Programming with C

Introduction of Basic in C

E-Gov Cell

Contents

Variables, Data types Input/Output

Operators in C

Conditional Statements

Loop Control Statements

Function & Recursion

Pointers

String

Q&A (Doubt Solving)

Variables

Variable is the name of a memory location which stores some data.

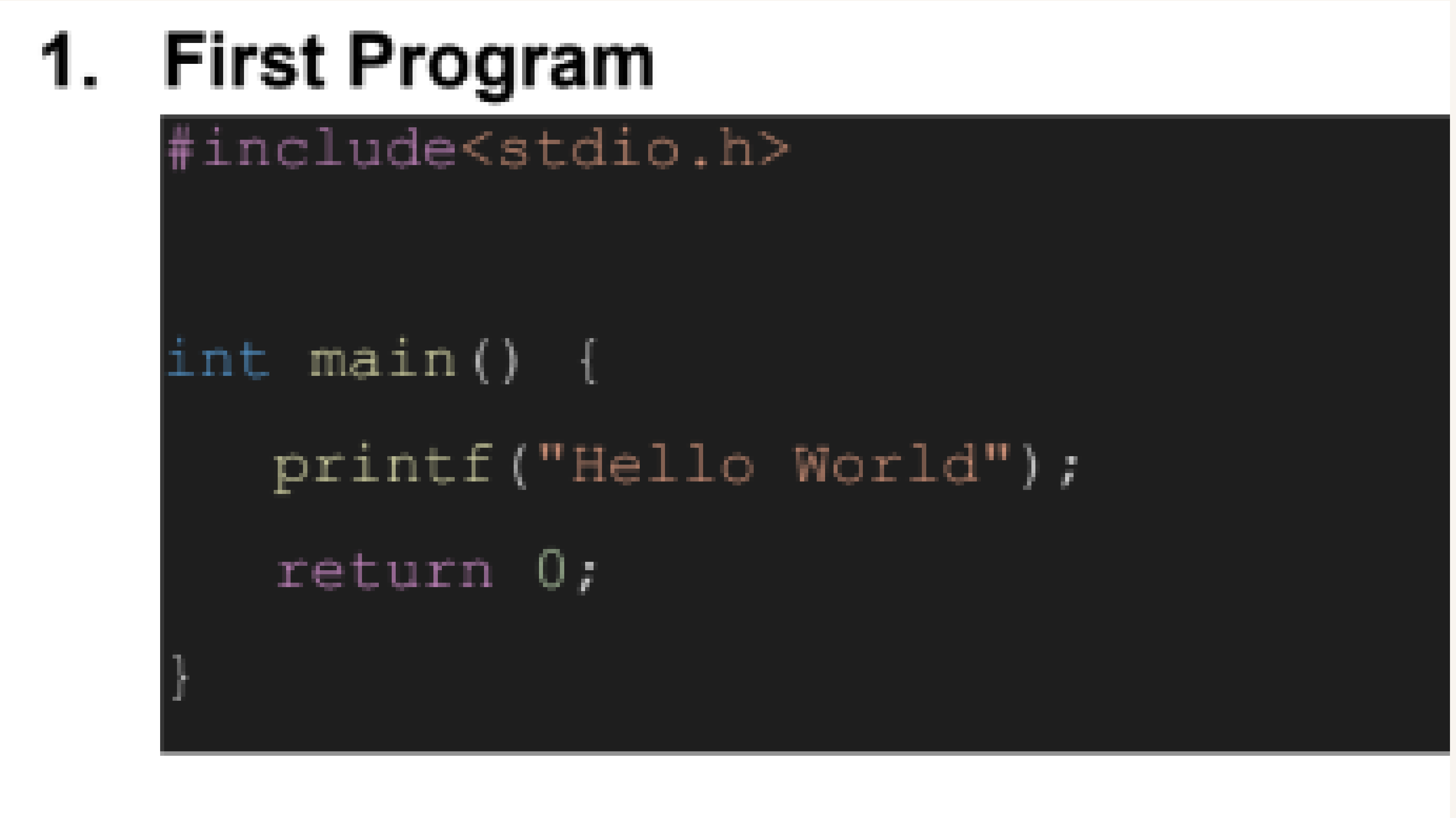
VARIABLES RULES -

a. Variables are case sensitive

b. 1st character is alphabet or '_'

c. no comma/blank space

d. No other symbol other than '_'

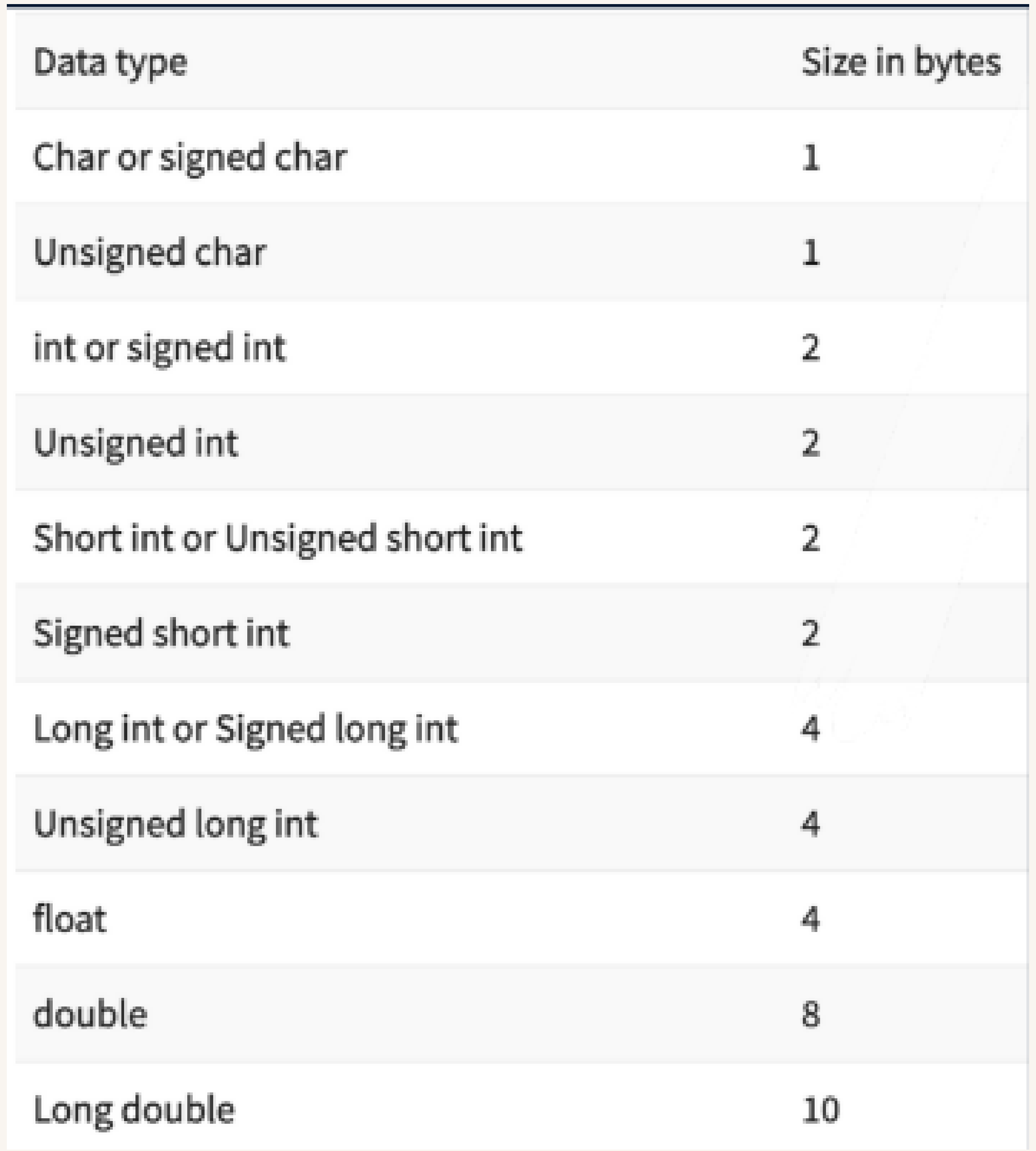


1. First Program

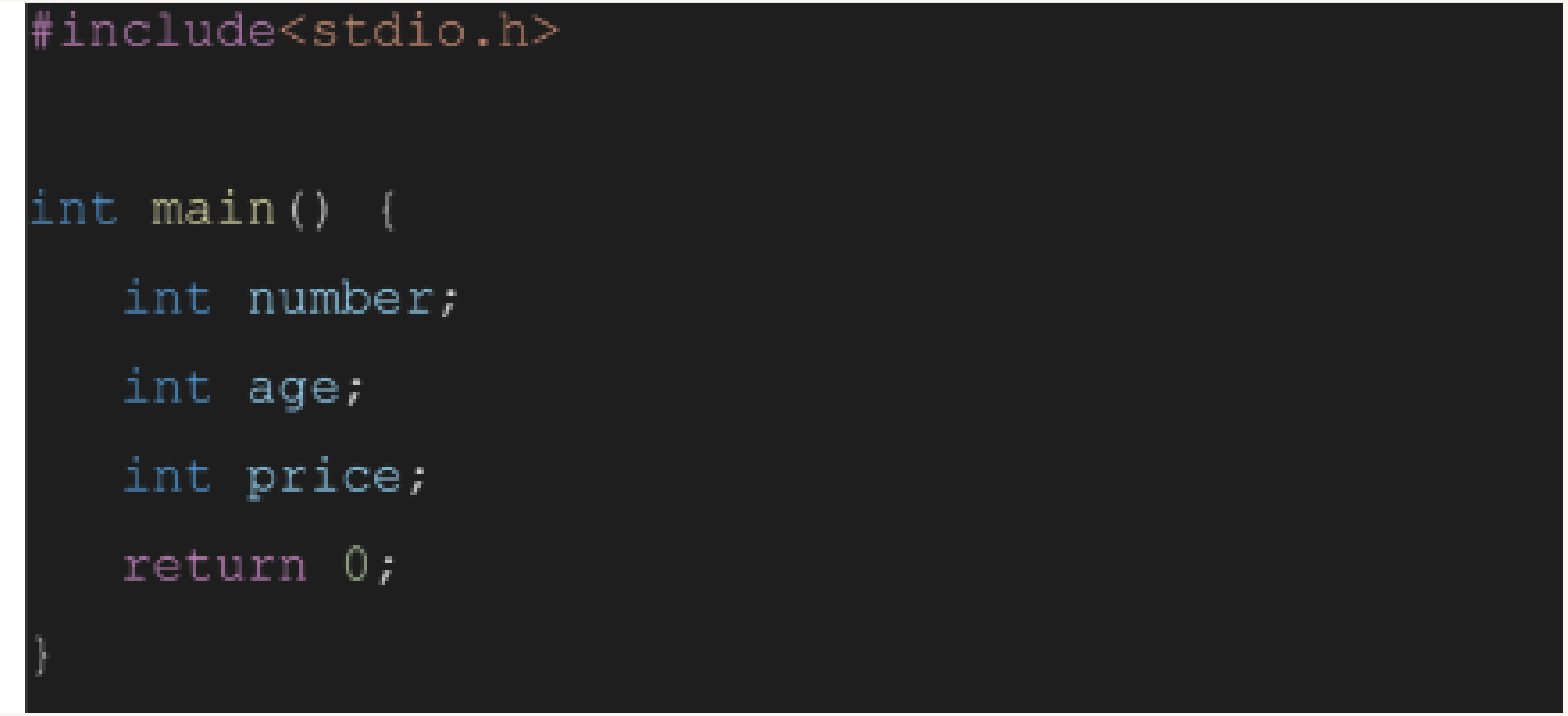
```
#include<stdio.h>

int main() {
   printf("Hello World");
   return 0;
}
```

Data Types

It specifies the type of data that the variable can store like integer, character, floating, double, etc.



