. To call a function a single instruction is required. So to perform same task at different places we do not need to write code at all places. By using functions over all length of program decreases.

- **Less Programming Time:**

  A program consists of several functions and each function is an independent program. More than one programmer can work on different functions to develop a program, so less time is required to develop a program.

STRUCTURED PROGRAMMING

In structured programming languages the entire logic of the program is divided into number of smaller modules or functions where each module or function implements a different functionality.

UNSTRUCTURED PROGRAMMING

In unstructured programming languages the entire logic of the program is implemented in a single module or function, which makes the program error prone, difficult to understand, modify and debug.

MULTIPLE CHOICE QUESTIONS

Q.No.1

(i) Which perform tasks that may need to be repeated many times?

(a) If statement (b) Condition (c) goto **(d) Function**

(ii) Which Functions are packaged in libraries?

(a) Built - in (b) User defined (c) Programmer defined (d) Both (a) & (b)

(iii) In which programming the whole program is divided into number of functions?

(a) Structured **(b)** Unstructured

(c) Object oriented **(d)** None of these

(iv) In which programming the whole program logic was contained in a single module?

(a) Object oriented **(b)** Structured

(c) Unstructured **(d)** None of these

SHORT QUESTIONS

Q.No.2

Q.1 What is function? **(BWP 2022) (MTN 2021)(K.B)**

Ans: A function is a self-contained piece of code that perform a specific operation. Functions are used to accomplish the similar kinds of tasks again and again without writing the same code into the program. Functions perform tasks that may need to be repeated many times.

Q.2. Write any two benefits of functions. **(U.B)**

Ans: The use of functions provides several benefits. Some of them are.

- **Divide Workload:**

  A program can consist of many functions. Different programmers can divide the workload by writing different functions.

- **Reusability:**
  Function can be reused as many times as necessary from different places in the program. Well written functions may be reused in multiple programs.

Q.3 Differentiate between structured and un-structured programing.

Ans: Following are the differentiate between structured and un-structured programing:

STRUCTURED PROGRAMMING (LHR2021)(GRW 2017)	UNSTRUCTURED PROGRAMMING (BWP 2022)
In structured programming languages the entire logic of the program is divided into number of smaller modules or functions where each module or function implements a different functionality.	In unstructured programming languages the entire logic of the program is implemented in a single module or function, which makes the program error prone, difficult to understand, modify and debug.

2nd YEAR COMPUTER

REVISION LECTURE NOTES

CHAPTER NO.13

(FUNCTIONS IN C)

LECTURE NO.2 **TIME: 30 MIN.**

PREVIEW OF TODAY'S TOPICS

B. Types of functions,

TYPES OF FUNCTIONS

EXTENSIVE QUESTION

Q.1 What is function? Write a detail note on its different type of function.

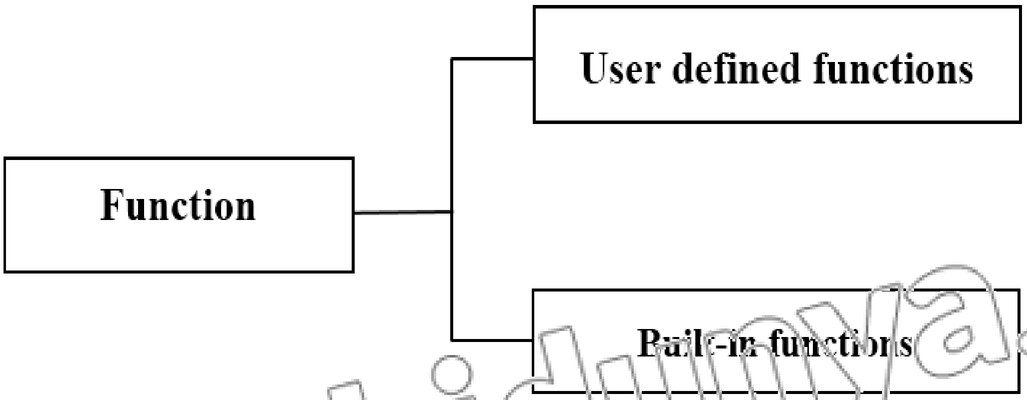
SHORT QUESTIONS

(i) What is function? **(MTN 2021)(K.B)**
Or why is function used in program? **(RWP2019)(U.B)**
(ii) Give an example of user defined and built in function. **(SWL 2019) (U.B)**
(iii) List different types of functions. **(DGK2021)(U.B)**
(iv) Define function declaration with its syntax. **(LHR 2017(K.B)**
(v) How a function is activated? **(GRW 2017(U.B)**

FUNCTION

Def: In structured programming the program consists of more than one parts. Each part of program is called a module or function. Every function is given a unique name and it is developed to perform a specific task. So function can be defined as “A named piece of code developed to perform a specific task is called function”.

