COMPUTER SCIENCE 10TH – DETAILED QUESTION ANSWERS

➔ FUNCTIONS

Chapter # 05

ADAMJEECOACHING.BLOGSPOT.COM

Q.1: What do you know about functions?

Ans: INTRODUCTION To FUNCTIONS

A function is a block of code that performs a particular task. It is also called a method or a sub-routine or a procedures etc. There are some situations when we need to write a particular block of codes for more than once in our program. This may lead to bugs and irritation for the programmer. C++ language provides an approach in which you need to declare and define a group of statements once and that can be called and used whenever required. This saves both time and space. Every C++ program has at least one function, which is main( ), and all the programs can define additional functions.

Q.2: Write down the advantages of functions.

Ans: ADVANTAGES OF FUNCTIONS

There are some advantages of using functions.

1. The complexity of the entire program can be divided into simple subtasks and function subprograms can be written for each subtask.
2. Functions help us to avoid unnecessary repetition of codes. It helps in code reusability.
3. Functions can have inputs and outputs and can process information.
4. The functions are short, easier to write.
5. The function are understandable and can be debugged.

Q.3: Define the types of functions.

Ans: TYPES OF FUNCTIONS

There are two types of functions in C++ programming

1. redefined / Built-in functions
2. User-defined Functions

Q.4: What is predefined or built in function?

Ans: PREDEFINED / BUILT IN FUNCTIONS

The built-in functions are standard library functions to handle tasks such as mathematical computations, I/O processing, string handling etc. These functions are defined in the header file and don't have need to declare and define. The definitions of most common functions are found in the cmath and cstdlib libraries.

Q.5: Describe user defined functions.

Ans: USER-DEFINED FUNCTIONS

C++ allows programmers to define functions to do a task relevant to their programs. Such functions created by the user are called user-defined functions. These functions need declaration and definition. A user can create as many user-defined functions as needed.

Q.6: Write down the major elements of user-defined functions

Ans. The user-defined function consists of two parts:

1. Function declaration (Function prototype)
2. Function definition

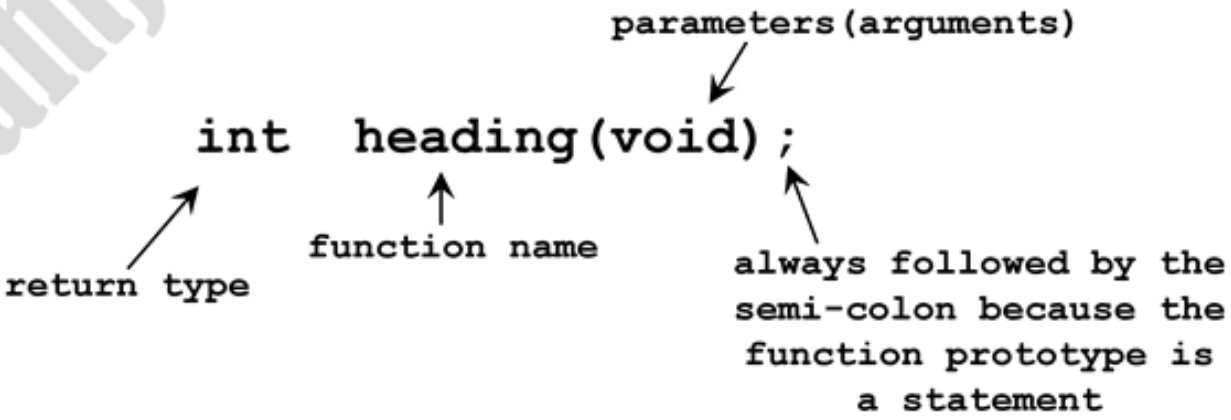
Q.7: Define function declaration.

Ans: FUNCTION DECLARATION / PROTOTYPE

The declaration of a function is called its prototype. Using function in programs requires that we have to declare the function first. It is declared before the main( ) function.

The general structure of the function prototype is:

```
return_datatype function_name(arguments)
```



It has four main components. These are:

1. Return data type
2. Name of the function
3. Arguments (parameters)
4. Statement terminator

RETURN DATA TYPE

The return data type of the function is the type of return value. If function does not return a value, the type is defined as void.

FUNCTION NAME

The function name is any identifier followed by parenthesis without any spaces in between.

ARGUMENTS /PARAMETERS

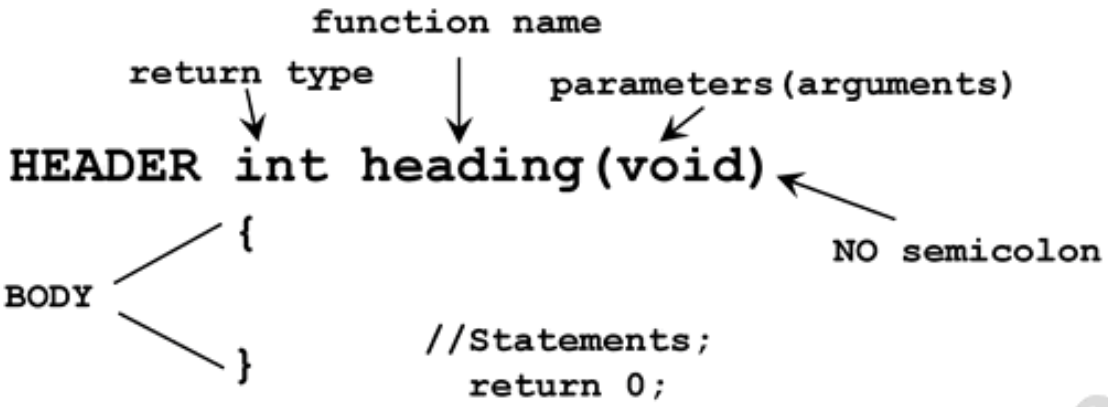
The arguments or parameters come inside the parenthesis, preceded by their types and separated by commas. If the function does not use any arguments, the word void is used inside the parentheses.