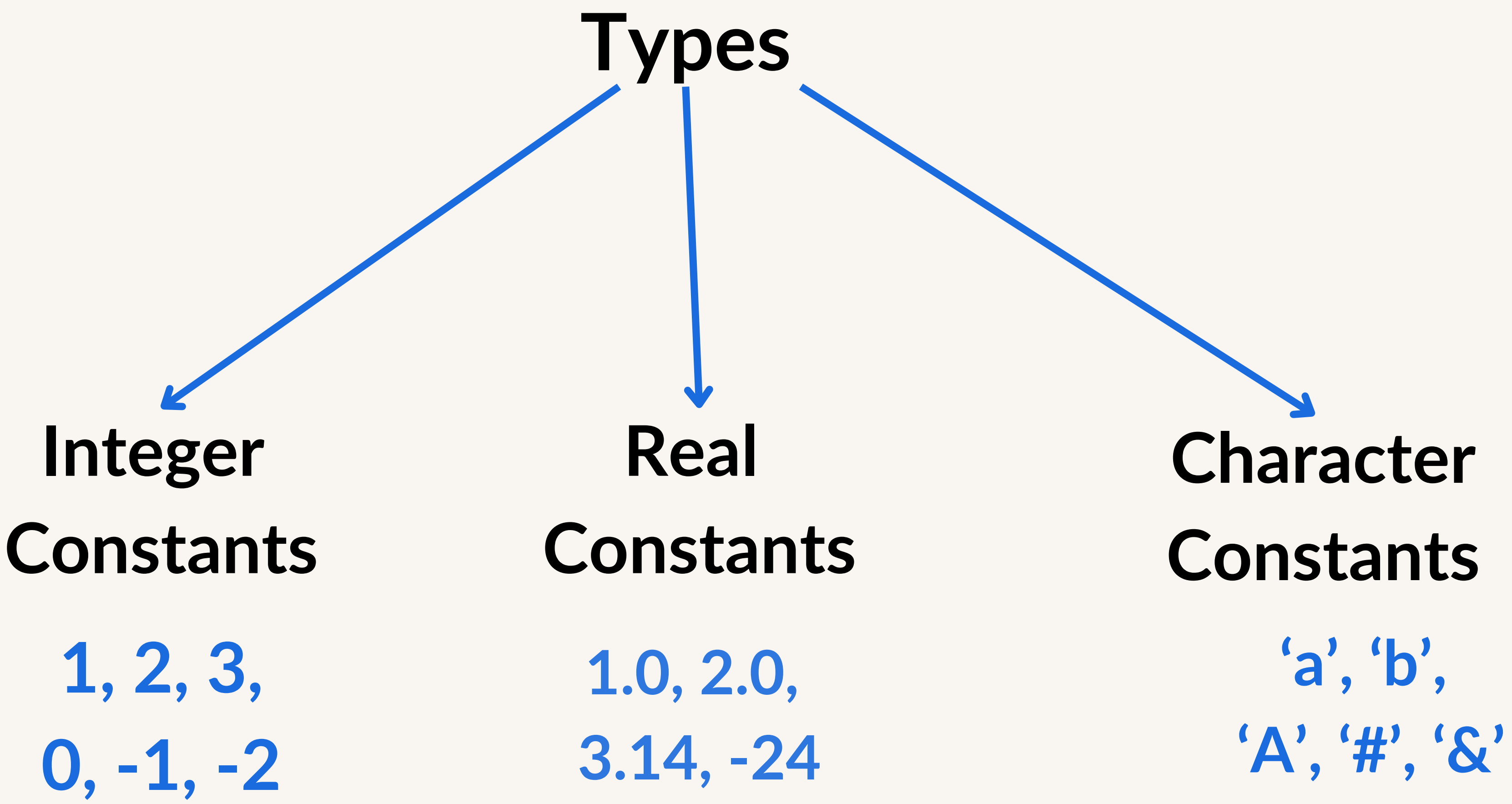
Data type	Size in bytes
Char or signed char	1
Unsigned char	1
int or signed int	2
Unsigned int	2
Short int or Unsigned short int	2
Signed short int	2
Long int or Signed long int	4
Unsigned long int	4
float	4
double	8
Long double	10



```
#include<stdio.h>

int main() {
   int number;
   int age;
   int price;
   return 0;
}
```

Constants

Vlaues that don’t change(fixed)



Types

Integer Constants

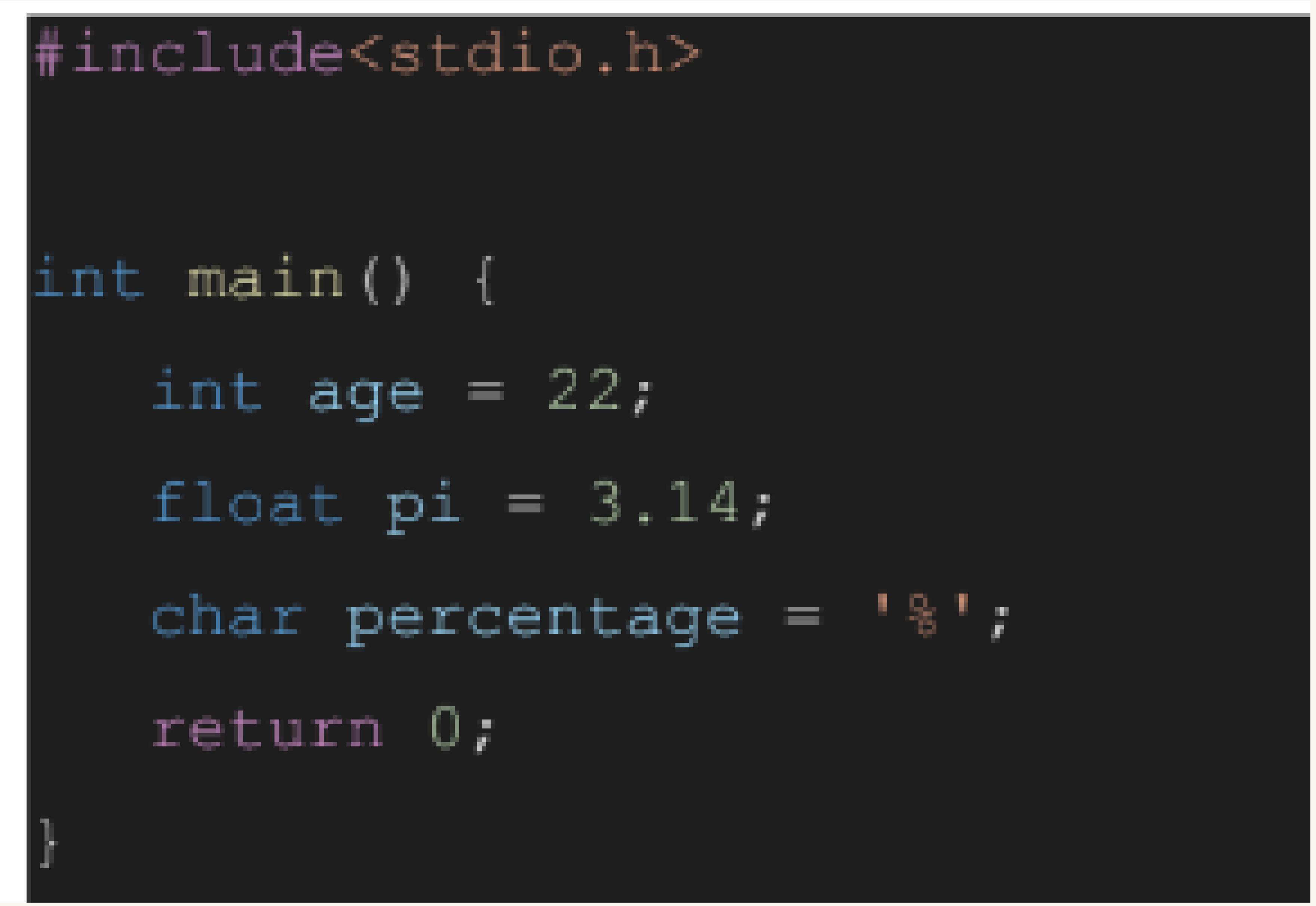
1, 2, 3,
0, -1, -2

Real Constants

1.0, 2.0,
3.14, -24

Character Constants

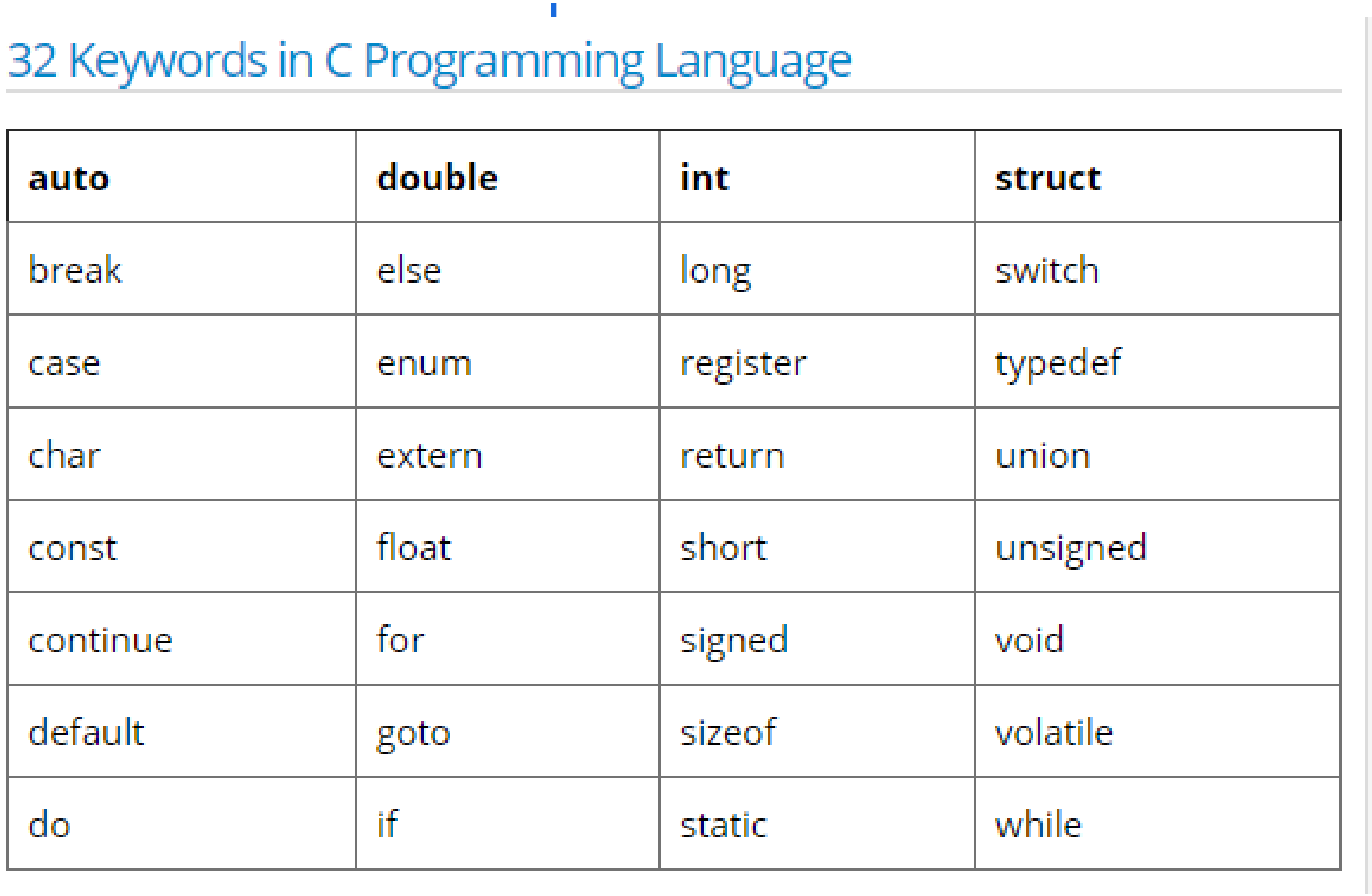
‘a’, ‘b’,
‘A’, ‘#’, ‘&’



```
#include<stdio.h>


int main() {
    int age = 22;
    float pi = 3.14;
    char percentage = '%';
    return 0;
}
```

