

ST. XAVIER'S COLLEGE

MAITIGHAR, KATHMANDU, NEPAL

Phone: 01-5321365, 01-5344636

Email: ktm@sx.edu.np



LAB ASSIGNMENT NUMBER: 06

“WEB TECHNOLOGY I AND C PROGRAMMING”

Submitted By	Submitted To	Signature
Name: Shreyashkar Shah Roll No: 1025 Class: 11 Section: J	Mr. Jaya Sundar Shilpakar Department of Computer Science, St. Xavier's College	

Submission Date: 3rd February, 2025

C PROGRAMMING SECTION

What is tokens (data types, reserved words, functions) in C Programming Language?

In C programming, tokens are the basic building blocks of the language. These are the smallest units in the source code that have meaningful representation. Tokens help the compiler understand the structure of the program. In C Programming language, there are several types of tokens:

Data types in C define the type of data that a variable can hold. They are fundamental for specifying the kind of operations that can be performed on the data. C has both primitive and derived data types.

Primitive Data Types	Derived Data Types
int: Used for integers (whole numbers)	Arrays
float: Used for single-precision floating-point numbers.	Pointers
double: Used for double-precision floating-point numbers.	Structures
char: Used for characters (usually a single byte).	Unions
void: Used for functions that don't return a value or indicate the absence of a data type.	

Reserved words (or keywords) are predefined words in C that have special meanings and cannot be used for variable names, function names, etc. These are part of the C syntax and are essential for defining the structure and flow of a C program. Some examples of reserved words in C include: `int`, `char`, `return`, `if`, `else`, `for`, `while`, `switch`, `struct`, `union`, `break`, `continue`, `void`, etc.

Functions in C are blocks of code that perform a specific task. In the context of tokens, a **function** is typically invoked by its name followed by parentheses and a block of code. Functions may also take arguments and return a value.

Some common standard library functions in C are:

- `printf()`: Prints output to the console.
- `scanf()`: Reads input from the console.
- `malloc()`: Allocates dynamic memory.
- `free()`: Frees dynamically allocated memory.

What is keywords in C Programming Language?

In C programming, **keywords** are reserved words that have a special meaning and are part of the syntax of the language. These words are predefined in the C language and cannot be used for anything other than their intended purpose. They are the building blocks that help define the structure and flow of a C program.

Each keyword in C serves a specific purpose, such as controlling the flow of execution, defining data types, or managing program functions.

Some of the most commonly used keywords in C include:

- **Data types:**
 - `int`: Defines integer type.
 - `char`: Defines character type.
 - `float`: Defines floating-point type.
 - `double`: Defines double-precision floating-point type.
 - `void`: Specifies no type or function returns nothing.
- **Control flow:**
 - `if`: Conditional statement.
 - `else`: Alternative path for `if` condition.
 - `switch`: Used to select one of many code blocks to execute.
 - `case`: Defines a branch in a `switch` statement.
 - `default`: The default case in a `switch` statement.
 - `break`: Exits a loop or `switch` statement.
 - `continue`: Skips the current iteration of a loop.
- **Function-related:**
 - `return`: Returns a value from a function.
 - `void`: Indicates a function does not return a value.
- **Looping:**
 - `for`: A loop that repeats a block of code a specific number of times.
 - `while`: A loop that repeats as long as a condition is true.
- **Structure-related:**
 - `struct`: Defines a structure (a custom data type).
 - `union`: Defines a union (a data structure where members share the same memory location).

What are the operators used in C Programming Language?

In C programming, operators are symbols that are used to perform operations on variables and values. Operators are essential for performing calculations, comparisons, and manipulating data in the program. C has a rich set of operators, which can be categorized into several types based on their functionality.

Arithmetic operators: These operators are used to perform mathematical calculations.

- `+` : Addition (e.g., `a + b`)
- `-` : Subtraction (e.g., `a - b`)
- `*` : Multiplication (e.g., `a * b`)
- `/` : Division (e.g., `a / b`)
- `%` : Modulo (remainder of division) (e.g., `a % b`)

In C programming, **operators** are symbols that are used to perform operations on variables and values. Operators are essential for performing calculations, comparisons, and manipulating data in the program. C has a rich set of operators, which can be categorized into several types based on their functionality.

Arithmetic Operators: These operators are used to perform mathematical calculations.