TYPES OF FUNCTIONS:



- User Defined Functions:
- Built-in Functions:

- **User Defined Functions:**
  - **i.** The functions that are written by the programmer to perform specific task are called user defined functions.
  - **ii.** These functions are also called programmer defined functions.
  - **iii.** These functions are written according to the requirement of the program.
- **Built-In Functions:**
  - **i.** Built-in functions are predefined functions that provide us convenient ways to perform variety of tasks. These functions are packaged in libraries.
  - **ii.** The functions that are provided as a part of C Language are called built-in functions.
  - **iii.** These functions are also called library functions.
  - **iv.** A large number of built-in functions are provided by C Language.
  - **v.** These functions are stored in different header files.
  - **vi.** If we want to use a built-in function in a program the relevant header files is included at the start of program. Preprocessor directive “include” is used for this purpose.

Examples:

Built-in functions make programming faster and easier:

- printf()
- scanf()
- clrscr()

Function prototyping:

i. The compiler must know functions used in the program.

ii. That's why we include corresponding header files in the source program before using built-in functions such as stdio.h and conio.h etc.

iii. A header file contains the prototypes of the function provided by the library.

iv. The compiler actually needs enough information to be able to identify the function that we are using.

v A function prototype is a statement that provides the basic information that the compiler needs to check and use a function correctly.

vi. It specifies the parameters to be passed to the function, the function name and the type of the return value.

Syntax:

The general form of the function prototype is as follows:

return-type function-name (parameters-list);

The main difference between the function header and function prototype is that function prototype end with a semicolon.

i. The prototype for a function which is called from another function must appear before the function call statement.

ii. Functions prototypes are usually placed at the beginning of the source file just before the function header of the main functions.

MULTIPLE CHOICE QUESTIONS

Q.No.1

(i) Function prototype for built-in functions are specified in: **(SWL 2021)**

(a) Source File (b) Object File

(c) Image File (d) **Header File**

(ii) A type of function written by the programmer is known as: **(LHR 2017)**

(a) **user-defined** (b) subprograms

(c) subroutines (d) built-in-function

(iii) How many types of functions in C languages?

(a) Built - in **(b)** User defined

(c) Programmer defined **(d) All of these**

(iv) printf (),scanf() is an example of which functions?

(a) User defined **(b) Built – in**

(c) Custom _ built **(d)** C++ functions

SHORT QUESTIONS

Q.No.2

Q.1 What are built in functions?

Ans: Built-in functions are predefined functions. These are used to solve different problems. They make programming easier and faster. These are packaged in libraries.

For example, printf, scanf, getch etc.

Q.2 What are user defined functions? **(SWL 2019)(SGD 2019-2021)(K.B)**

Ans: The functions that are written by the programmer depending on the nature of the problem being solved is called user defined function.

Q.3 What is meant by function prototype?

Ans: The functions that are written by the programmer depending on the nature of the problem being solved is called user defined function. Function, the function name and the type of the return value.

Syntax: The general form of the function prototype is as follows:

return-type function-name (parameters-list);

The main difference between the function header and function prototype is that function prototype end with a semicolon.

Q.4 Differentiate between Built in and user defined functions.

Ans: Following are the differentiate between built in and user defined functions.

Built in Function	Under defined
Built-in functions are predefined functions. These are used to solve different problems. They make programming easier and faster. These are packaged in libraries. For example, printf, scanf, getch etc.	The functions that are written by the programmer depending on the nature of the problem being solved is called user defined function. Function, the function name and the type of the return value. **Syntax:** The general form of the function prototype is as follows: **Return-type Function-name (parameters-list);** The main difference between the function header and function prototype is that function prototype end with a semicolon.