Keywords

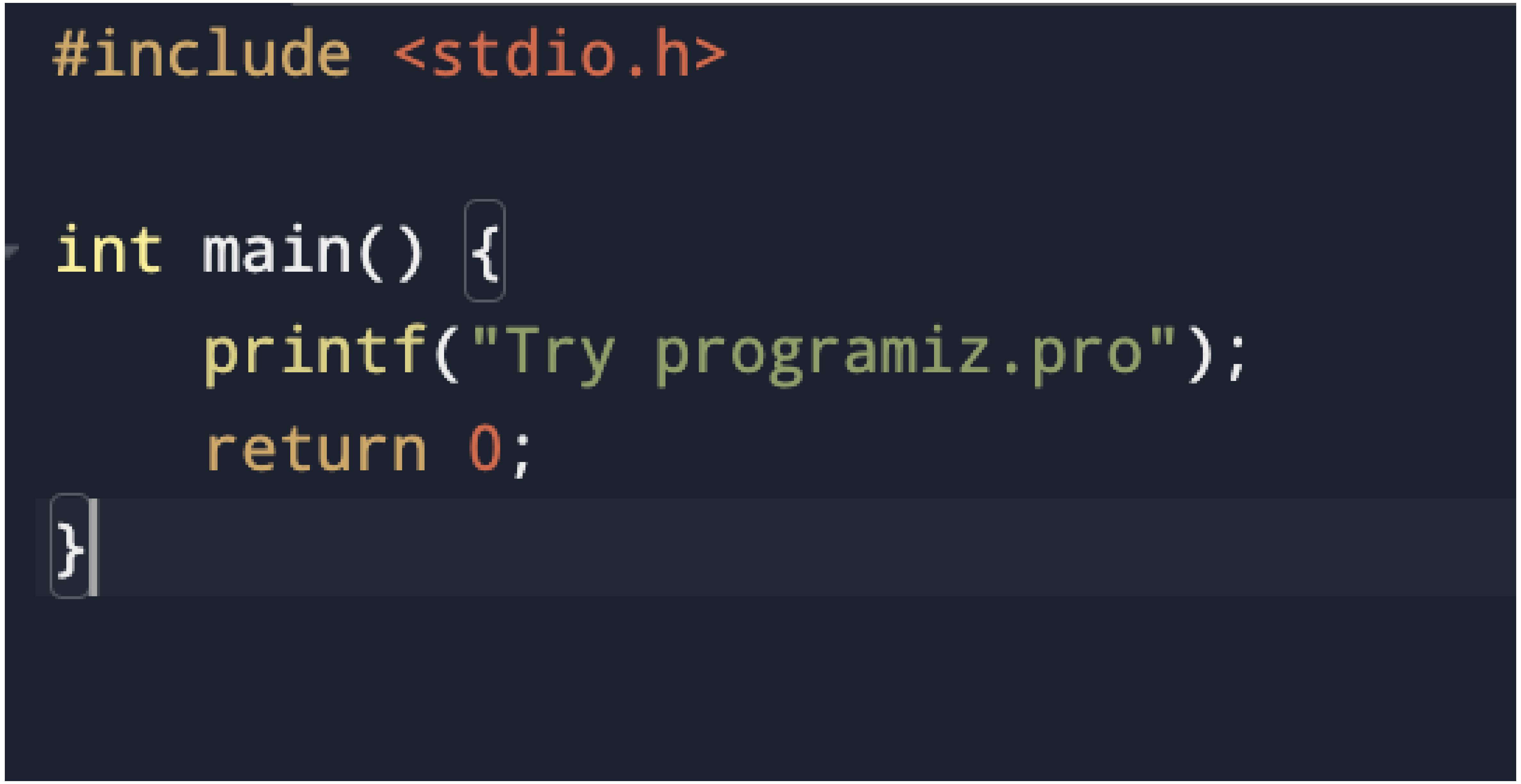
Reserved words that have special meaning to the compiler



32 Keywords in C Programming Language

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

Program Structure

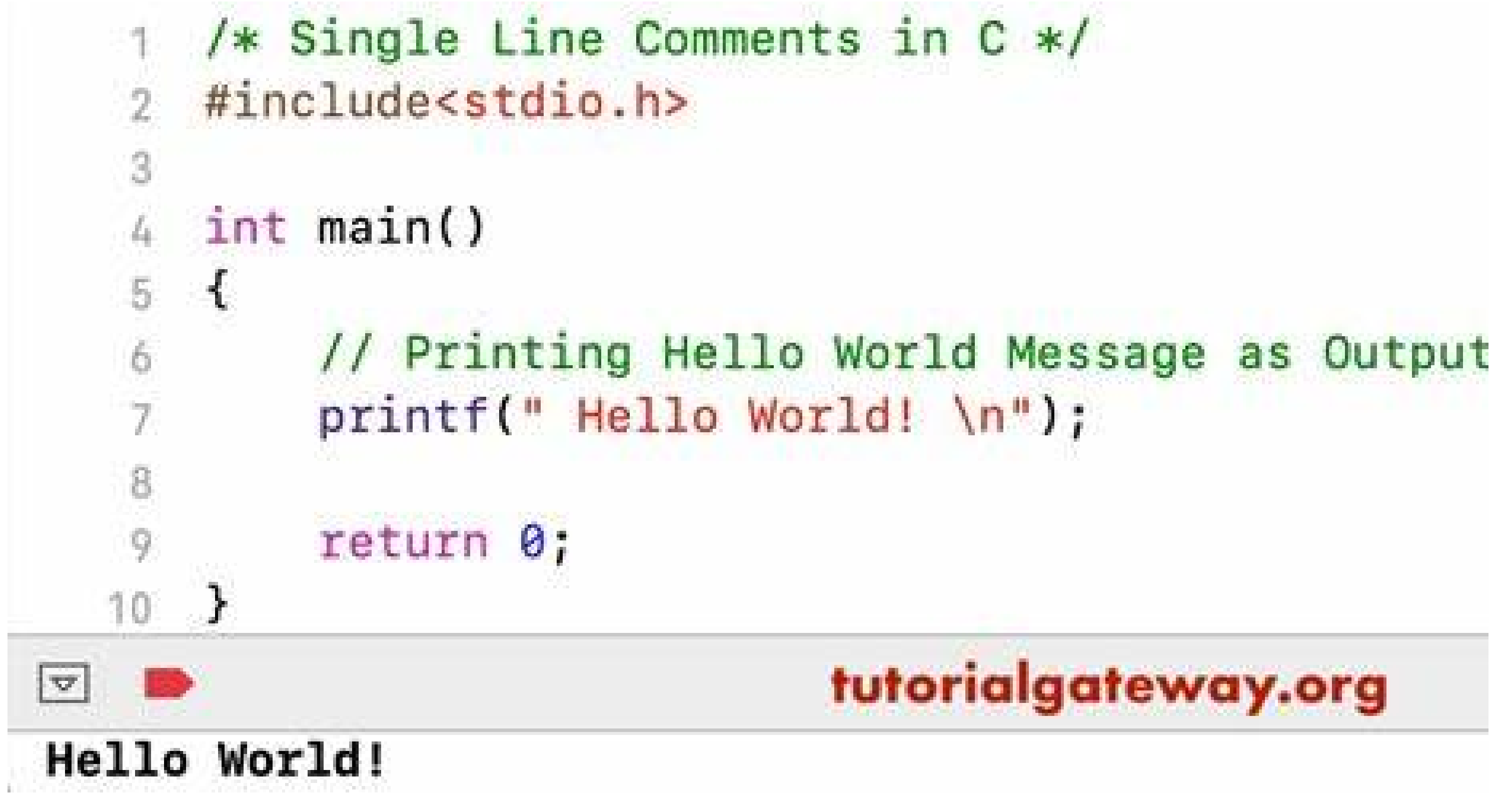


```c
#include <stdio.h>

int main() {
    printf("Try programiz.pro");
    return 0;
}
```

Comments

Comments are lines or blocks of text used to document a program's functionality and explain how a program works for the benefit of a programmer

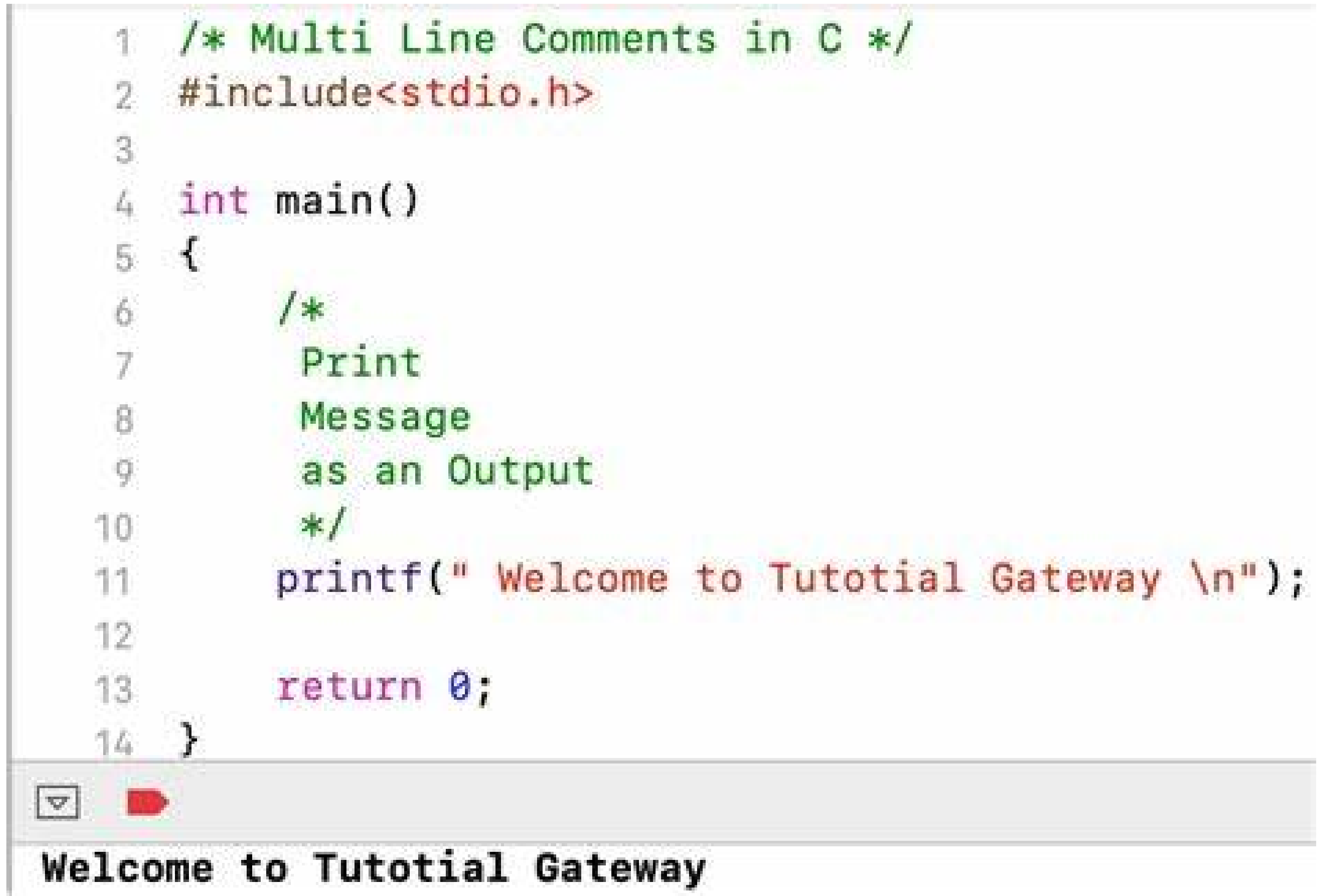
Single Line Comments :-



```
/* Single Line Comments in C */
#include<stdio.h>

int main()
{
     // Printing Hello World Message as Output
     printf(" Hello World! \n");

     return 0;
}
```

```
Hello World!
```

Multi Line Comments :-



```
/* Multi Line Comments in C */
#include<stdio.h>

int main()
{
     /*
      Print
      Message
      as an Output
      */
     printf(" Welcome to Tutotial Gateway \n");

     return 0;
}
```

```
Welcome to Tutotial Gateway
```

Format Specifiers

Cases

1. **Integers**

```
printf(“age is %d”, age);
```

1. **Real Numbers**

```
printf(“value of pi is  %f”, pi);
```

1. **Characters**

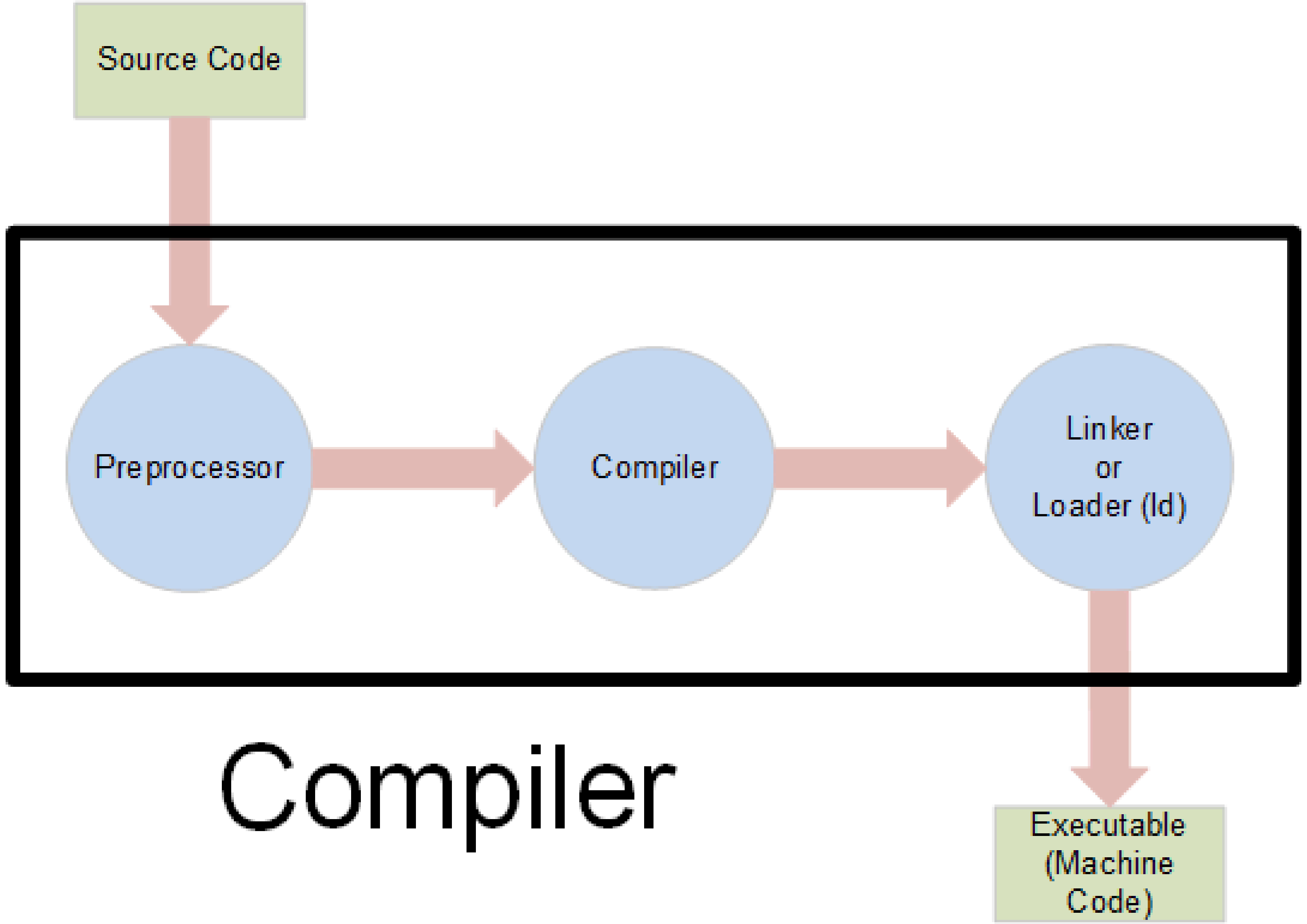
```
printf(“ Dollar looks like this  %c”, dollar);
```

