CL118 Programming Fundamentals

Lab 06

Function in C

Instructors: Ms. Maham Mobin Sheikh
Ms. Atiya jokhio

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

LAB 06

maham.mobin@nu.edu.pk|atiya.jokhio@nu.edu.pk

Functions:

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

Types of functions in C programming

Depending on whether a function is defined by the user or already included in C compilers, there are two types of functions in C programming

There are two types of functions in C programming:

- Standard library functions
- User defined functions

Standard library functions

The standard library functions are built-in functions in C programming to handle tasks such as mathematical computations, I/O processing, string handling etc.

These functions are defined in the header file. When you include the header file, these functions are available for use. For example:

The printf() is a standard library function to send formatted output to the screen (display output on the screen). This function is defined in "stdio.h" header file.

There are other numerous library functions defined under "stdio.h", such as scanf(), fprintf(), getchar() etc. Once you include "stdio.h" in your program, all these functions are available for use.

User-defined functions

As mentioned earlier, C allows programmers to define functions. Such functions created by the user are called user-defined functions.

Depending upon the complexity and requirement of the program, you can create as many user- defined functions as you want.

Benefits of Using Functions

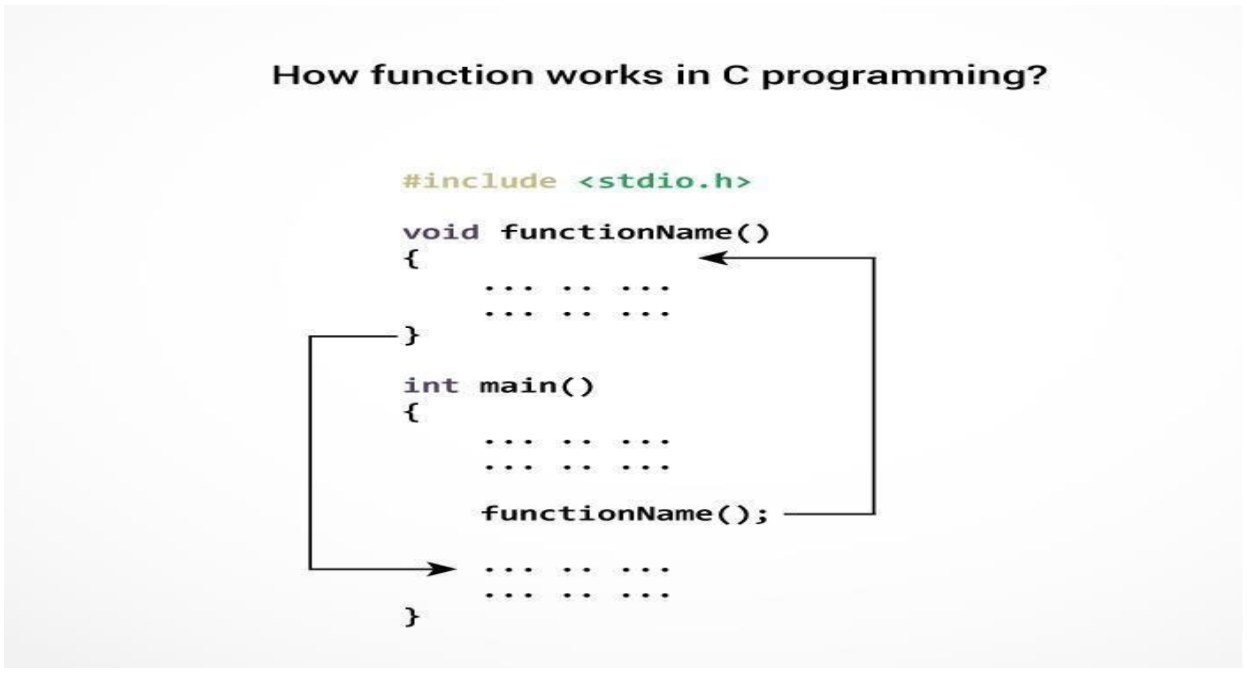
1. It provides modularity to your program's structure.

