Function:-

Functions are the building blocks of readable, maintainable, and reusable code.

A function is a set of statements to perform a specific task.

Function Declaration:- A function declaration tells the compiler about the function name, function parameters, and return type. The syntax of the function declaration is:

function functionName( [arg1, arg2, ...argN] );

Function definition: It contains the actual statements which are going to executes. It specifies what and how a specific task would be done. The syntax of the function definition is:

function functionName( [arg1, arg2, ...argN] ){ }

3. Function call: We can call a function from anywhere in the program. The

parameter/argument cannot differ in function calling

a function

and

declaration. We must pass the same number of functions as it is declared in the function declaration. The syntax of the function call is:

FunctionName();

ypes of Function

1. Named functions: When we declare and call a function by its given name, then this type of function is known as a named function.

function welcome(){

console.log("Welcome to Career Infotech"); }

welcome( );

2. Anonymous functions : A function without a name is known as an anonymous function. These type of functions are dynamically declared at runtime. It is defined as an expression. We can store it in a variable, so it does not need function names.

let result = function (x:number, y:number) : number { return x+y;

}

result (10,20);

we will use functions difference category of functions as below

1) Simple Function:

Simple function is function which does not have parameter or return Example – function fun(){

Console.log(“hello”); }

fun(); 2) Function Parameters

Function parameter can be categories into the following:

➢ Optional Parameter

o Optional Parameter should be the last argument o Optional Parameter will be represented using ?

function showDetails(id:number,name:string,e\_mail\_id?:string) { console.log("ID:", id, " Name:",name);

if(e\_mail\_id!=undefined) console.log("Email-Id:",e\_mail\_id);

}

showDetails(101,"Virat Kohli"); showDetails(105,"Sachin","sachin@ci.com");

➢ Default Parameter

o If the user does not pass a value to an argument, TypeScript initializes the default value for the parameter.

function displayName(name: string, greeting: string = "Hello") : string {

return greeting + ' ' + name + '!'; }

console.log(displayName('Career')); console.log(displayName('career', 'Hi'));➢ Rest Parameter

o The rest parameter is used to pass zero or more values to a function. o Rules to follow in rest parameter:

▪ Only one rest parameter is allowed in a function.

▪ It must be the last parameter in a parameter list. function sum(a: number, ...b: number[]): number

{

let result = a;

for (var i = 0; i < b.length; i++) { result += b[i];

}

return result; }

let result1 = sum(3, 5);

let result2 = sum(3, 5, 7, 9); console.log(result1 +"\n" + result2);

3) Function return: Functions may also return value along with control, back to the caller. Such functions are called as returning functions.

i. The return\_type can be any valid data type.

ii. A returning function must end with a return statement.

iii. A function can return at the most one value.

iv. The data type of the value returned must match the return type of the function.

4) Arrow function

a. ES6 version of TypeScript provides an arrow function which is the shorthand syntax for defining the anonymous function

b. Syntax

i. Parameters: A function may or may not have parameters.

ii. The arrow notation/lambda notation (=>)

iii. Statements: It represents the function's instruction set.