1. **Input**

```
scanf(“%d”, &age);
```

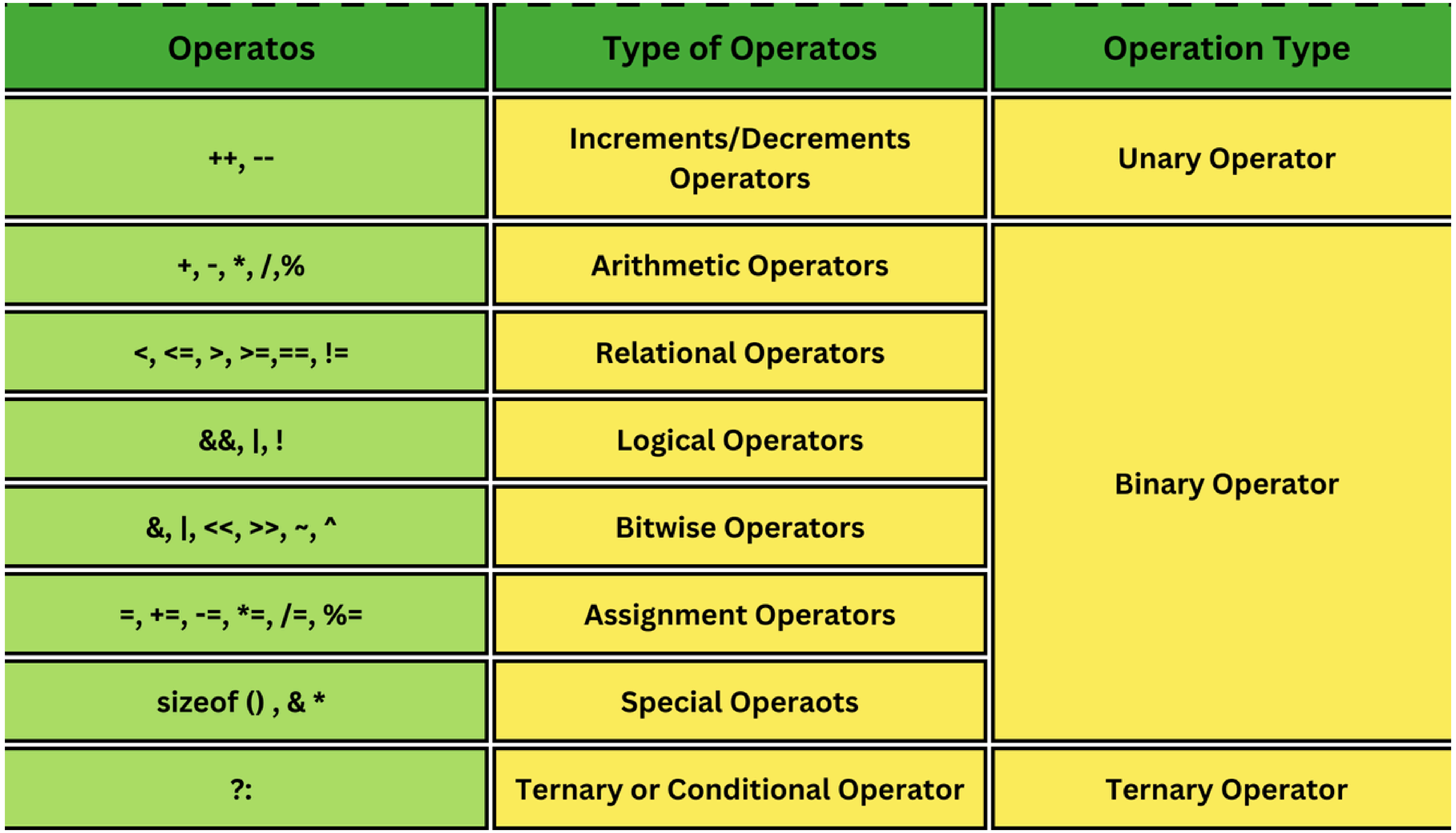
Compilation

The compilation is the process of transforming source code into object code. It is accomplished with the help of the compiler



Operators in C

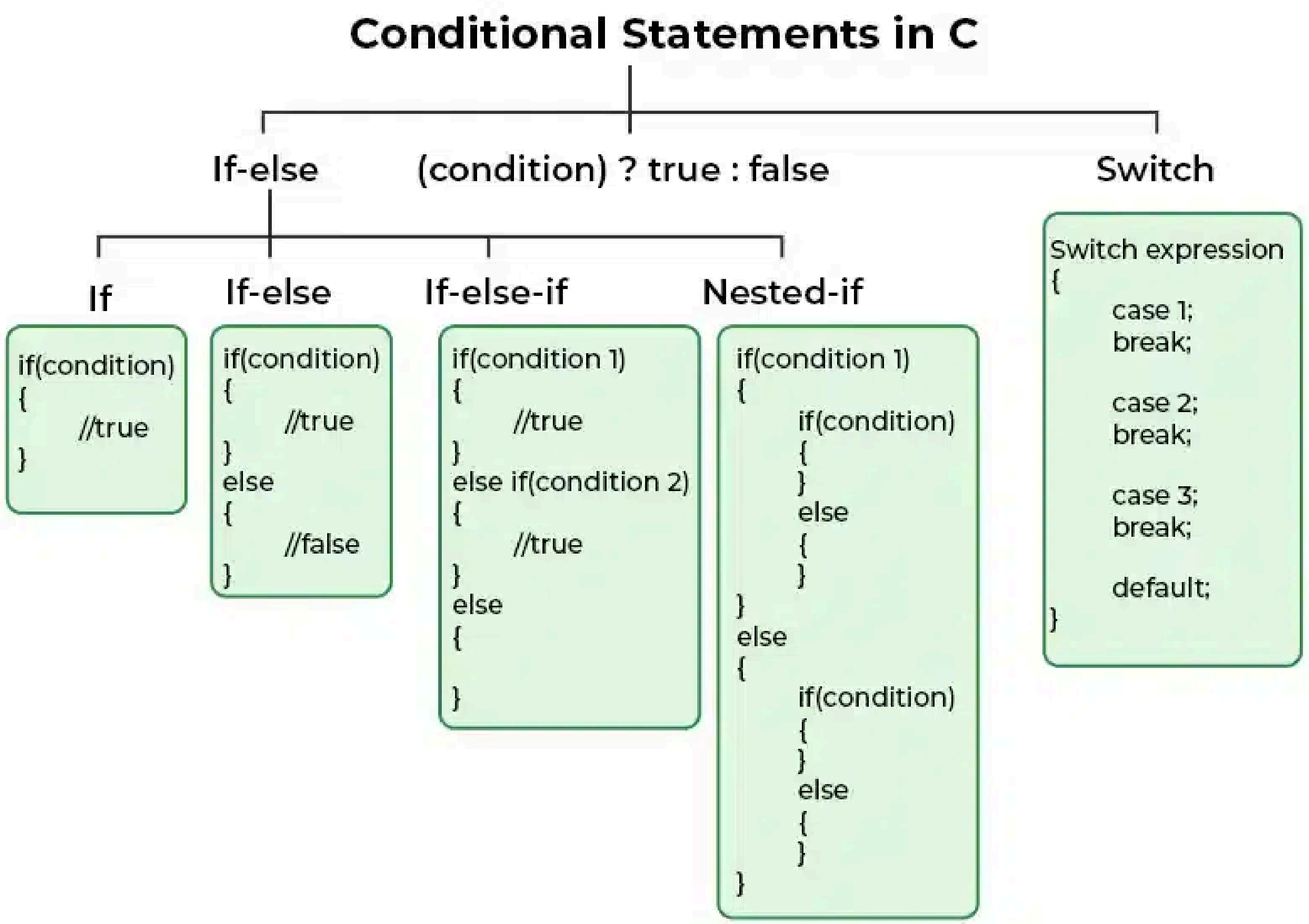
Operators are symbols that represent operations to be performed on one or more operands.



Operatos	Type of Operatos	Operation Type
++, --	Increments/Decrements Operators	Unary Operator
+, -, *, /,%	Arithmetic Operators	Binary Operator
<, <=, >, >=,==, !=	Relational Operators	
&&, |, !	Logical Operators	
&, |, <<, >>, ~, ^	Bitwise Operators	
=, +=, -=, *=, /=, %=	Assignment Operators	
sizeof () , & *	Special Operaots	
?:	Ternary or Conditional Operator	Ternary Operator

Conditional Statement

The conditional statements (also known as decision control structures) such as if, if else, switch, etc. are used for decision-making purposes in C programs.



Conditional Statements in C

```
if(condition)
{
        //true
}
```

```
if(condition)
{
        //true
}
else
{
        //false
}
```

```
if(condition 1)
{
        //true
}
else if(condition 2)
{
        //true
}
else
{

}
```

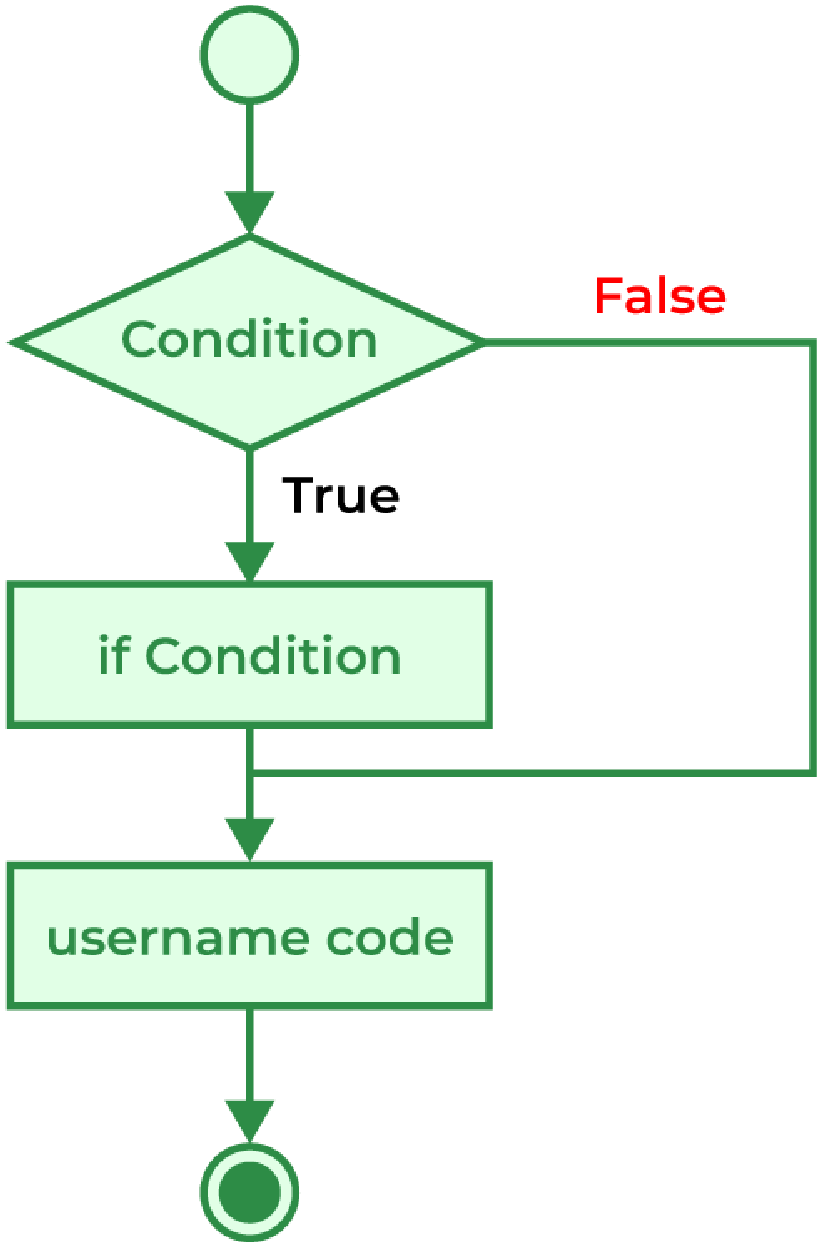
```
if(condition 1)
{
        if(condition)
        {
        }
        else
        {
        }
}
else
{
        if(condition)
        {
        }
        else
        {
        }
}
```

```
Switch expression
{
        case 1;
        break;

        case 2;
        break;

        case 3;
        break;

        default;
}
```

If Statement

The if statement in C is a control structure that evaluates a condition.