2. It makes your code reusable. You just have to call the function by its name to use it, wherever required.

3. In case of large programs with thousands of code lines, debugging and editing becomes easier if you use functions.

4. It makes the program more readable and easy to understand.

How user-defined function works?



```
#include <stdio.h>

void functionName()
{
    ... .. ...
    ... .. ...
}

int main()
{
    ... .. ...
    ... .. ...

    functionName();

    ... .. ...
    ... .. ...
}
```

Example: User-defined function

Here is an example to add two integers. To perform this task, a user-defined function addNumbers() is defined.

```
#include <stdio.h>

int addNumbers(int a, int b);       // function prototype

int main()
{ int n1,n2,sum;

    printf("Enters two numbers: "); scanf("%d
    %d",&n1,&n2);

    sum = addNumbers(n1, n2);       // function call

    printf("sum = %d",sum);

    return 0;
}

int addNumbers(int a,int b)       // function definition
{ int result;
    result = a+b;
   return result;               // return statement
}
```

Function prototype

A function prototype is simply the declaration of a function that specifies function's name, parameters and return type. It doesn't contain function body.

A function prototype gives information to the compiler that the function may later be used in the program.

Syntax of function prototype

```
returnType functionName(type1 argument1, type2 argument2,...);
```

In the above example, `int addNumbers(int a, int b);` is the function prototype which provides following information to the compiler:

1. name of the function is `addNumbers()`
2. return type of the function is `int`
3. two arguments of type `int` are passed to the function

The function prototype is not needed if the user-defined function is defined before the `main()` function.

Calling a function

Control of the program is transferred to the user-defined function by calling it.

Syntax of function call

```
functionName(argument1, argument2, ...);
```

In the above example, function call is made using `addNumbers(n1,n2);` statement inside the `main()`.

Function definition

Function definition contains the block of code to perform a specific task i.e. in this case, adding two numbers and returning it.