- `+` : Addition (e.g., `a + b`)

- - : Subtraction (e.g., $a - b$)
- * : Multiplication (e.g., $a * b$)
- / : Division (e.g., a / b)
- % : Modulo (remainder of division) (e.g., $a \% b$)

Relational (Comparison) Operators: These operators are used to compare two values and return a boolean result (either true or false).

- == : Equal to (e.g., $a == b$)
- != : Not equal to (e.g., $a != b$)
- > : Greater than (e.g., $a > b$)
- < : Less than (e.g., $a < b$)
- >= : Greater than or equal to (e.g., $a >= b$)
- <= : Less than or equal to (e.g., $a <= b$)

Logical Operators: These operators are used to perform logical operations, usually with boolean values.

- && : Logical AND (e.g., $a \&\& b$ returns true if both a and b are true)
- || : Logical OR (e.g., $a || b$ returns true if at least one of a or b is true)
- ! : Logical NOT (e.g., $!a$ returns true if a is false)

Assignment Operators: These operators are used to assign values to variables.

- = : Simple assignment (e.g., $a = b$)
- += : Add and assign (e.g., $a += b$ is equivalent to $a = a + b$)
- -= : Subtract and assign (e.g., $a -= b$ is equivalent to $a = a - b$)
- *= : Multiply and assign (e.g., $a *= b$ is equivalent to $a = a * b$)
- /= : Divide and assign (e.g., $a /= b$ is equivalent to $a = a / b$)
- %= : Modulo and assign (e.g., $a \% = b$ is equivalent to $a = a \% b$)

Increment and Decrement Operators: These operators are used to increase or decrease the value of a variable by 1.

- ++ : Increment (e.g., $a++$ or $++a$)
- -- : Decrement (e.g., $a--$ or $--a$)

IMPLEMENTATIONS

1. Header Files and Function Prototypes

```

#include <stdio.h>
#include <stdlib.h>

// Function prototype
typedef struct {
    int id;
    char name[50];
} Student;

// Function declaration
void displayMessage();
int add(int a, int b);

```

2. Main Function & Primitive Data Types

```

int main() {
    // Primitive Data Types
    int num = 10;
    float price = 12.5;
    double pi = 3.141592;
    char letter = 'A';
    void *ptrVoid;
}

```

3. Derived Data Types

```

// Derived Data Types
int arr[] = {1, 2, 3, 4, 5};
int *ptr = &num; // Pointer
Student student1 = {1, "John Doe"}; // Structure
union {
    int intValue;
    float floatValue;
} myUnion;
myUnion.intValue = 10;

```

4. Control Flow (If-Else & Switch Statement)

```
// Control Flow
if (num > 5) {
    printf("Number is greater than 5\n");
} else {
    printf("Number is less than or equal to 5\n");
}

switch (num) {
    case 10:
        printf("Number is 10\n");
        break;
    default:
        printf("Number is not 10\n");
}
```

5. Looping (For & While Loop)

```
// Looping
for (int i = 0; i < 5; i++) {
    printf("Array element %d: %d\n", i, arr[i]);
}

int count = 0;
while (count < 3) {
    printf("While loop count: %d\n", count);
    count++;
}
```

6. Function Calls

```
// Function usage
displayMessage();
printf("Sum of 5 and 3: %d\n", add(5, 3));
```

7. Arithmetic Operators

```
// Arithmetic Operators
int a = 10, b = 5;
printf("Addition: %d\n", a + b);
printf("Subtraction: %d\n", a - b);
printf("Multiplication: %d\n", a * b);
printf("Division: %d\n", a / b);
printf("Modulo: %d\n", a % b);
```

8. Relational Operators

```
// Relational Operators
printf("Equal: %d\n", a == b);
printf("Not Equal: %d\n", a != b);
printf("Greater Than: %d\n", a > b);
printf("Less Than: %d\n", a < b);
printf("Greater Than or Equal: %d\n", a >= b);
printf("Less Than or Equal: %d\n", a <= b);
```

9. Logical Operators

```
// Logical Operators
printf("Logical AND: %d\n", (a > 5 && b > 2));
printf("Logical OR: %d\n", (a > 5 || b > 10));
printf("Logical NOT: %d\n", !(a > b));
```

10. Assignment Operators

```
// Assignment Operators
a += 5;
printf("After += : %d\n", a);
a -= 2;
printf("After -= : %d\n", a);
a *= 2;
printf("After *= : %d\n", a);
a /= 2;
printf("After /= : %d\n", a);
a %= 3;
printf("After %= : %d\n", a);
```