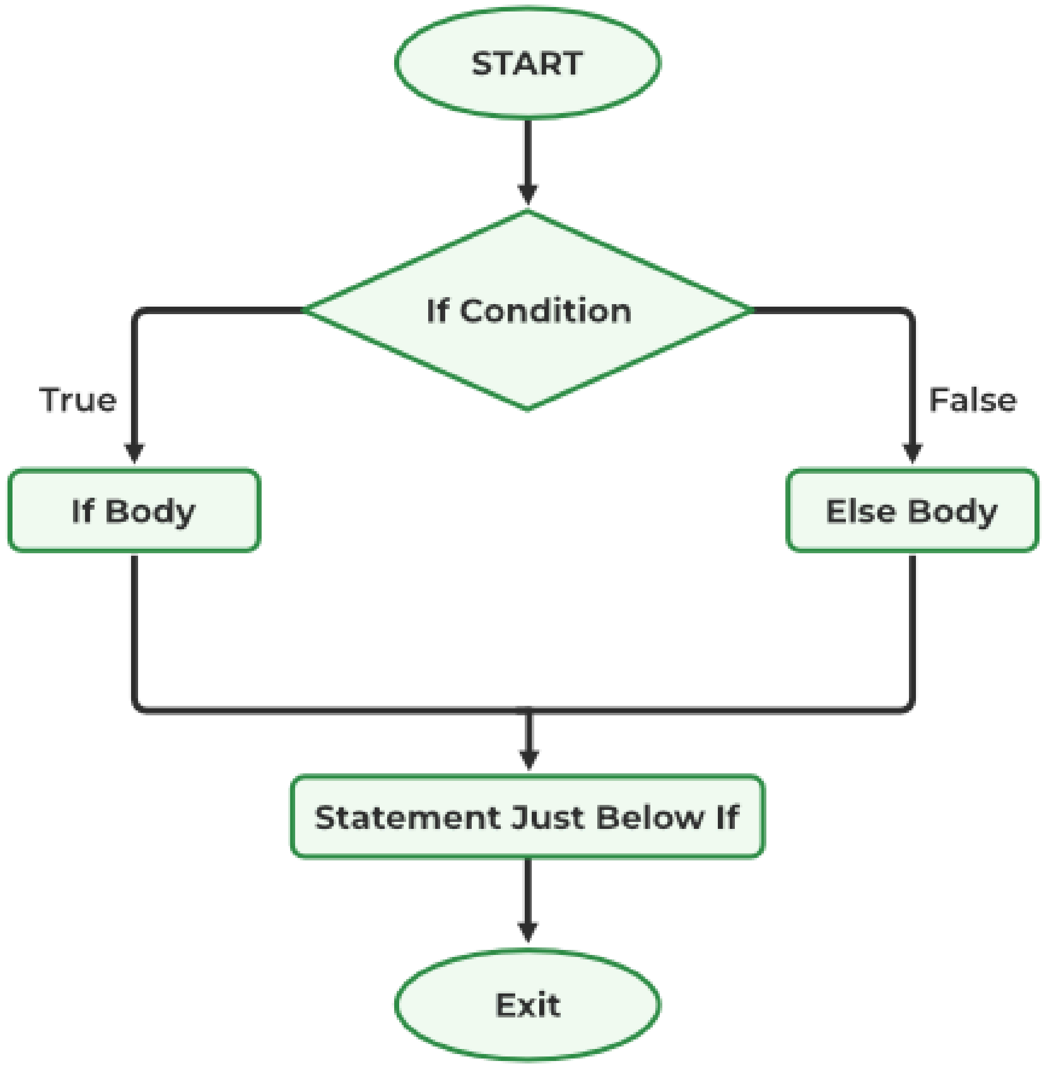
```
// C program to illustrate If statement
#include <stdio.h>

int main()
{
    int i = 10;

    if (i > 15) {
        printf("10 is greater than 15");
    }

    printf("I am Not in if");
return 0; }
```



If else Statement

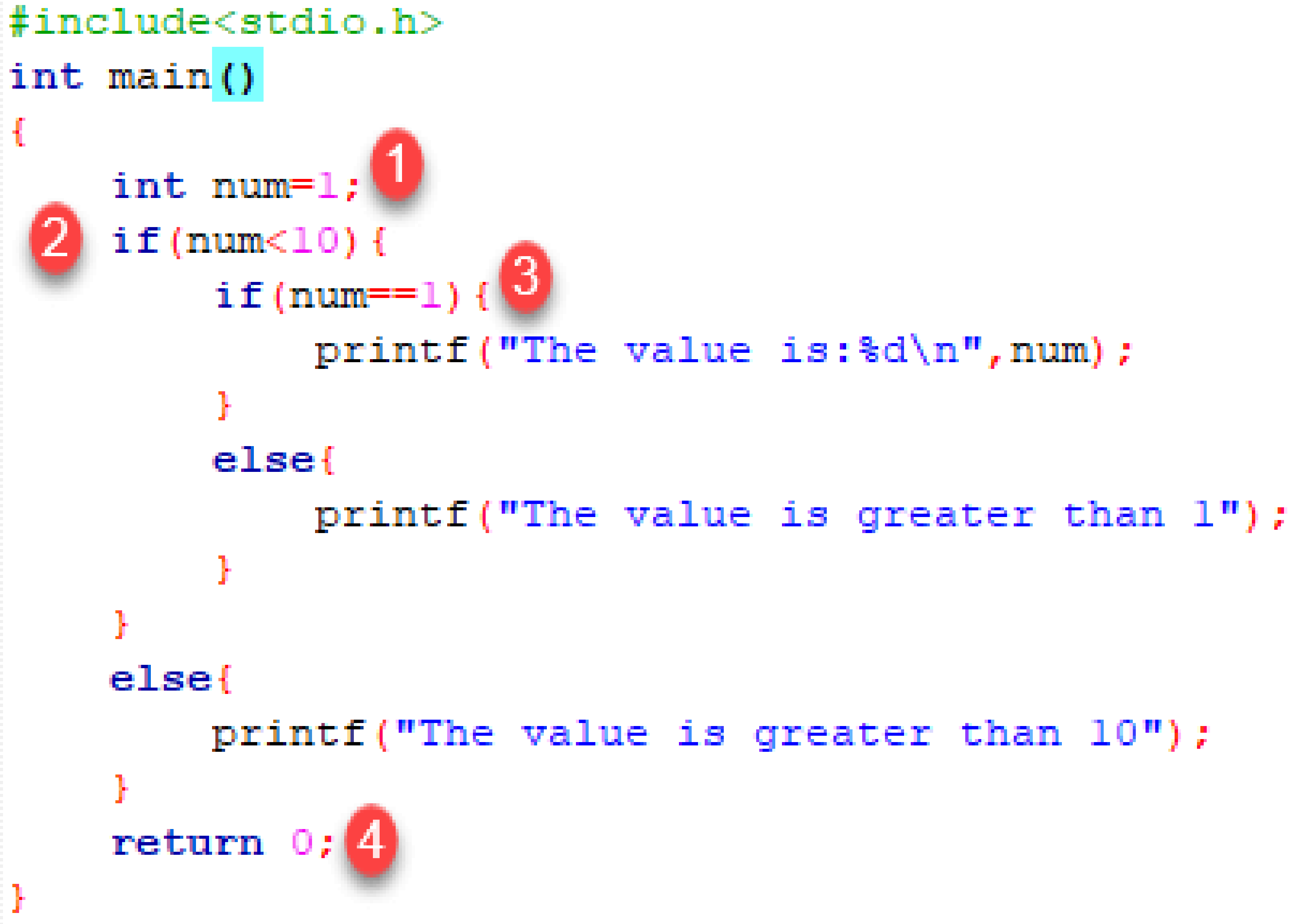
The if-else statement enables the user to execute different statements based on different conditions

```
// C program to illustrate If statement
#include <stdio.h>

int main()
{
   int i = 20;

   if (i < 15) {

       printf("i is smaller than 15");
   }
   else {

       printf("i is greater than 15");
   }
   return 0;
}
```

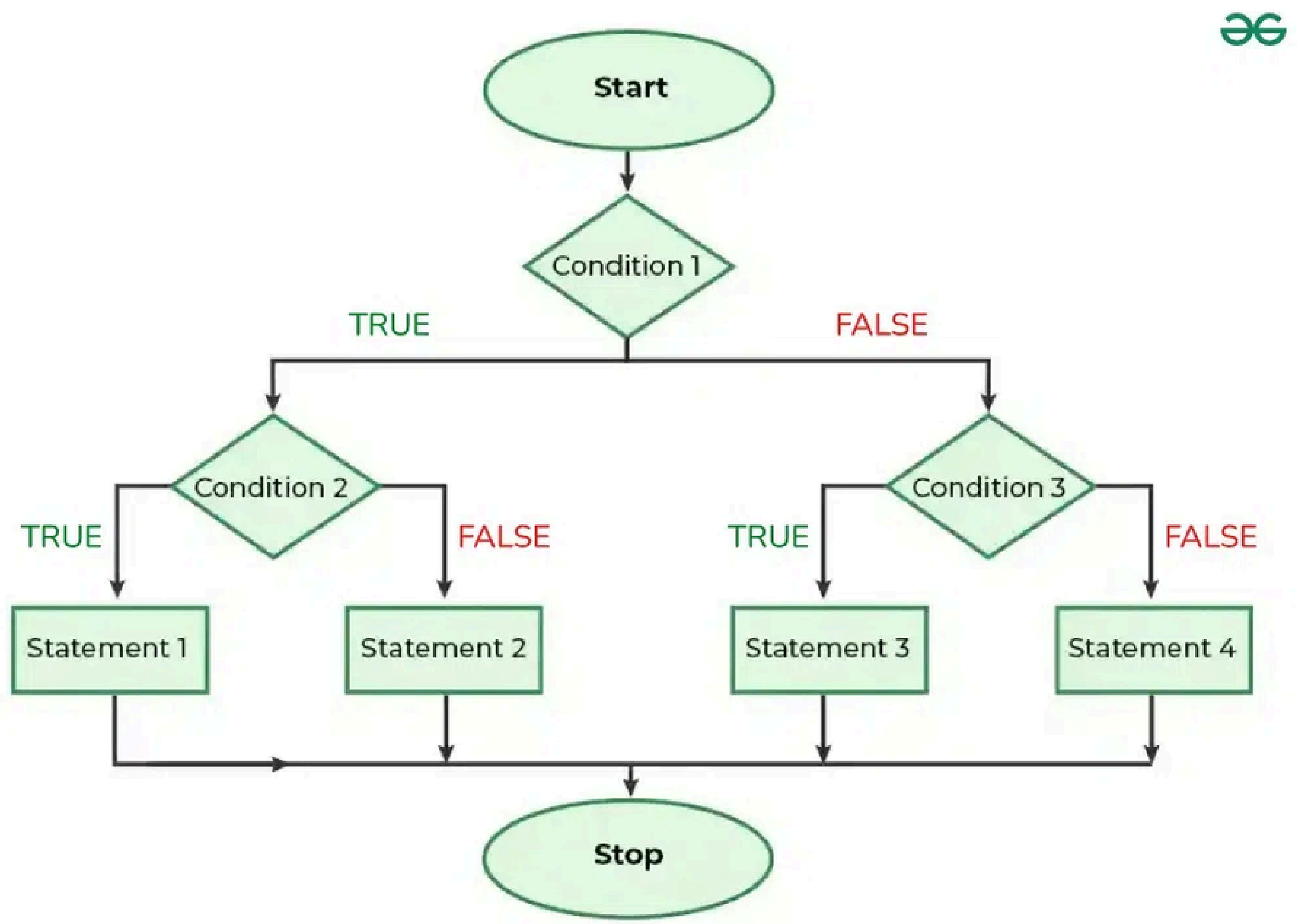


nested if else Statement

A nested if-else statement in C is an if-else statement that is placed inside another if or else statement