Syntax of function definition

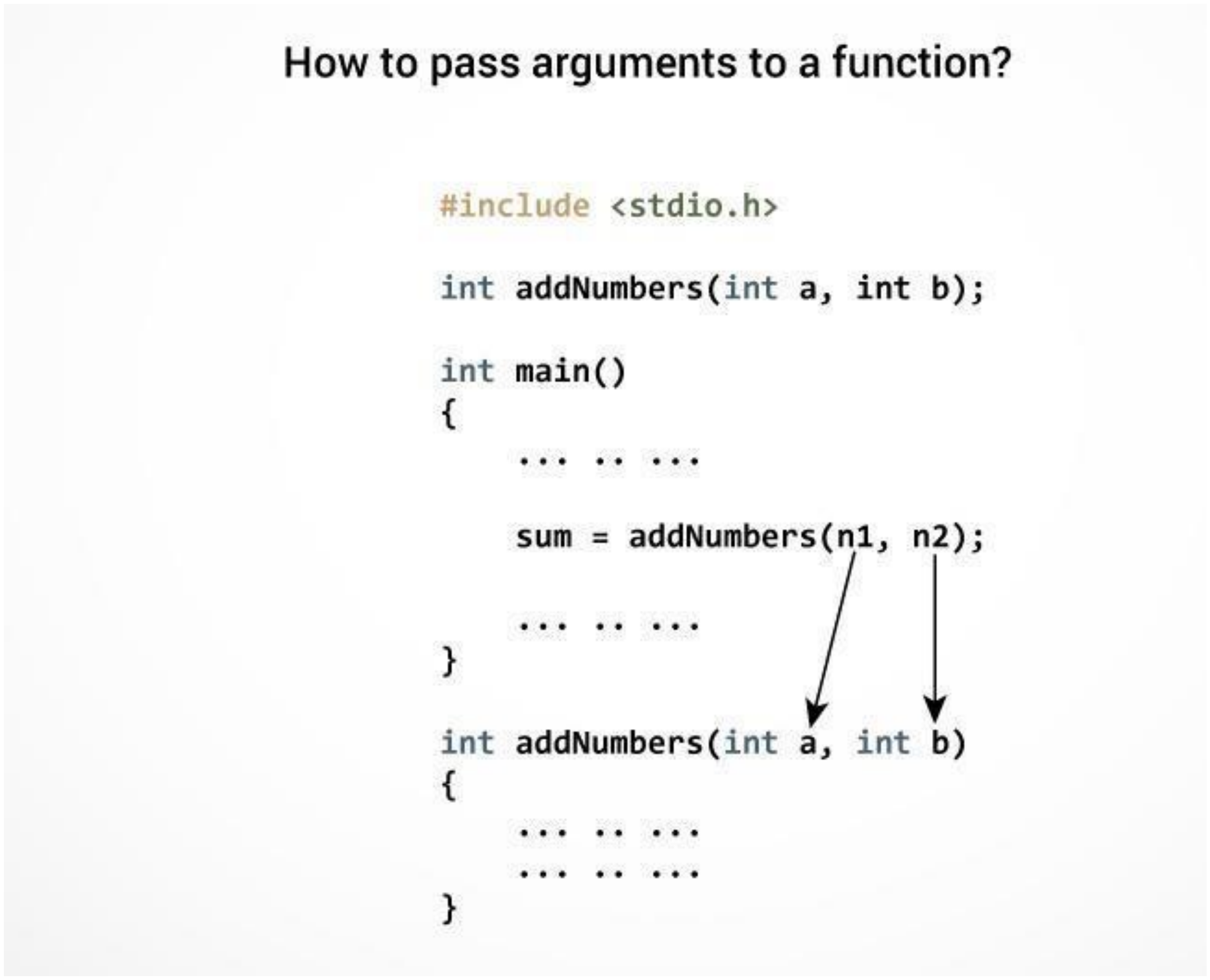
```
returnType functionName(type1 argument1, type2 argument2,
...)
{

    //body of the function



}
```

When a function is called, the control of the program is transferred to the function definition. And, the compiler starts executing the codes inside the body of a function.

Passing arguments to a function

In programming, argument refers to the variable passed to the function. In the above example, two variables n1 and n2 are passed during function call.

The parameters a and b accepts the passed arguments in the function definition. These arguments are called formal parameters of the function.



```c
#include <stdio.h>

int addNumbers(int a, int b);

int main()
{
    ... .. ...

    sum = addNumbers(n1, n2);

    ... .. ...
}

int addNumbers(int a, int b)
{
    ... .. ...
    ... .. ...
}
```