REVISION LECTURE NOTES

CHAPTER NO.13

(FUNCTIONS IN C)

LECTURE NO 3 **TIME: 30 MIN.**

PREVIEW OF TODAY'S TOPICS

C. Writing functions in C, Function prototype, Calling a function

WRITING FUNCTIONS IN C, FUNCTION PROTOTYPE CALLING A FUNCTION

EXTENSIVE QUESTION

Q.1 Explain Function writing in C Language.

SHORT QUESTIONS

(i) What is function call? **(DGK 2022)(MTN 2022)(GRW2019-2022)(BWP 2021-22)(K.B)**
Or (How function is activated?)

(ii) What is function header? **(SWL 2021-2022)(LHR 2018))(K.B)**

(iii) What is function definition? **(BWP 2022) (RWP 2019)(GRW 2018) (K.B)**

(iv) Write a note on function body. **(FSD2021) (GRW 2021)(K.B)**

(v) What is return statement? **(SWL 2022)(K.B)**
Or why return statement is used in function?

(vi) What is function prototype? **(MTN 2022)(BWP 2019)(GRW 2018)(SGD 2016) (K.B)**
Or Define function declaration.

(vii) How many (maximum) values can a function return using return statement?

WRITING FUNCTION IN C

Def: We are familiar with the main () function, which is the mandatory part of every C program. Every function in C has almost the same basic structure.

A function in C consists of a function header which identifies the function followed by the body of the function between curly braces containing the executable code for the function.

Syntax:

Every function in C is written according to the following general form:

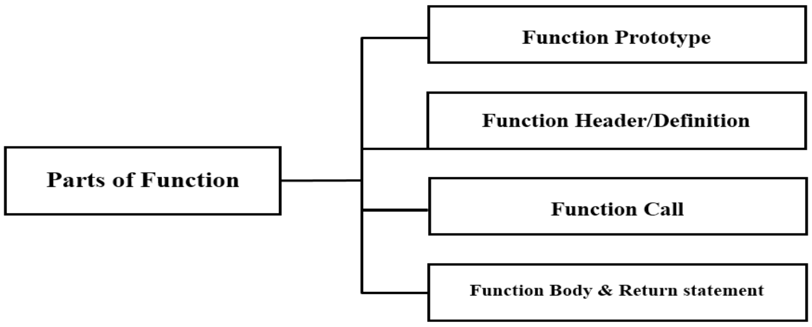
return type Function Name (parameter list)

{

Executable Statement (s)

return expression;

}



- **Function Prototype**
  - A function prototype is a statement that provides the basic information that the

compiler needs to check and use a function correctly. It specifies the parameters to be passed to the function, the function name and the type of the return value.

The general form of the function prototype is as follows

Return-type Function-name(parameters-list);

- The main difference between the function header and function prototype is that function prototype end with a semicolon.

- **Function Header:**

The first line of function definition is called the function header i.e.

Syntax:

return_type FunctionName (parameter_list)

It consists of three parts:

- **The type of return value**
- **The name of the function**
- **The parameters of the function enclosed in parentheses.**

i. The **return_type** can be any valid data type. If the function does not return a value, the return type is specified by the keyword void.

ii. A function that has no parameter specifies the **keyword void** as its parameter list. Hence, a function that has no parameter and does not return any value to the calling function will have the header:

void FunctionName (void)

iii. However the keyword void is optional. The above function header for a function that has no argument can be written as follows:

void FunctionName ( )

- **The Function Body:** **(FSD2021) (GRW 2021)(K.B)**
  - Variables declaration and the program logic are implemented in the function body.
  - Function body makes use of the arguments passed to the function.
  - It is enclosed in curly braces.
  - A function can be called in the body of another function.

- **The return Statement:**

The return statement is used to specify the value returned by a function.

Syntax:

The general form of return statement is:

return [expression];

- When the return statement is executed, expression is evaluated and returned as the value of the function.
- Execution of the function stops when the return statement is executed, even if there are other statements still remaining in the function body.
- If the type of the return value has been specifying as void in the function header
- then there is no need to use a return statement.

- **Calling a Function:**
  - Function call is a mechanism that is used to invoke a function to perform a specific task.
  - A function call can be invoked at any point in the program.
  - In C the function name, the arguments required and the statement terminator (;) are specified to invoke a function call.
  - When function call statement is executed, it transfers control to the function that is called.
  - The memory is allocated to variables declared in the function and then the statements in the function body are executed.
  - After the last statement in the function is executed, control returns to the calling function.