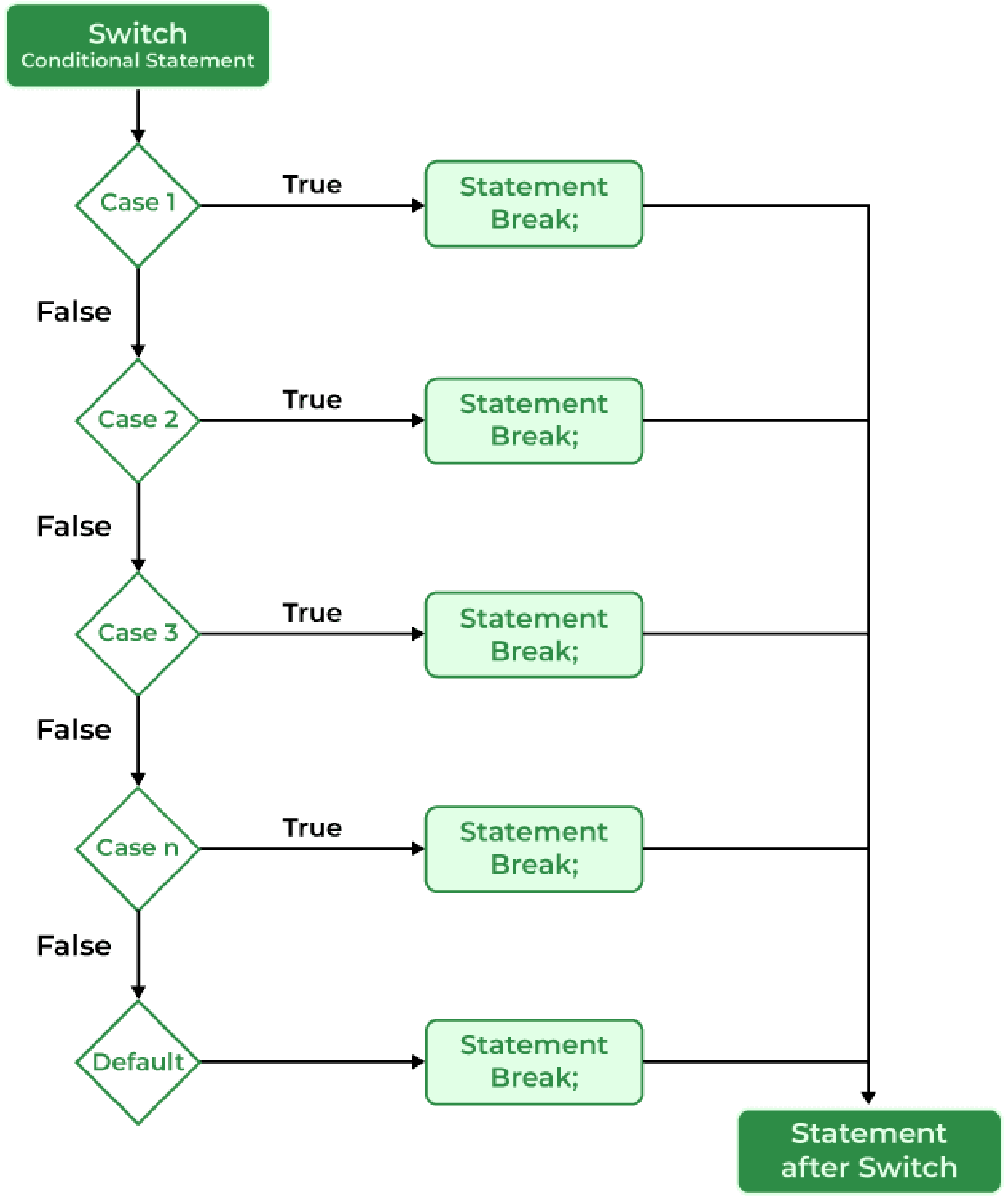
```
#include<stdio.h>
int main()
{
    int num=1;
    if(num<10){
        if(num==1){
            printf("The value is:%d\n",num);
        }
        else{
            printf("The value is greater than 1");
        }
    }
    else{
        printf("The value is greater than 10");
    }
    return 0;
}
```



Switch Statement

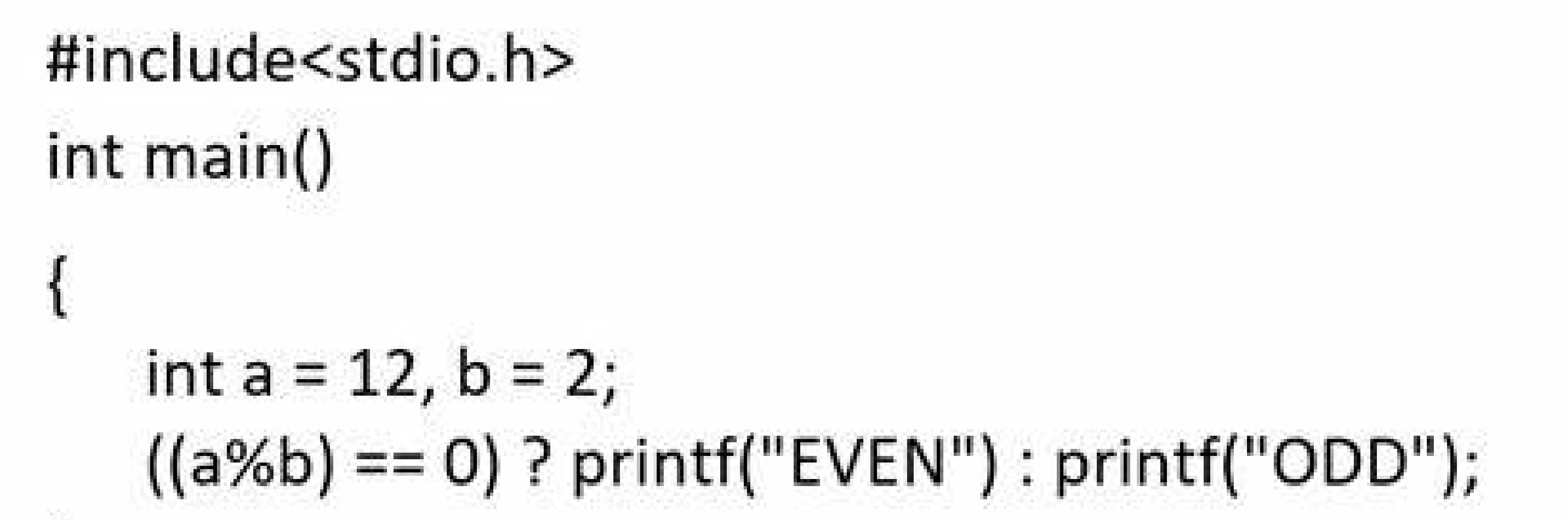
The switch statement in C is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possibles values of a single variable called switch variable.

```
// C Program to illustrate the use of switch statement
#include <stdio.h>

int main()
{
    // variable to be used in switch statement
    int var = 2;

    // declaring switch cases
    switch (var) {
    case 1:
        printf("Case 1 is executed");
        break;
    case 2:
        printf("Case 2 is executed");
        break;
    default:
        printf("Default Case is executed");
        break;
    }

    return 0;
}
```



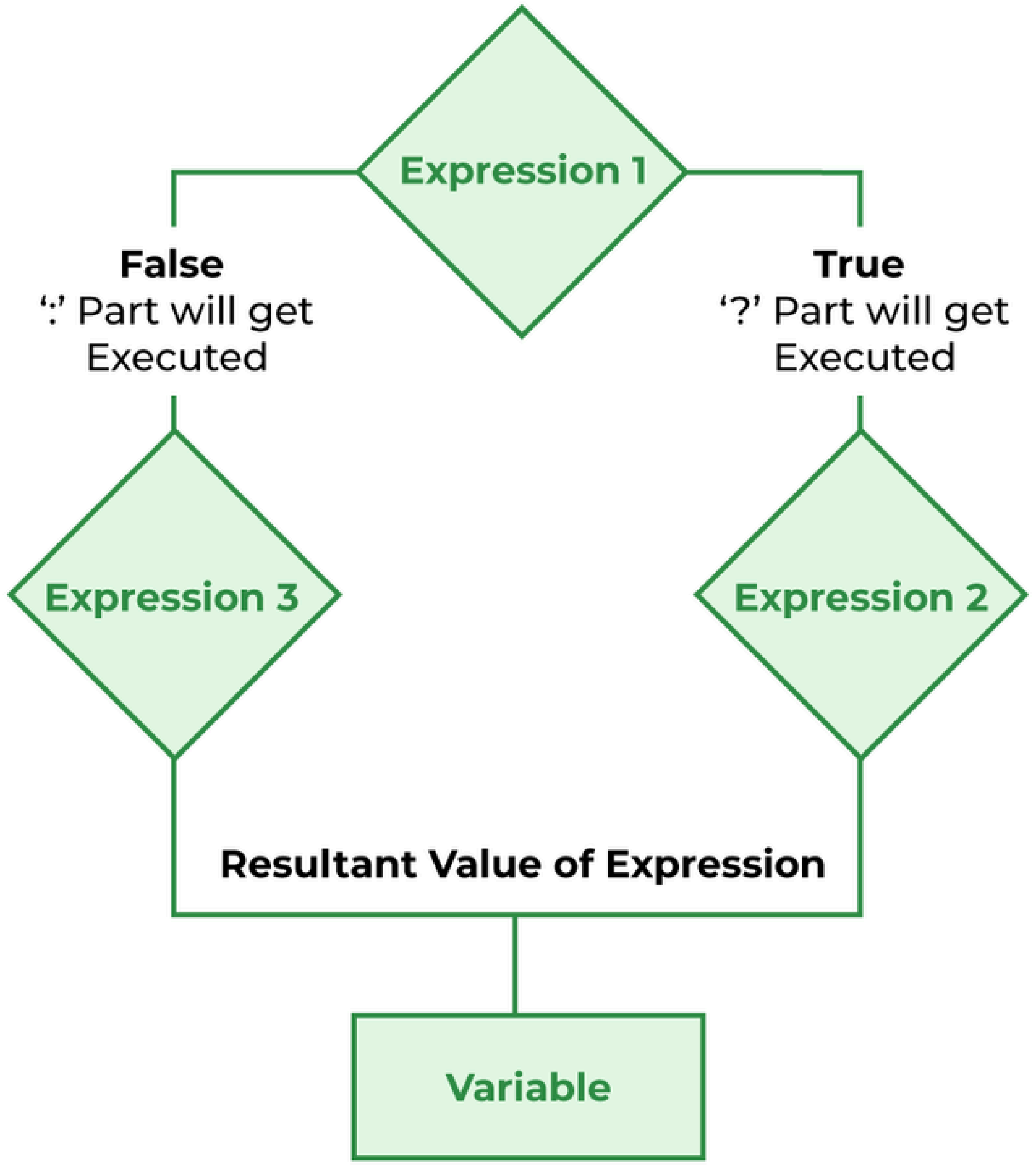
Ternary Conditional Operator

Condition ? doSomething if TRUE : doSomething if FALSE;