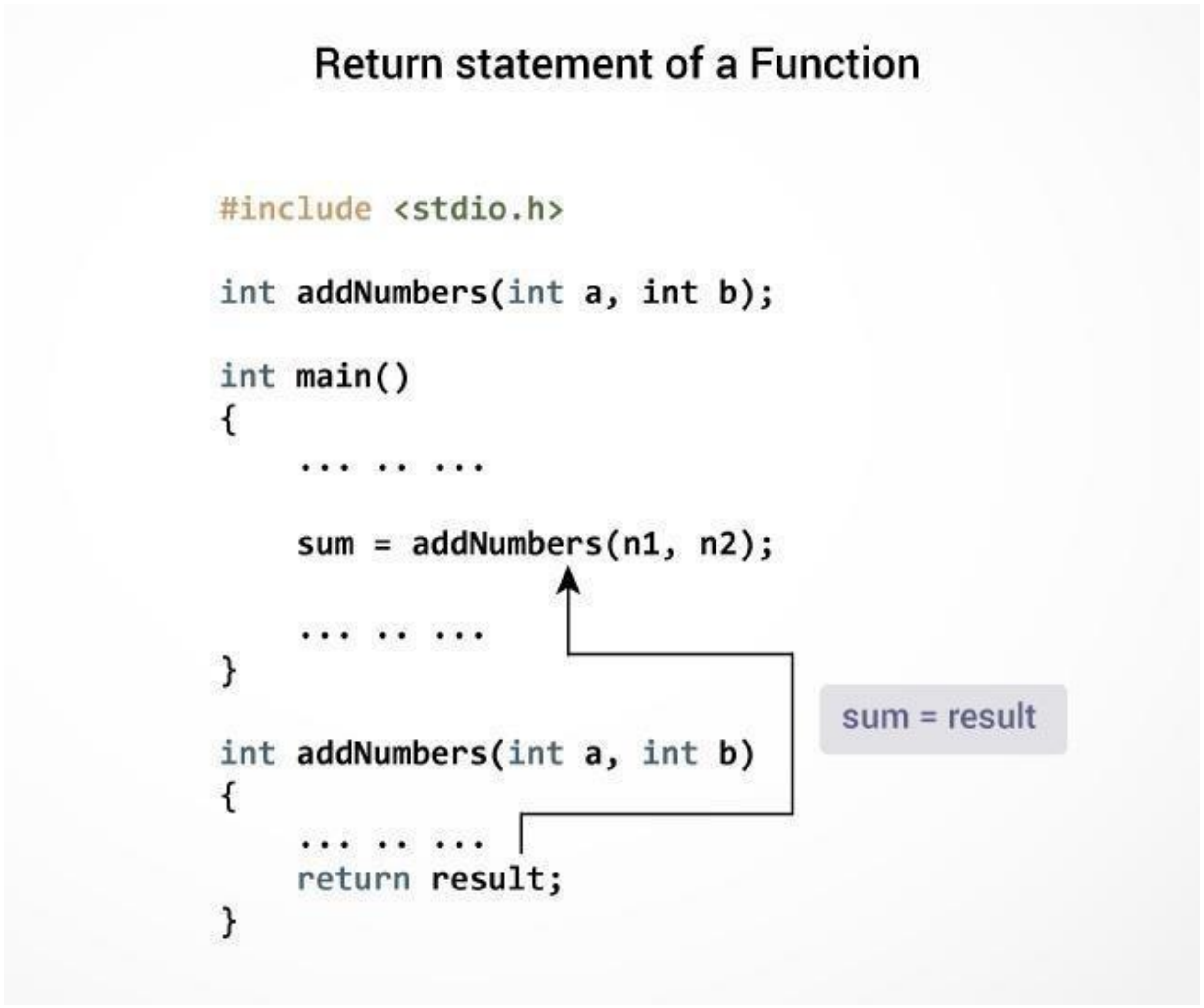
The type of arguments passed to a function and the formal parameters must match, otherwise the compiler throws error.

If n1 is of char type, a also should be of char type. If n2 is of float type, variable b also should be of float type.
A function can also be called without passing an argument.

Return Statement

The return statement terminates the execution of a function and returns a value to the calling function. The program control is transferred to the calling function after return statement.

In the above example, the value of variable result is returned to the variable sum in the main() function.



```
#include <stdio.h>

int addNumbers(int a, int b);

int main()
{
    ... .. ...

    sum = addNumbers(n1, n2);

    ... .. ...
}

int addNumbers(int a, int b)
{
    ... .. ...
    return result;
}
```

Syntax of return statement

```
return (expression);
```

For example,

```
   return a;
return (a+b);
```

The type of value returned from the function and the return type specified in function prototype and function definition must match.

User Defined Header File

The purpose to understand and learn header file is that, it also contains specific function define in it. You are required to call its header and can use its defined function in your program.
Meanwhile header file serves two purposes.