MULTIPLE CHOICE QUESTIONS

Q.No.1

(i) **Which provides information about the function to the complier** **(GRW 2021)**

(a) Function header　　(b) Function Body

(c) Function call　　**(d) Function Prototype**

(ii) **Which statement is used by function to return a value?** **(FSD 2021)(GRW 2017)**

OR They keyword used to specify the valued returned by a function is:

(a) Give　　(b) Send

(c) **Return**　　(d) Call

(iii) **Which mechanism that is used to invoke a function to perform a specific task?**

OR the statement that activates a function known as: **(LHR 2021) (SWL 2019)**

(a) Function call　　(b) Function prototype

(c) Local variable　　(d) Global variable

(iv) **Declaration of function is called:**

(a) Function header　　(b) Function definition

(c) Function call　　**(d) Function prototype**

(v) **A function can not return more than how many value(s) through return statement?**

(a) Two　　(b) Three

(c) Nine　　**(d) One**

(vi) **First line of the function definition is called:**

(a) Function header　　(b) Function definition

(c) Function call　　(d) Function prototype

SHORT QUESTIONS

Q.No.2

Q.1 **What is function call?** **(DGK 2022)(MTN 2022)(GRW2019-2022)(BWP 2021-22)(K.B)**

Or (How function is activated?)

Ans: Function call is a mechanism that is used to invoke a function to perform a specific task. A function call can be invoked at any point in the program. In C the function name, the arguments required and the statement terminator (;) are specified to invoke a function call. When function call statement is executed, it transfers control to the function that is called.

Q.2 **What is function header?** **(SWL 2021 2022)(LHR 2018)(K.B)**

Ans: The first line of function definition is called the function header i.e.

Syntax:

return type FunctionName (parameter_list)

It consists of three parts:

- The type of return value
- The name of the function
- The parameters of the function enclosed in parentheses.

Q.3 **What is function definition?** **(BWP 2022) (RWP 2019)(GRW 2018) (K.B)**

Ans: A function definition consists of executable code written in curly braces that explain what a function does.

The general form to write function is as follows

```
return-type Function-name (parameters-list)
{
        executable statement(s)
        return expression;
}
```

Where the first line of function definition is called function header.

Q.4 **What is return statement?** **(SWL 2022)(K.B)**

Or Why return statement is used in function?

Ans: The return statement is used to specify the value returned by a function.

The general form of return statement is as follows:

return (expression);

When the return statement is executed, expression is evaluated and returned as the value of the function. If the type of return value has been specified as void in the function header then there is no need to use a return statement. Return statement is the last statement of the function body.

Q.5 What is function prototype? **(MTN 2022)(BWP 2019)(GRW 2018)(SGD 2016) (K.B)**

Ans: A function prototype is a statement that provides the basic information that the compiler needs to check and use a function correctly. It specifies the parameters to be passed to the function, the function name and the type of the return value.

The general form of the function prototype is as follows:

Return-type Function-name(parameters-list);

The main difference between the function header and function prototype is that function prototype end with a semicolon.

REVISION LECTURE NOTES

CHAPTER NO.13

(FUNCTIONS IN C)

LECTURE NO.4 **TIME: 30 MIN.**

PREVIEW OF TODAY'S TOPICS

E. Local variable and their scope & Global variable and their scope

LOCAL VARIABLE AND THEIR SCOPE & GLOBAL VARIABLE AND THEIR SCOPE

EXTENSIVE QUESTION

Q.1 What are local and global variable? Explain with the help of examples.

SHORT QUESTIONS

(i) **Define scope of variable.** **(GRW-2022)(K.B)**

(ii) **Write a difference between Global and Local Variable?** **(MTN2021)(SGD2019-2021)(U.B)**

(iii) **Define automatic variable? (define local variable)** **(LHR2022)(DGK2022)BWP(2021)(K.B)**