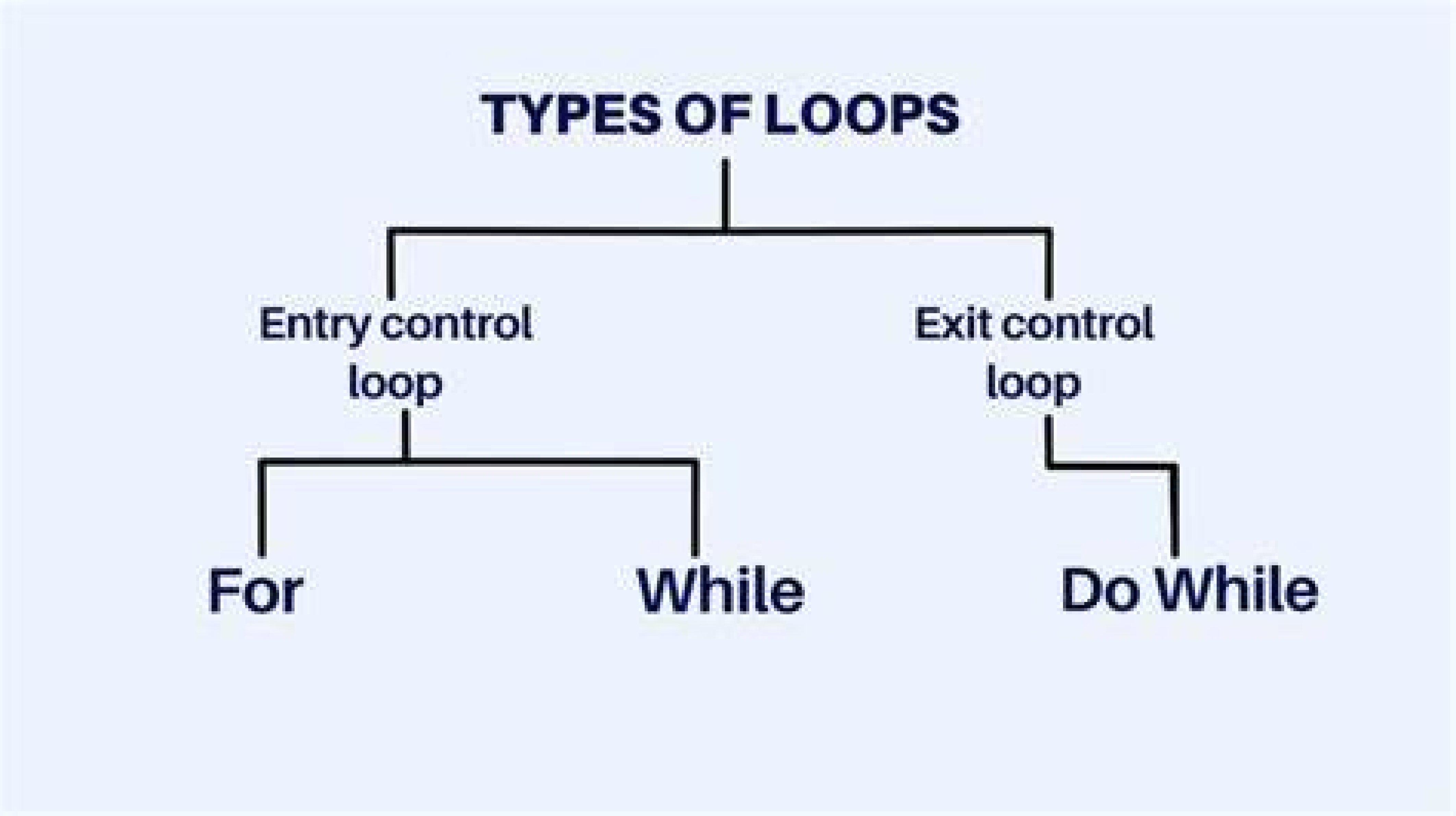
```
#include<stdio.h>
int main()
{
     int a = 12, b = 2;
     ((a%b) == 0) ? printf("EVEN") : printf("ODD");
```

```
return 0;
}
```



Loop Control Instructions

To repeat some parts of the program.



TYPES OF LOOPS

Entry control loop

Exit control loop

For

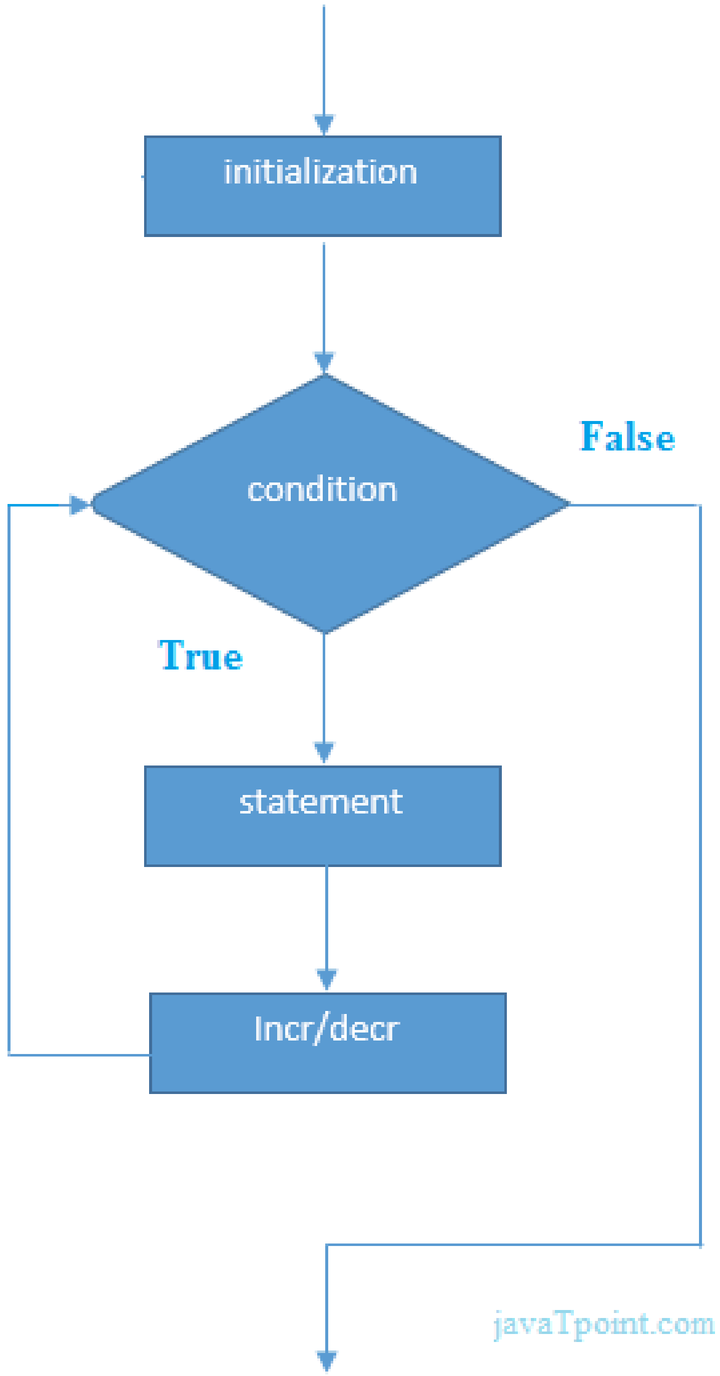
While

Do While

For Loop

```
for(initialisation; condition; updation){
        //do something
}
```

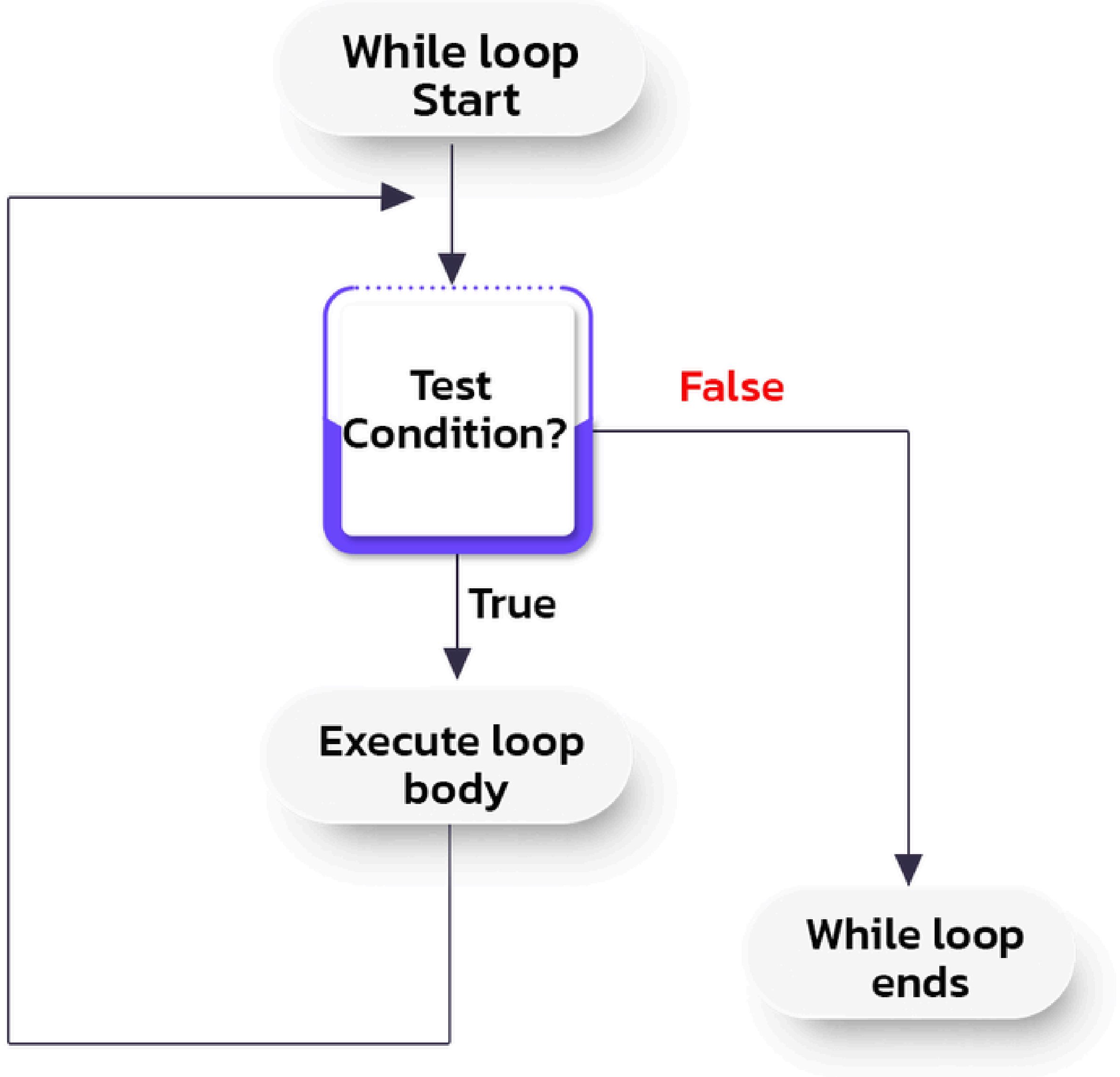
```
# Program to print the numbers from one to ten
 #include<stdio.h>
 int main(){
      int i;
     for(i=1;i<=10;i++){
         printf("%d \n",i);
     }
     return 0;
 }
```



While Loop

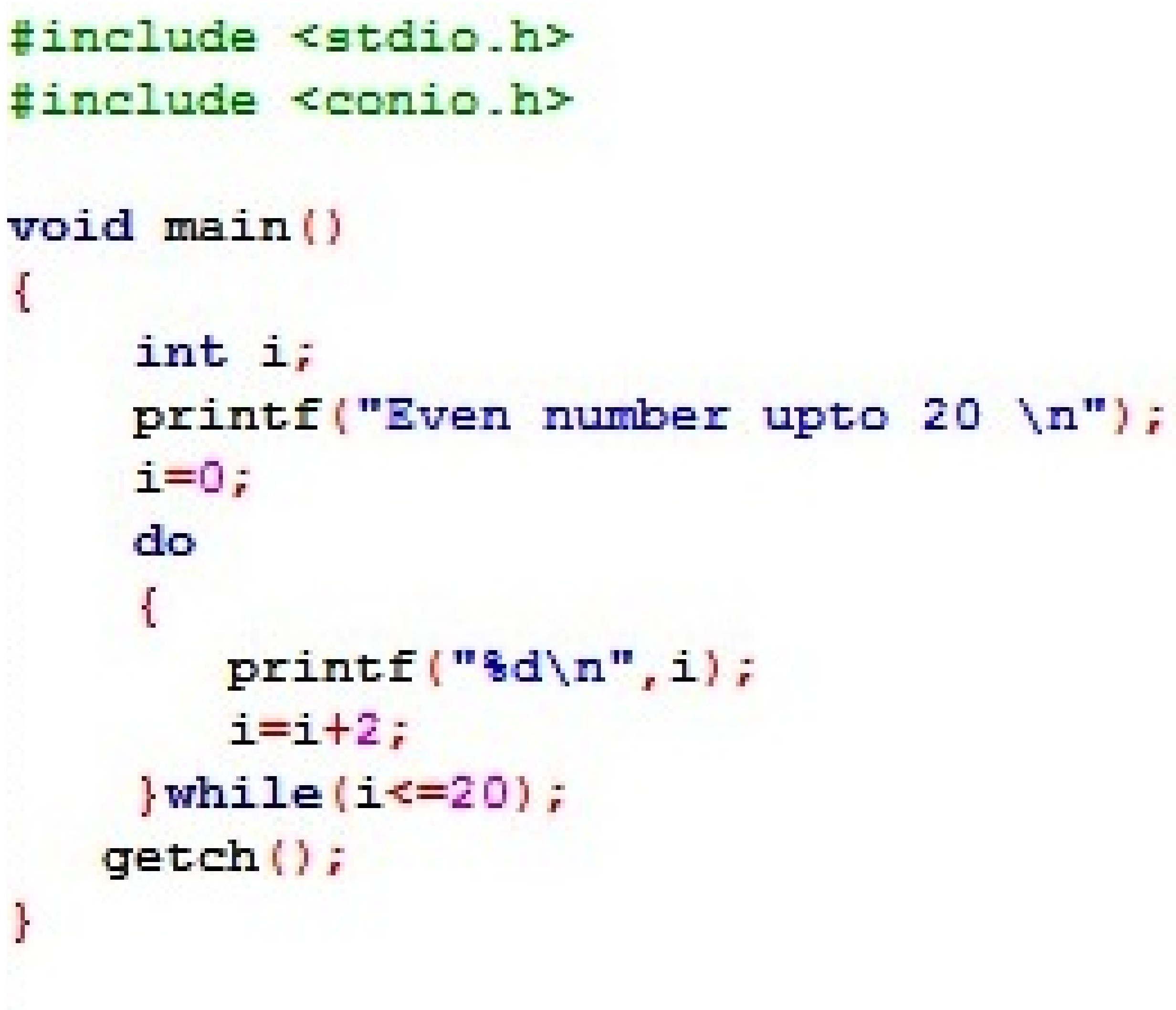
```
while(condition) {
        //do something
        //updation
}
```

```
# Program to print the numbers from one to ten
#include <stdio.h>
int main() {
    int i = 1;
    while (i <= 10) {
        printf("%d ", i);
        i++;
    }
    return 0;
}
```

