Chapter: 13

Functions in C

www.pakcity.org

Multiple Choice Questions

Q 1: Following perform tasks that may need to be repeated many times:

Ⓐ Condition Ⓑ Module Ⓒ Program **Ⓓ Function**

Q 2: In C Language, first line of the function definition is known as:

Ⓐ Function header Ⓑ Arguments Ⓒ Function body Ⓓ Parameters

Q 3: In the whole logic of program is contained in main function, it is called:

Ⓐ Structured programming Ⓑ Object-oriented programming
Ⓒ Un-structures programming Ⓓ Modular programming

Q 4: Which of the following is the advantage of function?

Ⓐ Easy to write program Ⓑ Reusability
Ⓒ Eliminate duplicate **Ⓓ All of these**

Q 5: Another name for predefined functions is:

Ⓐ User defined Ⓑ Custom built **Ⓒ Built in** Ⓓ Modal

Q 6: Built-in functions make our task:

Ⓐ Complex Ⓑ Length **Ⓒ Simple and easy** Ⓓ Technical

Q 7: Another name of built in function is:

Ⓐ Library function Ⓑ Arithmetic function Ⓒ User defined function Ⓓ All of these

Q 8: The predefined functions that are part of C language are called:

Ⓐ User defined **Ⓑ Subprograms** Ⓒ Subroutines Ⓓ Built- in functions

Q 9: A type of function is written by the programmer is known as:

Ⓐ User-defined Ⓑ Subprograms Ⓒ Subroutines Ⓓ Built-in-function

Q10: A built - in function:

Ⓐ Cannot be redefined Ⓑ Cannot return a value
Ⓒ Can be redefined Ⓓ Should be redefined

Q11: The parameters specified in the function header are called:

Ⓐ Formal parameters Ⓑ Default parameters
Ⓒ Actual parameters Ⓓ Original parameters

Q12: A block of code is surrounded by:

Ⓐ ( ) Ⓑ [ ] **Ⓒ { }** Ⓓ " "

Q13: A function that does not return any thing has return type:

Ⓐ Nothing Ⓑ Float **Ⓒ Void** Ⓓ Null

Q14: The actual body of the function is defined in:

Ⓐ Function declaration **Ⓑ Function definition**
Ⓒ Function call Ⓓ Function header

Q15: Following term of a function describes the number and type of its arguments and the return of the function:

Ⓐ Size Ⓑ Name **Ⓒ Header** Ⓓ Type

Q16: The first line of function definition is known as:

Ⓐ Function body Ⓑ Function call
Ⓒ Function header Ⓓ Function argument

Q17: What is true about a function prototype?

Ⓐ It is also referred to as function declaration Ⓑ It is a single statement
Ⓒ It is terminated with a semicolon (;) **Ⓓ All of these**

Q18: Function declaration consists of:

Ⓐ Function name Ⓑ Function return type
Ⓒ Number and types of parameters **Ⓓ All of these**

Q19: A function:

Ⓐ May return more than one values Ⓑ Cannot return any value
Ⓒ May return only one value Ⓓ May return two values

Q20: The actual values are passed to the function in:

Ⓐ Function declaration Ⓑ Function definition
Ⓒ Function call Ⓓ Called function body

Q21: A function is called with help of its:

Ⓐ Name Ⓑ Parameter Ⓒ Definition Ⓓ Return value

Q22: The statement that activates a function is called:

Ⓐ Function call Ⓑ Function output Ⓒ Invoking a function Ⓓ Function input

Q23: Which of the following looks for the prototype of functions, when a function is called?

Ⓐ Linker Ⓑ Loader **Ⓒ Compiler** Ⓓ Parser

Q24: Actual argument is used in:

Ⓐ Function declaration **Ⓑ Function call**
Ⓒ Function prototype Ⓓ Function definition

Q25: The scope of a variable refers to its:

Ⓐ Length Ⓑ Name **Ⓒ Accessibility** Ⓓ Data type

Q26: The region of a program in which a variable is accessible is called:

Ⓐ Area **Ⓑ Scope** Ⓒ Function Ⓓ Use

Q27: Local variable are also called:

Ⓐ Automatic variable Ⓑ Register variable
Ⓒ Static variable Ⓓ Run time variable

Q28: The variables declared inside any function are known as:

(A) Global variables
(B) External variables
(C) Private variables
(D) Local variables

Q29: Memory is allocated to a local variable at the time of its:

(A) Declaration (B) Destruction **(C) Definition** (D) First reference

Q30: Data can be shared between functions using:

(A) Local variable (B) Static variable **(C) Global variable** (D) Register variable

Q31: The variables that are declared outside all blocks are called:

(A) General variables **(B) Global variables** (C) Local variables (D) Global data items

Q32: Global variable are crated in:

(A) RAM (B) ROM (C) Hard disk (D) Cache

Q33: Which of the following type of variables are destroyed when the program is terminated?