(iv) **Define global variable.** **(SWL 2021) (GRW 2019)(K.B)**

Or what type of variables are declared outside any function? **(LHR 2021)(U.B)**

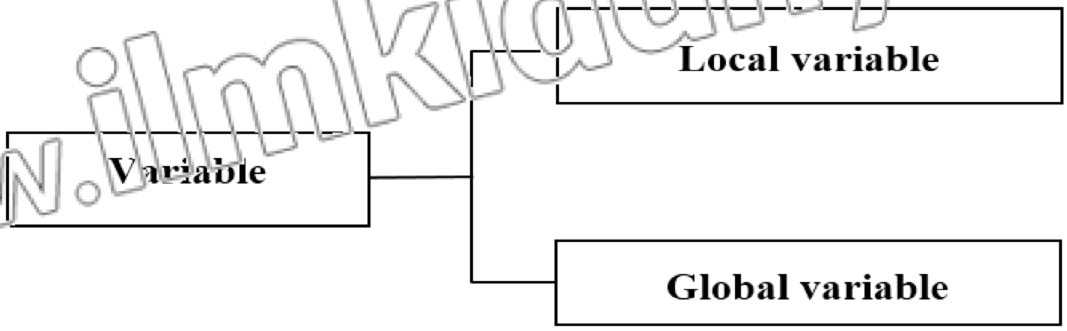
VARIABLE

Def: A variable is a name given to a memory location. It is the basic unit of storage in a program. The value stored in a variable can be changed during program execution. A variable is only a name given to a memory location; all the operations done on the variable effects that memory location.

"The named memory locations used to store input data and result, during the execution of the program is called variable".

LOCAL VARIABLES:

"The variables declared inside main () function, inside any user defined function or header of function definition are called local variables."



Local variables are also called automatic variables.

Syntax

The general syntax of declare a local variable is as follows.

auto data-type variable-name;

The use of auto keyword is optional, local variables can also be declared as:

SCOPE OF LOCAL VARIABLE:

- The scope of a variable refers to the region of a program in which it is accessible.
- The name of a variable is only valid within its scope.
- So a variable cannot be referred outside its scope.

- Local variables have a limited scope.
- They can only be used in the function in which they are declared.
- Compiler generates an error if we want to access a local variable, outside its scope.
- Local variables and have local scope.

LIFE TIME OF LOCAL VARIABLES: **(SWL 2022)(FSD 2021) (GRW 2021) (K.B)**

- The duration in which a variable exists in memory is called lifetime of the variable.
- Life time of local variables is also limited, when control enters in the function and variable declaration statement executed, they are created in memory.
- When the control exits from the function these variables are destroyed and their life ends when variables are destroyed the data stored in them also becomes in accessible.

GLOBAL VARIABLES:

- The variables that are declared outside main( ) function or any other function are called global variable.
  - Global variables are called external variables. Global variables can used by all functions in the program. All functions can share their value.
  - If value of a global variable is changed in a function that changed value is also available in other functions.

SCOPE OF GLOBAL VARIABLE:

- Global variables can be accessed in all modules of program.
- They are accessible in main () function as well as all other user defined functions.
- Global variable have global scope.

LIFE TIME OF GLOBAL VARIABLES:

- When program starts execution, global variables are created in memory.
- They remain in memory till the termination of program.
- When the program is terminated global variables are destroyed from memory.
- Therefore, life time of global variables is between starting termination of program.

MULTIPLE CHOICE QUESTIONS

Q.No.1

(i) **Global variable are created in :** **(DGK 2021)(BWP2021)**

(a) RAM (b) ROM

(c) Hard Disk (d) Cache

(ii) **Local variable is also called:**

(a) Automatic variable (b) Fixed variable

(c) Global variable (d) Static variable

(iii) **The duration in which variable exists variable memory is called.**

(a) Scope of variable (b) Glob of the variable

(c) Life time variable (d) None of these

(iv) **Which of the following refer to the region of a program in which it is accessible?**

(a) Life time variable (b) Glob of the variable

(c) Local time variable **(d) Scope of the variable**

(v) **The variable which is declared outside all blocks:**

(a) Automatic variable (b) Fixed variable **(c) Global variable** (d) Static variable

(vi) **Life time of the global variable?**

(a) **Until the termination of program** (b) End life of program

(c) End of program (d) None of these

SHORT QUESTIONS

Q.No.2

Q.1 **Define scope of variable.** **(GRW-2022)(K.B)**

Ans: The scope of a variable refers to the region of a program in which it is accessible.

The name of a variable is only valid within its scope. So a variable cannot be referred outside its scope. Local variables have a limited scope. They can only be used in the function in which they are declared. Compiler generates an error if we want to access a local variable, outside its scope. Local variables and have local scope.

