

ST. XAVIER'S COLLEGE

MAITIGHAR, KATHMANDU, NEPAL

Phone: 01-5321365, 01-5344636

Email: ktm@sx.c.edu.np



LAB/WRITTEN ASSIGNMENT NUMBER: 06

"C programming Part I"

Submitted By	Submitted To	Signature
Name : Pranjal Khatri Roll : 929 Class : 11 Section : I	Department Of Computer Science (+2) St.Xavier's College	

Submission Date: 21 November, 2024

Table of Contents

Introduction to C Programming	3
What is C Programming?	3
Why is C Called a Procedural Programming Language?	3
What Are the Features of C Language?	3
What values do the printf() and scanf() functions return ?	4
Sample Code	7
Some Sample outputs.....	10
FlowChart	11

Introduction to C Programming

What is C Programming?

C Is a general-purpose programming language created by Dennis Ritchie in 1972. C is a very popular and widely used programming language, Despite being old this programming language is still use in the fields of computer science. This program was initially developed for the programming of the UNIX Programming language.

The language allows programmers to write low-level code, meaning it can directly interact with the hardware, which is why it is sometimes referred to as a "middle-level language." At the same time, it provides features of high-level languages like functions and libraries, making it versatile and powerful.

Why is C Called a Procedural Programming Language?

A procedural oriented programming language contains a small bunch of code blocks to solve Problem. C is called a Procedural Programming Language because the programming pattern of C was developed to be procedural. These types of programming language help debug because we can easily isolate the error containing procedure.

What Are the Features of C Language?

The major features of C are as:

- Procedural language:
In a procedural language like C step by step, predefined instructions are carried out. C program may contain more than one function to perform a particular task. Most of the commonly used paradigm is an object-oriented programming language.
- Fast
In comparison to newer programming languages, C is faster as these other programming languages has more features and hence due to additional processing, they are slower. C programming language as the middle-level language provides programmers access to direct manipulation with the computer hardware but higher-level languages do not allow this.
- Modularity
The concept of storing C programming language code in the form of libraries for further future uses is known as modularity. This programming language can do very little on its own most of its power is held by its libraries. C language has its own library to solve common problems.
- General-Purpose Language

From system programming to photo editing software, the C programming language is used in various applications. C programming is used in the programming of operating systems like Windows, Linux, android, etc. and many databases like Oracle, MySQL.

What values do the printf() and scanf() functions return ?

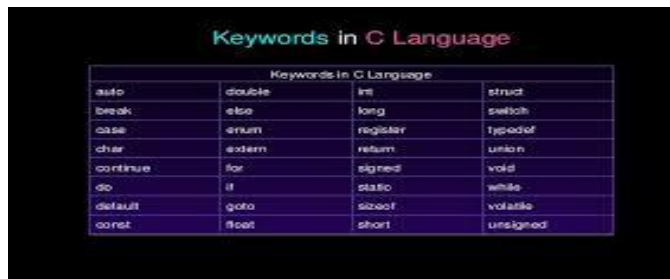
The printf() function returns the number of characters printed, excluding the null terminator, while the scanf() function returns the number of input items successfully read or EOF in case of an error.

Tokens

Tokens are the smallest building blocks of a program. Think of them as the words and symbols in a programming "sentence."

Keywords

Keywords are reserved words in a programming language that have a specific meaning. For example, if, else, for, and while are keywords. You can't use them for anything else, like naming variables.



auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	volatile
const	float	short	unsigned

Variables

Variables are like labels boxes that store information. For eg

```
int age = 20;
```

Constant

Constants are like permanent markers—you can't change their value once they're set.

```
const float PI = 3.14;
```

Data Types

Data types define what kind of data a variable can hold. Examples include:

- int (for integers like 10)
- float (for decimal numbers like 3.14)
- char (for single characters like 'A')
- string (for words like "Hello")

For eg

```
int score;  
float percentage;  
char grade;  
// These are variables and their data types
```

Escape Sequences

Escape sequences are special characters used with a backslash (\) to perform actions like:

\n → New line
\t → Tab space
\\ → Backslash

```
// print to ask user for their marks  
printf( " Enter Your marks (0 to 100 ): \n");  
scanf("%d" , &score);
```

sizeof Function

the size of function tells us what the set variable takes memory

```
printf("%lu", sizeof(int));
```

Algorithm

An algorithm is a step-by-step procedure or set of rules to solve a specific problem or perform a task. It is like a recipe that guides you from the starting point to the end, ensuring that the problem is solved correctly.

Operators and Their Types

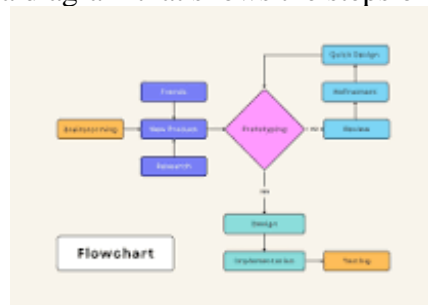
Operators are symbols that perform actions on variables and values. Examples:

- Arithmetic Operators: +, -, *, /, % (e.g., $5 + 3 = 8$)

- Relational Operators: >, <, ==, != (e.g., 5 > 3 is true)
- Logical Operators: &&, ||, ! (e.g., true && false = false)
- Assignment Operators: =, +=, -= (e.g., x += 2 means x = x + 2)
- Bitwise Operators: &, |, ^, <<, >> (e.g., 5 & 3 = 1)

Flowchart

A flowchart is a diagram that shows the steps of a program using shapes



Control Structures

Control structures decide the flow of a program.