(A) Register variables
(B) Automatic variables
(C) Local variables
(D) Global variables

Q34: Function prototypes for built-in functions are specified in:

(A) Source files **(B) Header file** (C) Object file (D) Image file

Q35: Global variables are created in:

(A) RAM (B) ROM (C) Hard disk (D) Cache

Q36: While of the following statement is true about a function call?

(A) Stops the execution of the program
(B) Transfers control to the main function
(C) Transfers control to the called function
(D) Resumes the execution of the program

Q37: Which of the following looks for the prototypes of functions used in a program?

(A) Linker (B) Loader **(C) Compiler** (D) Parser

Q38: Memory is allocated to a local variable at the time of its:

(A) Declaration (B) Destruction **(C) Definition** (D) First reference

Q39: The name of actual and formal parameters:

(A) May or may not be same
(B) Must be different
(C) Must be same
(D) Must be in lowercase

Q40: Formal arguments are also called:

(A) Actual arguments
(B) Original arguments
(C) Dummy arguments
(D) Referenced arguments

Q41: printf () is a:

(A) Built-in function
(B) User-defined function
(C) Local function
(D) Keyword

Q42: A built-in function:

Ⓐ Cannot be redefined
Ⓑ Cannot return a value
Ⓒ Can be redefined
Ⓓ Should be redefined

Q43: In a C program, two functions can have:

Ⓐ Same name
Ⓑ Same name and same parameters
Ⓒ Same parameters
Ⓓ Same name but different parameters

Q44: The process of sending an argument to a function is called:

Ⓐ Sending Ⓑ Filtering Ⓒ Delivering **Ⓓ Passing**

Short Questions

Q1: What is modular programming?

Ans: A programming technique in which a program consists of many independent parts is called modular programming. These parts are called modules. These parts are also called function. Each module can perform different tasks. The development speed of a program increases as different programmers can write different modules of a program. Different modules are combined to make a complete program.

Q2: What is a function?

Ans: In structured programming, the program consists of more than more one part. 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".

Q3: Why functions are used?

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

- Easy programming
- Easy modification
- Easy debugging Reuse-ability
- Eliminates duplicate code
- Less programming time

Q4: What are built-in functions?

Ans: The functions that are provided as a part of C language are called built-in functions. These functions are also called library function. A large number of built- in functions are provided by C language. These functions are stored in different header files. If we want to use a built-in function in a program the relevant header files is included at the start of the program in Preprocessor directive.

Q5: What are user defined functions?

Ans: The functions that are written by the programmer to perform specific task are called user defined functions. These functions are written according to the requirement of the program.

Q6: What is function prototypes?

Ans: Function declaration is also called function prototype. It is a statement that provides basic information to compiler about the structure of the function like other C language

statement; function declaration statement also ends with semicolon. Function declaration is necessary like variable declaration. A function must be declared in a C language program. Function can be declared before the main() function or inside the main() function.

Q7: What is function definition?

Ans: Every function performs some specific task. The task is performed when the set of instructions execute. Writing set of statements of a function is called function definition. Function definition is always done outside main () function.

Q8: What is function header?

Ans: The first line of the function definition is called function header. Its general syntax is as follow:
Return-Type Name(parameters)

Q9: What is function calling?

Ans: The statement that is written to use a function is called function call. A function can be called at any point in the program. A function is called by using its name. The required parameters are maintained after the name in braces at the end of the function call statement. Semicolon is used at the end of statement in which function is called.

Q10: What is return statement?

Ans: Keyword "return" is used to return a value from the body of called function to calling function. The statement in which "return" keyword is used is called return statement. The general syntax for return statement is as follow:
return expression;

Q11: What are parameters?

Ans: Parameters are also called arguments. These are the values that are provided toa function when it is called. when function is called parameters are written after function name in parenthesis. These parameters can be variables or constants. More than one parameter is separated by comma.

Q12: What is a local variable?

Ans: The variables declared inside main() function, inside any user defined function or header of function definition are called local variables. Local variable also called automatic variables. The general syntax to declare a local variable is as follows:
auto data-type variables-name;

Q13: What is a global variable?

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

Q14: What is meant by life time of a variable?

Ans: Lifetime of a local variable is limited, when control enters into the function and variable declaration statement is 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 inaccessible.

Q15: What is meant by scope of a variable?

Ans: 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.

Q16: What is scope of global variable?

Ans: Global variables can be accessed in all modules of program. They are accessible in main() function as well as all other user defined functions.

Q17: What is life time of global variable?

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