STATEMENTTERNUNATOR

Semicolon (;) is used as statement terminator at the end of function declaration or prototype.

FUNCTION DEFINITION

A function definition is the function itself. Function definition consists of a function header, a function body and code block. The definition begins with a header which is exactly same as the function prototype except it must not be terminated with semicolon (;). It has three main components; return type of the function, name of the function, arguments I parameters. Body of the function includes statements in braces and defines before or after main function.



Q.8: Define function calling.

Ans: FUNCTION CALL

A user defined function is called from the main program simply by using its name, including the parentheses which follow the name. The parentheses are necessary so that compiler knows you are referring to a function.

GENERAL SYNTAX

function_name( );

Q.9: Define function arguments.

Ans. FUNCTION PASSING ARGUMENTS OR PARAMETERS

Sending data to a function is called passing arguments. It is basically sending variables, constants, or expression whose value are needed by the function. Actual values that are passing to function as argument with function call statement are known as actual arguments. These values are received in variables of the header of the function definition. These receiving variable or arguments are called formal arguments.

Q.10: Define function return value.

Ans: RETURNING VALUE FROM FUNCTIONS

In C++, when a function completes its execution, it can return a value to the calling function using return keyword. Return type must be defined with the function header in declaration and definition.

Q.11: Write down the differences between function definition and function call.

Ans: DIFFERENCES BETWEEN FUNCTION DEFINITION AND FUNCTION CALL

FUNCTION DEFINITION	FUNCTION CALL
The function definition is the part of function where function is actually defined.	Invoke the code of the function is called function call.
A user defined function may define before or after the main function.	A user defined function is called from the main program simply by using its name.

Syntax: data_type function_name (arguments)
{
statements;
}
	Syntax:
	variable- name function- name (arguments);

Example:
int_sum(int a, int b)
{
int c;
c = a + b;
return(c);
}
	Example:
	z = sum(x, y)

Q.12: Define functions types based on arguments and return value.

Ans. DIFFERENT WAYS TO USE USER-DEFINED FUNCTION BASED ON ARGUMENTS AND RETURN TYPE

Function can be used in four variations in C++ based on arguments and return value from the functions.

- **No return value and no passing arguments**

  void function_name(void);

Return value but no passing arguments

int/float/char function_name(void);

- **No return value but passing arguments**

  void function_name(int, float, char);

- **Return value and passing arguments**

  int/float/char function_name(int, float, char);

Q.13: Write a program using predefined functions.

Ans: EXAMPLE PREDEFINED FUNCTIONS

```
#include<iostream>
#include<cmath.h>
using namespace std;
int main( )
{
  int sq;
  cout << "Enter an integer to find square root: ";
  cin >> sq;
  cout << " \n The Square root of a given integer is: " << sqrt(sq);
  return 0;
}
```


Q.14: Write a program using user-defined functions.

Ans: EXAMPLE USER-DEFINED FUNCTIONS

```
#include<iostream>
using namespace std;
// function prototype
int add(int a, int b);
int main( )
{
  int x, y, sum;
  cout << "Enter 1st number: ";
  cin >> x;
  cout << "Enter 2nd number: ";
  cin >> y;
  sum = add(x, y);
  cout << "The sum of two numbers is = " << sum << "\n";
  return 0;
}
// function definition
```

```
int add(int a, int b)
{
  int c;
  c = a + b;
  return (c);
}
```

Q.15: Write down the differences between predefined function and user-defined function.

Ans: DIFFERENCES BETWEEN PREDEFINED AND USERDEFINED FUNCTIONS

PREDEFINED FUNCTION	USER-DEFINED FUNCTION
It is also called built in functions or library functions.	These are called end-user functions created by programmer.
It cannot be edited or modified	It can be edited or modified by programmer.
Function definition is not required because definition is a part of compiler.	Function declaration and definition are needed in user-defined function.
Example : pow( ), sqrt( ), getche( ), etc.	**Example:** add( ), int sum( ); float avg(float a, float b), etc.

Q.16: Define variables in C++.

Ans: VARIABLES IN USER-DEFINED FUNCTIONS

In C++ structure programming, there are two types of variables used:

1. Local variable
2. Global variable

Q.17: What is local variable? Also write its example.

Ans: LOCAL VARIABLES

Local variable is defined as a type of variable declared within programming block or functions. It can only be used inside the function or code block in which it is declared. The local variable exists until the block of the function is under execution. After that, it will be destroyed automatically.

LOCAL VARIABLES EXAMPLE

```
#include<iostream>
using namespace std;
int main( )
{
  int b;                  // local variable
  cout << "Enter an integer: ";
  cin >> b;
  cout << " \n The given integer is: " << b;
  return 0;
}
```


Q.18: What is global variable? Also write its example.

Ans: GLOBAL VARIABLES

A global variable in the program is a variable defined outside the subroutine or function. It has a global scope means it holds its value throughout the lifetime of the program. Hence, it can be accessed throughout the program by any function defined within the program.

GLOBAL VARIABLES EXAMPLE

```
#include<iostream>
using namespace std;
int add(int a);
int b = 10;          // Global Variable
int main( )
{
  int x, sum;          // Local Variables
  cout << "Enter 1st number: ";
  cin >>  x;
  sum = add(x);
  cout << "The sum of two numbers is = " << sum << "\n";
  return 0;
  }
  // function definition
```

```
int add(int a)
{
int c;                  // Local Variable
c = a + b;
return (c);
}
```