1. Sequential: Code runs from top to bottom (default flow).
2. Decision-making: Use if, else, or switch to make choices. For eg

```

89 // Using switch-case for checking passing or failing
90 switch (status) {
91     case 1:
92         printf("You have passed the exam!!\n");
93         break;
94     case 0:
95         printf("You failed the exam. Better luck next time.\n");
96         break;
97     default:
98         printf("Unexpected result.\n");
99         break;
100 }
101
102 if (percentage >= 90) {
103     grade = 'A';
104 } else if (percentage >= 75) {
105     grade = 'B';
106 } else if (percentage >= 50) {
107     grade = 'C';
108 } else if (percentage >= PASS_MARK) {
109     grade = 'D';
110 } else {
111     grade = 'F';
112 }
  
```

Sample Code

```
#include <stdio.h>
// This is header file

/*
Tokens include keywords , identifiers , constants , strings etc
Keywords: int , return , if , for
identifiers : score , percentage
constant : MAX_SCORE , PASS_MARKS
operators : + , * , /
*/

// This is a code to generate a marksheet in exam

#define MAX_SCORE 100
#define PASS_MARKS 40
// This defines the value of MAX_SCORE and PASS_MARKS

int main() {

    float maths, physics, english, chemistry, nepali, computer;
    float total, percentage;
    char grade;
    int status;
    // These are variables and their data types

    // Print to ask user for their marks
    printf("Enter your marks for the following subjects (out of 100):\n");

    printf("Maths: ");
    scanf("%f", &maths); // Use %f for float

    printf("Physics: ");
    scanf("%f", &physics);

    printf("English: ");
    scanf("%f", &english);

    printf("Chemistry: ");
    scanf("%f", &chemistry);

    printf("Nepali: ");
    scanf("%f", &nepali);
```

```

printf("Computer: ");
scanf("%f", &computer);

// Here we check if the input from user is valid
if (maths < 0 || maths > MAX_SCORE || physics < 0 || physics > MAX_SCORE ||
    english < 0 || english > MAX_SCORE || chemistry < 0 || chemistry > MAX_SCORE ||
    nepali < 0 || nepali > MAX_SCORE || computer < 0 || computer > MAX_SCORE) {
    printf("Invalid input! Marks should be between 0 and 100.\n");
    return 1; // Exit the program
}

// This computes the user's total marks and percentage
total = maths + physics + english + chemistry + nepali + computer;
percentage = (total / 600) * 100; // Total marks are 600 (100 per subject)

// Check if the student failed in any subject
if (maths < PASS_MARKS || physics < PASS_MARKS || english < PASS_MARKS ||
    chemistry < PASS_MARKS || nepali < PASS_MARKS || computer < PASS_MARKS)
{
    status = 0; // Failed due to low marks in one or more subjects
} else {
    status = 1; // Passed (if no subject score is below the pass mark)
}

// Figure out the grade based on percentage
if (status == 1) { // Only assign grade if passed
    if (percentage >= 90) {
        grade = 'A';
    } else if (percentage >= 75) {
        grade = 'B';
    } else if (percentage >= 50) {
        grade = 'C';
    } else if (percentage >= PASS_MARKS) {
        grade = 'D';
    }
} else {
    grade = 'F'; // If failed in any subject, the grade is 'F'
}

// Print the grade
printf("Your Grade is: %c\n", grade);

// Print the result
printf("Percentage: %.1f%%\n", percentage);

```



```
// Using switch-case for checking passing or failing
switch (status) {
    case 1:
        printf("You have passed the exam!!\n");
        break;
    case 0:
        printf("You failed the exam. Better luck next time.\n");
        break;
    default:
        printf("Unexpected result.\n");
        break;
}

// Additional information
printf("Program executed by Pranjali Khatri\n");
printf("Roll No: 929\n");

return 0;
}
```

Some Sample outputs

```
Output
^ Enter your marks for the following subjects (out of 100):
  Maths: 50
  Physics: 79
  English: 96
  Chemistry: 87
  Nepali: 60
  Computer: 57
  Your Grade is :C
  Percentage: 71.5%
  You have passed the exam!!
  Program executed by Pranjali Khatri
  Roll No 929

  === Code Execution Successful ===
```

```
Output
^ Enter your marks for the following subjects (out of 100):
  Maths: 12
  Physics: 56
  English: 56
  Chemistry: 56
  Nepali: 56
  Computer: 56
  Your Grade is: F
  Percentage: 48.7%
  You failed the exam. Better luck next time.
  Program executed by Pranjali Khatri
  Roll No: 929

  |
  === Code Execution Successful ===
```

```
Output
^ Enter your marks for the following subjects (out of 100):
  Maths: 300
  Physics: 200
  English: 50
  Chemistry: 80
  Nepali: 86
  Computer: 234
  Invalid input! Marks should be between 0 and 100.

  === Code Exited With Errors ===
```

FlowChart

