**Functions**

Table of Contents

[Types of Functions 2](#_Toc65498127)

[Built-in Functions 2](#_Toc65498128)

[User-defined Functions 2](#_Toc65498129)

[Attributes of Functions 3](#_Toc65498130)

[Writing Functions 4](#_Toc65498131)

## Types of Functions

### Built-in Functions

Some functions are already defined inside header files. They can be used after the appropriate header file is included.

### User-defined Functions

These are functions that are created by the user. (int main() is a function.)

## Attributes of Functions

Return type: The type of data that the function returns (output).

Number of parameters: The number of data that the function takes as input.

Types of Parameter: The types of data that the function takes as input.

## Writing Functions

Function that prints new line:

void printline()  
{  
 printf("\n");  
}

C

Here, void is the return type, printline is the name of the function, the data in the bracket is the input (no input here), and the operations inside the second brackets give the output.

void printsum(int a, int b) *// a and b are local variables in this function*{  
 printf("%d", a+b);  
}  
printsum(3, 5); *// prints 8*

C

int givesum(int a, int b)  
{  
 int c = a + b;  
 return c; *// return statement causes function to stop executing*}  
int s = givesum (3, 5); *// stores the integer 8 in the variable s*

C

Function must be defined before it can be used. To fix this, the function must be declared beforehand. Such a declaration is called a prototype. Prototypes are declared like this:

int givesum(int a, int b);

C

This way the compiler knows that a function exists but does not know what to do yet (since it has not been defined yet). So that part of the operation is left incomplete for the time being.

Prototypes help in checking parameter types. They also make organization easier in large codes, and allow use of values that are going to be found later on.

Functions can also be nested.

During implementation, control is passed from one function to the other as needed. To keep track of this, memory is used. Data is stacked (first-in, last-out method). The first function to start executing is the last to finish executing.

If a function is nested within itself (other than main function), the situation is called recursion. They can be used to set up loops.

Function that prints product of sum and difference of two integers:

#include<stdio.h>  
  
int sum (int a, int b)  
{  
 int c = a + b;  
 return c;  
}  
  
int diff (int a, int b)  
{  
 int c = a - b;  
 return c;  
}  
  
int solve (int a, int b)  
{  
 int c = sum(a, b);  
 int d = diff(a, b);  
 return c\*d;  
}  
  
int main()  
{  
 int c, a=5, b=2;  
 c = solve(a, b);  
 printf("%d", c);  
}

C