Q.2 Write a difference between Global and Local Variable?

(MTN2021)(BWP2022)(MTN 2021)(SGD2019-2021)(U.B)

Ans: Following are the difference between Global and Local Variable:

GLOBAL VARIABLE	LOCAL VARIABLE
The variables which are declared outside the main and all other functions are called global variable.	The variables that are declared within a block that is a pair of curly braces. These are called local variable.

Q.3 Define automatic variable? **(LHR2022)(DGK2022)(BWP 2021) (K.B)**

Ans: The variables declared inside main() function, inside any user defined function or header of function definition are called local variables.

Local variables are also called automatic variables.

Syntax

The general syntax of declare a local variable is as follows.

auto data-type variable-name;

The use of auto keyword is optional, local variables can also be declared as:

Q.4 Define global variable. **(SWL 2021) (GRW 2019)(K.B)**

Or what type of variables are declared outside any function? **(LHR 2021)(U.B)**

Ans: The variables that are declared outside main () function or any other function are called global variable. Global variables are called external variables. Global variables can used by all functions in the program. All functions can share their value. If value of a global variable is changed in a function that changed value is also available in other functions.

Q.5 Write a note on Life time the local variable? **(FSD 2021)(K.B)**

Ans: The duration in which a variable exists in memory is called lifetime of the variable.

Life time of local variables is also limited, when control enters in the function and variable declaration statement executed, they are created in memory. When the control exits from the function these variables are destroyed and their life ends when variables are destroyed the data stored in them also becomes in accessible.

LECTURE NO.5 **TIME: 30 MIN.**

PREVIEW OF TODAY'S TOPICS

F. Function without argument and function that return a value & accepts arguments

FUNCTION WITHOUT ARGUMENT AND FUNCTION THAT RETURN A VALUE & ACCEPTS ARGUMENTS

EXTENSIVE QUESTION

Q.1 What is the meant by function without arguments. Give example.

SHORT QUESTIONS

(i) **Different between actual & formal parameters.** **(MTN-2022)(U.B)**

(ii) **What is actual parameters?** **(BWP2019) (SWL2019)(K.B)**

(iii) **What is meant by function without argument?** **(U.B)**

(iv)

FUCTION WITHOUT ARGUMENTS

This is simplest type of function that return no value and no arguments are passed to the function. The return type of such function is void and the parameter list may be empty or void can be written in parenthesis.

EXAMPLE:

Write a program named Draw_ Asterisks that will print asterisks(*) according to the pattern shown in the following and make a function call from function main to print the asterisks pattern.

```
*********
*******
*****
***
*
```

PROGRAM:

```
#include<stdio.h>
void Draw_Asterisks();
void main()
        {
                Draw_Asterisks();
        }
void Draw_Asterisks()
{
        int outer, inner;
        for(outer = 9; outer >= 1; outer -=2 )
        {
                for(inner = 1; inner <= outer; inner++ )
                    printf( "*" );
                printf("\n");
        }
}
```

FUNCTION THAT RETURN A VALUE AND ACCEPT ARGUMENTS

There are lot of such type of built-in functions in C that return a value and accept arguments for example sqrt(), topper() etc. Now we learn to write these types of functions in C.

The general form of function header that returns a value and accept arguments are:

return_type Function_name (parameter_list);

The return_type specifies the data type of the value that the function returns, parameter_list is a comma separated list which specifies the data type and the name of each parameter in the list

EXAMPLE:

Write a program that prompts the user to enter a number and calls a function Factorial () to compute its factorial. Write the function Factorial () that has one input parameter and returns the factorial of the number passed to it.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
int Factorial (int n);
void main()
{
        int fact , f;
        printf("Enter a number to find factorial : ");
        scanf("%d", &fact);
        f = Factorial(fact);
        printf("Factorial of given number is = %d ", f);
        getch();
}
int Factorial (int n)
{
        int r = 1;
        while (n >= 1)
        {
                r *= n;
                n--;
        }
        return (r);
}
```

EXAMPLE:

Write a program to calculate the area of a triangle by using function i.e. the value of base and altitude is passed to function and function return the area to main function.