11. Increment and Decrement Operators

```
// Increment and Decrement Operators
printf("Pre-increment: %d\n", ++a);
printf("Post-increment: %d\n", a++);
printf("After Post-increment: %d\n", a);
printf("Pre-decrement: %d\n", --a);
printf("Post-decrement: %d\n", a--);
printf("After Post-decrement: %d\n", a);
```

12. Dynamic Memory Allocation

```
// Dynamic Memory Allocation
int *dynamicVar = (int *)malloc(sizeof(int));
*dynamicVar = 100;
printf("Dynamically allocated value: %d\n", *dynamicVar);
free(dynamicVar);

return 0;
}
```

13. Function Definitions

```
// Function definitions
void displayMessage() {
    printf("Hello from function!\n");
}

int add(int a, int b) {
    return a + b;
}
```

WEB PROGRAMMING LANGUAGE SECTION

What is tags in HTML? (Paired tags, Single tags)

In HTML (Hyper Text Markup Language), tags are used to define elements and structure web pages. Tags are enclosed in angle brackets (< >) and usually come in pairs, but some can be single (self-closing).

Paired Tags: These tags have an opening tag and a closing tag. The closing tag is written with a forward slash (/). E.g. <p>...</p>, ..., <h1>...</h1>, etc.

Single Tags: These tags do not have a closing tag. Instead, they are self-contained and often used for elements that do not need wrapping content. E.g. ,
, <hr>, <input>, etc.

What is attributes and values in CSS?

In CSS (Cascading Style Sheets), attributes are called properties, and they define how HTML elements should be styled. The values are assigned to these properties to specify how they should appear.

CSS Attributes: CSS properties define what aspect of an element you want to style, such as color, size, font, or layout.

- `color` → Sets the text color.
- `background-color` → Defines the background color.
- `font-size` → Specifies the text size.
- `width` and `height` → Control the dimensions of an element.
- `border` → Sets a border around an element.
- `margin` and `padding` → Control spacing around and inside elements.

CSS Values: Each property is assigned a value, which determines how the style is applied.

- `color: red;` → The text color is set to red.
- `font-size: 20px;` → The font size is 20 pixels.
- `background-color: yellow;` → The background color is yellow.

What are the operators used in Web Programming Language?

In web programming, different languages like HTML, CSS, JavaScript, PHP, Python, and SQL use various operators for calculations, comparisons, and logic.

JavaScript is widely used in web development, and it supports several types of operators:

Arithmetic Operators:

- `+` (Addition) → `a + b`
- `-` (Subtraction) → `a - b`
- `*` (Multiplication) → `a * b`
- `/` (Division) → `a / b`
- `%` (Modulus) → `a % b` (Remainder)
- `++` (Increment) → `a++`
- `--` (Decrement) → `b--`

Assignment Operators:

- `=` (Assign) → `x = 5`
- `+=` (Add & Assign) → `x += 3`
- `-=` (Subtract & Assign) → `x -= 2`
- `*=` (Multiply & Assign) → `x *= 2`
- `/=` (Divide & Assign) → `x /= 2`
- `==` (Equal to) → `x == y`

Comparison Operators:

- `===` (Strict Equal) $\rightarrow x === 10$
- `!=` (Not Equal) $\rightarrow x != y$
- `!==` (Strict Not Equal) $\rightarrow x !== '10'$
- `>` (Greater than) $\rightarrow x > y$
- `<` (Less than) $\rightarrow x < y$
- `>=` (Greater than or Equal) $\rightarrow x >= 10$
- `<=` (Less than or Equal) $\rightarrow y <= 5$

IMPLEMENTATIONS

HTML Output


Welcome to My Website

Your go-to place for amazing content!

- [Home](#)
- [About](#)
- [Services](#)
- [Contact](#)

Home

This is the home section of the website. Here you can find the latest updates and news.

Placeholder Image

About Us

We are a team of passionate individuals dedicated to providing the best content for our users.

- Experienced Professionals
- Customer-Centric Approach
- Innovative Solutions

Our Services

We offer a wide range of services to meet your needs:

1. Web Development
2. Graphic Design
3. SEO Optimization

Contact Us

Name:

Email:

Message:

Submit

Latest News

Check out our new blog post on web development trends!

[Read More](#)

© 2023 My Website. All rights reserved.