- System header files declare the interfaces to parts of the operating system. You include them in your program to supply the definitions and declarations you need to invoke system calls and libraries.
- Your own header files contain declarations for interfaces between the source files of your program. Each time you have a group of related declarations and macro definitions all or most of which are needed in several different source files, it is a good idea to create a header file for them.

Follow the step to create your header file:
Suppose we want to make header for sum function.

1. Make a header file with .h extension and give it unique name e.g sumfile-> sumfile.h
2. Define your program in header extension file.
   a. `int add(int a,int b){return(a+b);}`
3. Make source file of c, where your main program is set.
   a. `#include<stdio.h>`
   b. `#include "sumfile.h" // don’t use ‘<>’, instead of it use””.`
   c. `void main()`
   d. `{ int num1 = 10, num2 = 10, num3;`
   e. `num3 = add(num1, num2);`
   f. `printf("Addition of Two numbers : %d", num3);}`
4. Keep .h file path directory same as source file.
5. In the above program the ‘add’ function is basically called from the heder file of sumfile.h Which we have explicitly defined.

LAB ACTIVITY:

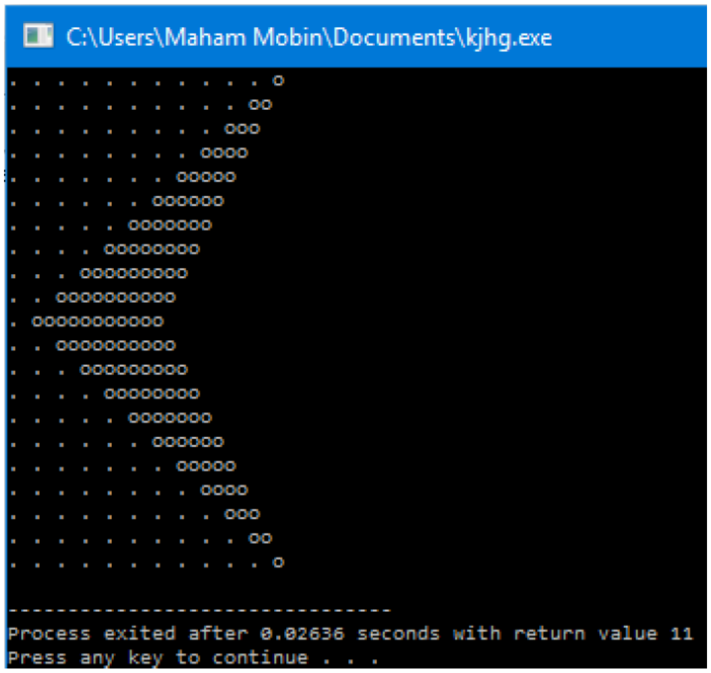
Question 01:

Bilal is shifting from Karachi to Islamabad, he wants to go by Air. He has packed his all stuff in rectangular/cube cardboard cartons. But the targeted airline has restrictions over the volume of carriage cartons., not more than 2000000 cm3.So he decided to find volume of all cartons he packed. Problem arises when he saw that cartons have mentioned only zero, one or two dimensions, he knew that the company he bought cartons from has a minimum dimension of 30 cm for a default volume of 27000 cm3 Cube carton. Help Bilal in finding volumes of all cartons he packed by defining a separate a separate function to calculate volume.

Question 02:

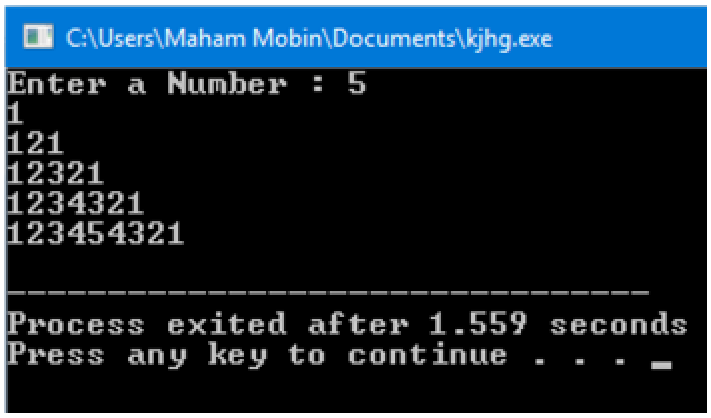
Write a function that takes N as argument and print shapes. Shape A) should have 2N+1 rows as shown below here N=10.