```
#include <stdio.h>

float area_of_triangle (int base, int altitude);

void main()
{
        int b, a;
        float area;
        printf("Enter value of base : ");
        scanf("%d", &b);
        printf("Enter value of altitude : ");
        scanf("%d", &a);
        area = area_of_triangle(b, a);

        printf(" Area of triangle is = %f ", area);
```

```
          getch();
     }
     float area_of_triangle(int base, int altitude)
     {
          return(0.5 * base * altitude);
}
```

MULTIPLE CHOICE QUESTIONS

Q.No.1

(i) **Multiple arguments to functions are separated by:** **(SWL 2019)**

(A) Comments (B) semicolons
(C) Colons (D) commas

(ii) **The parameters in function declaration.** **(GRW 2019)**

(a) Actual parameters (b) formal parameters
(c) returned parameters (d) call parameters

(iii) **What is the variable that is used by function to receive an argument:** **(LHR2018)**

(a) Formal parameter **(b)** Actual parameter
(c) Runtimes parameter **(d)** Local parameter

(iv) **The parameters passed to a function in the function call are called:**

(a) Actual parameter **(b)** Formal parameter
(c) Global parameter **(d)** None of these

(v) **Formal parameters are also called:**

(a) Actual **(b) Dummy**
(c) Simple **(d)** None of these

(vi) **The name of actual and dummy parameters.**

(a) May or may not be same **(b)** Must be same
(c) Must be different **(d)** None of these

SHORT QUESTIONS

Q.No.2

Q.1 **Different between actual & formal parameters.** **(MTN-2022)(U.B)**

Ans: **Following are the difference between Actual and formal parameters**

Actual Parameters	Formal Parameter
The variables that are passed to function and specified in the function call statement such variables are called actual parameters.	The parameters that are specified in the function header are called formal arguments or formal parameters of the function and their scope is the body of function. These are also called dummy arguments.

Q.2 **What is meant by function without argument?** **(K.B)**

Ans: This is simplest type of function that return no value and no arguments are passed to the function. The return type of such function is void and the parameter list may be empty or void can be written in parenthesis.

Q.3 **Write the general form of function header that returns a value and accept arguments** **(U.B)**

Ans: The general form of function header that returns a value and accept arguments are

Return_type Function_name (parameter_ist);

The return_type specifies the data type of the value that the function returns, parameter_list is a coma separated list which specifies the data type and the name of each parameter in the list.

Q.4 What is dummy/formal parameters? **(K.B)**

Ans: The parameters that are specified in the function header are called formal arguments or formal parameters of the function and their scope is the body of function. These are also called dummy arguments.

Q.5 What is actual parameters? **(BWP2019) (SWL2019)(K.B)**

Ans: The variables that are passed to function and specifies in the function call statement such variables are called actual parameters

STUDENTS LEARNING OBJECTIVES (SLOs)

MULTIPLE CHOICE QUESTIONS

Knowledge Based Questions

(i) In which programming the whole program is divided into number of functions?

(a) Structured (b) Unstructured

(c) Object oriented (d) None of these

(ii) Which perform tasks that may need to be repeated many times?

(a) If statement (b) Condition

(c) goto **(d) Function**

(iii) Which of the following is the part of function header?

(a) Return type (b) Function name

(c) Parameters **(d) All of above**

(iv) Which is a mechanism that is used to invoke a function to perform a specific task?

(a) Function prototype **(b) Function call**

(c) Function header (d) Function definition

(v) The duration in which a variable exists in memory is called __________ of the variable.

(a) Scope (b) Region
(c) Life time (d) None of these

Understanding Based Questions

(i) Function header consists of how many parts?
(a) 2 **(b) 3**
(c) 4 (d) 5

(ii) If a function does not return value, the keyword __________ is specified.
(a) Return (b) Main
(c) Break **(d) Void**

(iii) The general form of return statement is.
(a) return [expression]; (b) return
(c) return (constant); (d) return – value

(iv) The name of actual and dummy parameters.
(a) May or may not be same (b) Must be same
(c) Must be different (d) None of these

(v) The first line of function definition is called:
(a) Function prototype (b) Function call
(c) Function header (d) Return type

Application Based Questions

(i) Which manages the allocation and de-allocation of memory for program variables?
(a) C **(b) Compiler**
(c) OS (d) Loader

(ii) Which variables exists in memory from the start to the end of the program?
(a) Global (b) Local
(c) Dummy (d) Actual

(iii) Which of the following is true about a function call?
(a) Stops the execution of the program
(b) Transfers control to the called function
(c) Transfers control to the main function.
(d) Resumes the execution of the program.

(iv) Which of the following looks for the prototypes of functions used in a program?
(a) Linker (b) Loader
(c) Compiler (d) Parser

(v) Memory is allocated to a local variable at the time of its:
(a) Declaration (b) Destruction
(c) Definition (d) First reference

SHORT QUESTIONS

Knowledge Based Questions

Q.1 What is function?

Ans: A function is a self-contained piece of code that perform a specific operation. Functions are used to accomplish the similar kinds of tasks again and again without writing the same code into the program. Functions perform tasks that may need to be repeated many times.

Q.2 What are built in functions?

Ans: Built-in functions are predefined functions. These are used to solve different problems. The make programming easier and faster. These are packaged in libraries.

Example:

printf, scanf, getch etc.

Q.3 What is function header?

Ans: The first line of function definition is called function header.

The syntax of function header is as:

Return-type Function-name(parameters-list)

The function header consists three parts.

(i) Return-type of value
(ii) Function-name
(iii) parameters of function

Q.4 What is return statement?

Ans: The return statement is used to specify the value returned by a function. The general form of return statement is as follows

return (expression);

When the return statement is executed, expression is evaluated and returned as the value of the function. If the type of return value has been specified as void in the function header, then there is no need to use a return statement. Return statement is the last statement of the function body.

Q.5 What is function prototype?

Ans: A function prototype is a statement that provides the basic information that the compiler needs to check and use a function correctly. It specifies the parameters to be passed to the function, the function name and the type of the return value.

The general form of the function prototype is as follows

Return-type Function-name(parameters-list);

The main difference between the function header and function prototype is that function prototype end with a semicolon.

Understanding Based Questions

Q.1 How can you write functions in C?

Ans: Every function in C has almost the same basic structure as main function. A function is C consists of a function header which identifies the function followed by the body of function written in curly braces containing the executable code of the function.

The general form to write function is as follows:

