Top Down Design of Functions

Library Functions

User Defined Functions

Types of Functions

1) Function without argument and no return value

2) Function without argument and return value

3) Function with argument and without return value

4) Function with argument and with return value

Function input arguments: arguments used to pass information into a function subprogram.

Function output arguments: arguments used to return results to the calling function.

Actual Parameter: Actual parameters are the values that are passed to the function during a function call. They are also known as arguments. Actual parameters are used to provide the values to the formal parameters of the function. Actual parameters can be of any data type such as int, float, char, etc.

Formal parameter: Formal parameters are also called function parameters or function arguments. They are used in the function definition to accept the values from the caller of the function. Formal parameters are declared in the function definition and are used to represent the data that will be passed to the function at the time of the function call. Formal parameters can be of any data type such as int, float, char, etc.

#include<stdio.h>

int sum(int x, int y); //

int main(){

int y, a, b;

y = sum(a, b);

printf(“%d\n”, y);

}

sum(int x, int y){

int z;

z = x + y;

return(z);

}

a and b → Actual parameters

x and y → Formal parameters