A)



```
. . . . . . . . . . . o
. . . . . . . . . . oo
. . . . . . . . . ooo
. . . . . . . . oooo
. . . . . . . ooooo
. . . . . . oooooo
. . . . . ooooooo
. . . . oooooooo
. . . ooooooooo
. . oooooooooo
. ooooooooooo
. . oooooooooo
. . . ooooooooo
. . . . oooooooo
. . . . . ooooooo
. . . . . . oooooo
. . . . . . . ooooo
. . . . . . . . oooo
. . . . . . . . . ooo
. . . . . . . . . . oo
. . . . . . . . . . . o


--------------------------------
Process exited after 0.02636 seconds with return value 11
Press any key to continue . . .
```

B) Generate the following patterns:



```
Enter a Number : 5
1
121
12321
1234321
123454321

--------------------------------
Process exited after 1.559 seconds
Press any key to continue . . .
```

Question 03:

Hasham is a student of computer science. He is currently self-studying programming in C language. He wants to develop a program for the purpose of reusability, in which he is performing mathematical operations like addition, subtraction, production, and division. Keep in mind, He wants to perform these mathematical operations in different programs since he may need to use any of these operations in any of the developed programs.

Question 04:

Suppose you are a coach of university tennis team. There are seven boys and three girls on the team of ten people. You have to select four people from this group to participate in country championship. Write a function that help you to find in how many ways you can choose the team. (HINT: factorial and combination formulae can be used).

Question 05:

A certain grade of steel is graded according to the following conditions

i. Hardness must be greater than 50
ii. Carbon content must be less than 70
iii. Tensile strength must be greater than 5600

The grades are as follows:

- ✓ Grade is 10 if all three conditions are met.
- ✓ Grade is 9 if conditions (i) and (ii) are met.
- ✓ Grade is 8 if conditions (ii) and (iii) are met.
- ✓ Grade is 7 if conditions (i) and (iii) are met.
- ✓ Grade is 6 if only one condition is met.
- ✓ Grade is 5 if none of the conditions are met.

Design and develop a C function that accepts values of hardness, carbon content and tensile strength of the steel under consideration and returns the grade of the steel. Write a C program that invokes this function and print grade of steel.

Question 06:

Write a program in which user enters his NTS and Intermediate marks and your function will help student in selection of university. Based on these marks Student will be allocated a seat at different department of different university.

University Of Karchi

IT: Above 70% in Fsc. and 70 % in NTS Electronics: Above 70% in Fsc. and 60 % in NTS Telecommunication Above 70% in Fsc. and 50 % in NTS

IQRA University:

IT: 70% - 60 % in Fsc. and 50 % in NTS Chemical: 59% – 50 % in Fsc. and 50 % in NTS Computer: Above 40% and below 50 % in Fsc. and 50 % in NTS

Question 07:

Bob's Discount Bolts charges the following prices:

5 cents per bolt
3 cents per nut
1 cent per washer

Write a program that asks the user for the number of bolts, nuts, and washers in their purchase and then calculates and prints out the total. As an added feature, the program checks the order. A correct order must have at least as many nuts as bolts and at least twice as many washers as blots, otherwise the order has an error. For an error the program writes out "Check the Order: too few nuts" or "Check the Order: too few washers" as appropriate. Both error messages are written if the order has both errors. If there are no errors the program writes out "Order is OK."

```
Number of bolts:            12
Number of nuts:             8
Number of washers:          24

Check the Order:            too few nuts
Total cost:                 108
```