```
        Return-type Function-name(parameters-list)
{
executable statement(s)l
return expression;
}
```

Q.2 What are the differences of local and global variables?

- **Local Variables**

  The variables that are declared within a block that is a pair of curly braces. These are called local variable.

- **Global variable**

  The variables which are declared outside the main and all other functions are called global variable.

Q.3 What is the scope of a local variable?

Ans: The scope of a local variable refers to the region of a program in which a variable is accessible. So a variable cannot refer outside its scope. Any attempt to do so will cause a compiler error.

Q.4 Differentiate between formal and actual parameters of a function.

Ans:

(i) Formal parameters

The parameters that are specified in the function header are called formal arguments or formal parameters of the function. These are also called dummy arguments.

(ii) Actual parameters

The variables that are passed to function and specifies in the function call statement such variables are called actual parameters.

Q.5 How Function is declared in C language?

Ans: Function declaration is also called function prototype. A function declaration is a statement that provides the basic information that the compiler needs to check and use a function correctly. It specifies the parameters to be passed to the function, the function name and the type of the return value.

The general form of the function declaration is as follows

Return-type Function-name (parameters-list);

Application Based Questions

Q.1 State the purpose of a function argument.

Ans: The arguments that specified in the function call statement, these arguments are called actual arguments. The values of these arguments are copied in the formal arguments of the functions. The function uses its formal arguments for processing data passed to it.

Q.2 How many (maximum) values can a function return using RETURN statement?

Ans: A function can return only a single value. The return type in function prototype indicates the type of value returned by a function.

Q.3 When is a function executed, and where should a function prototype and function definition appear in a source program?

Ans: When a function is called in the program. The control is transferred to said function and the statements in the function body are executed.

Function prototype is usually placed in the beginning of the source file just before the function header of the main function.

The function definition can be written at the following places

(i) Before the main function

(ii) After the main function.

(iii) In a separate file.

Q.4 How a function call made in a C program? Discuss briefly.

Ans: The statement that is used to invoke a function to perform its specific task is called function call. A function can be called at any point in the program. The function name, arguments required and the statement terminator are specified to function call.

Q.5 Write the general form of function header that returns a value and accept arguments

Ans: The general form of function header that returns a value and accept arguments are

Return_type Function_name (parameter_ist);

The return_type specifies the data type of the value that the function returns, parameter_list is a coma separated list which specifies the data type and the name of each parameter in the list.