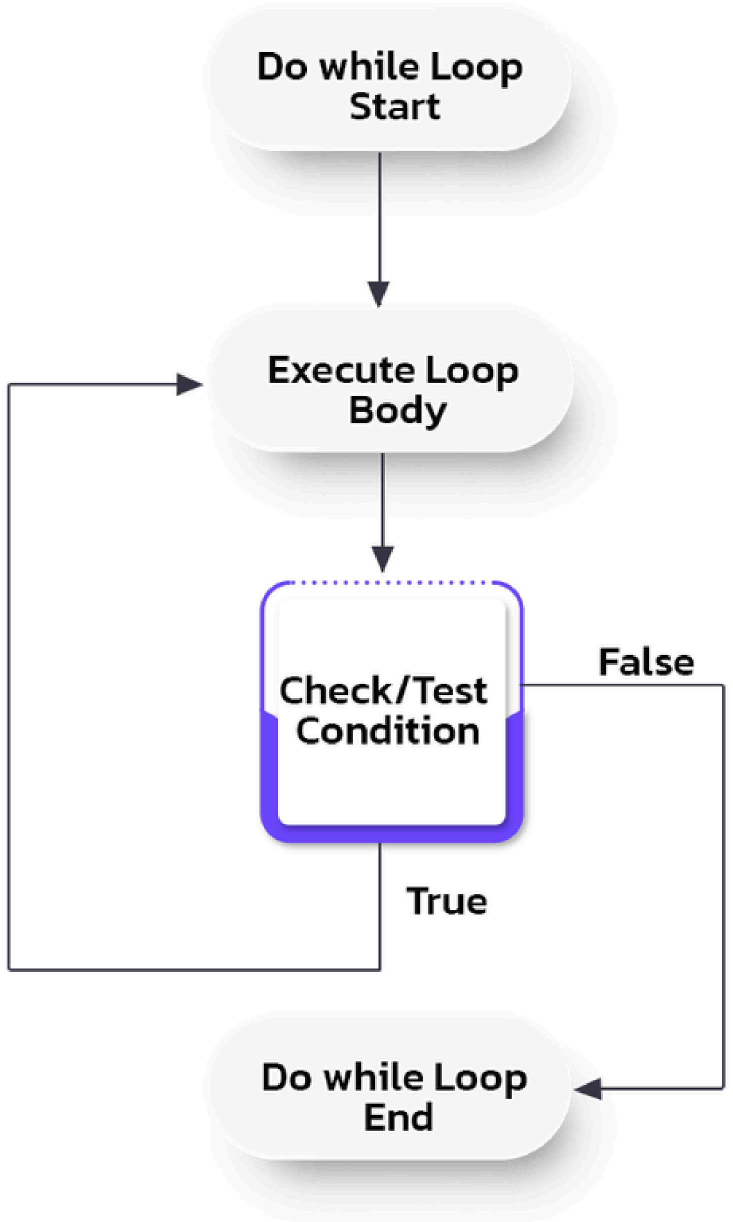


Do While

```
do {
        //do something
        //updation
} while(condition);
```



```
#include <stdio.h>
#include <conio.h>

void main()
{
     int i;
     printf("Even number upto 20 \n");
     i=0;
     do
     {
        printf("%d\n",i);
        i=i+2;
     }while(i<=20);
    getch();
}
```



Break Statement

```
#include<stdio.h>
#include<stdlib.h>
void main ()
{
 int i;
 for(i = 0; i<10; i++)
 {
 printf("%d ",i);
 if(i == 5)
 break;
 }
 printf("came outside of loop i = %d",i);

}
```

break Statement

↓

Exit the loop

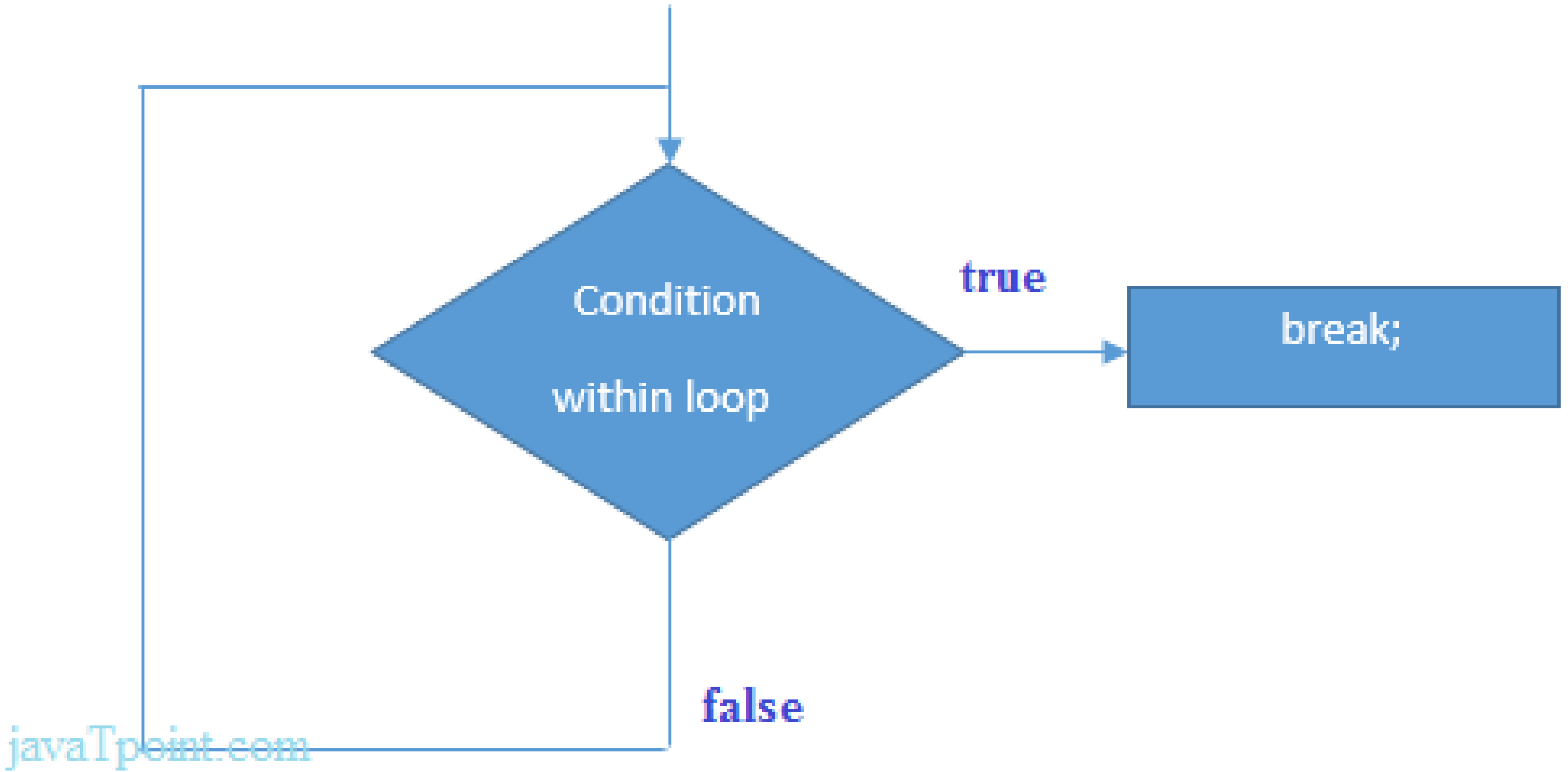
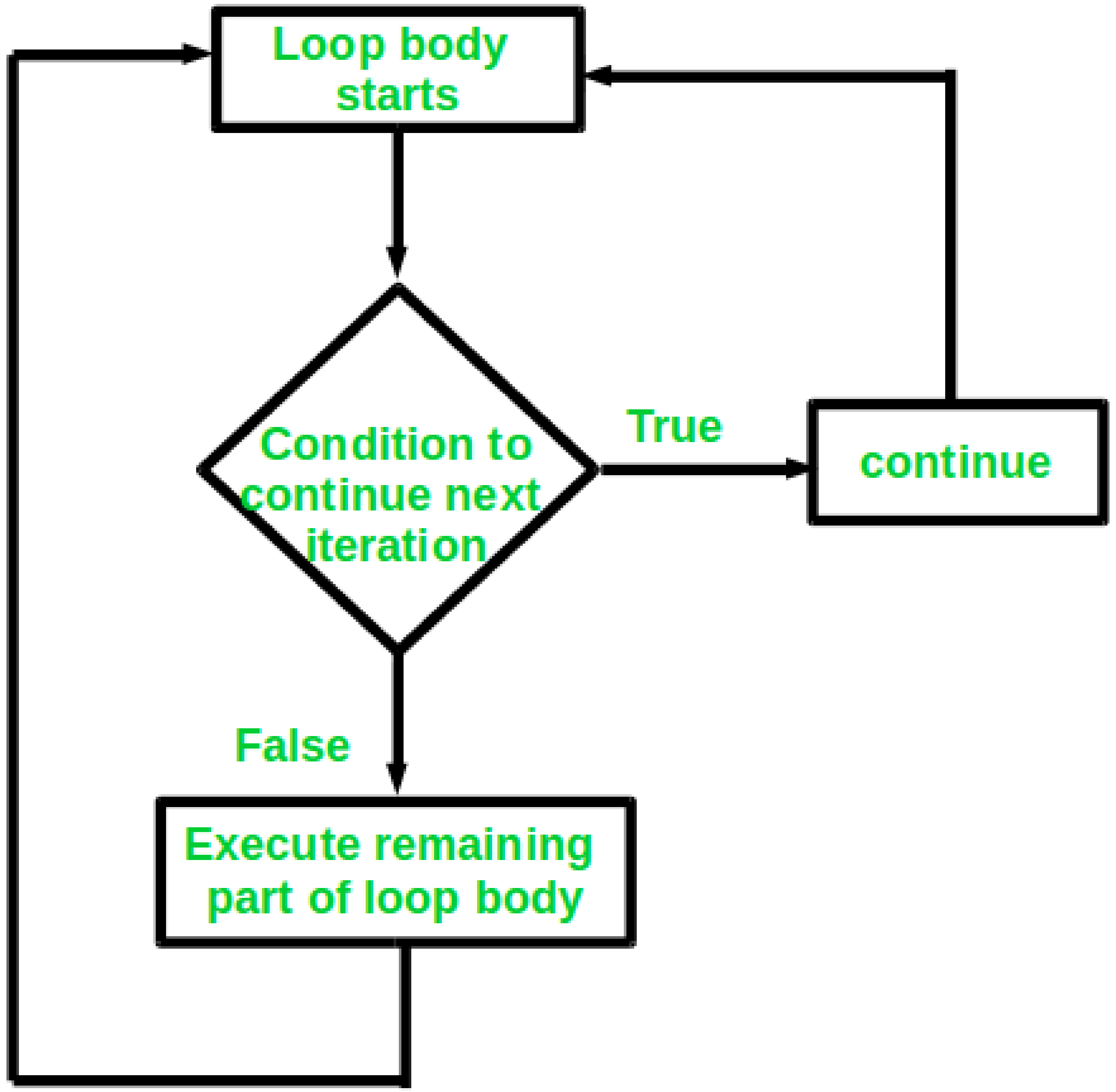


Figure: Flowchart of break statement

Continue Statement

Continue Statement → Skip to next iteration

```
#include<stdio.h>
int main ()  {
    int i = 0;
    while(i!=10) {
        printf("%d", i);
        continue;
        i++;
    }
return 0;
}
```

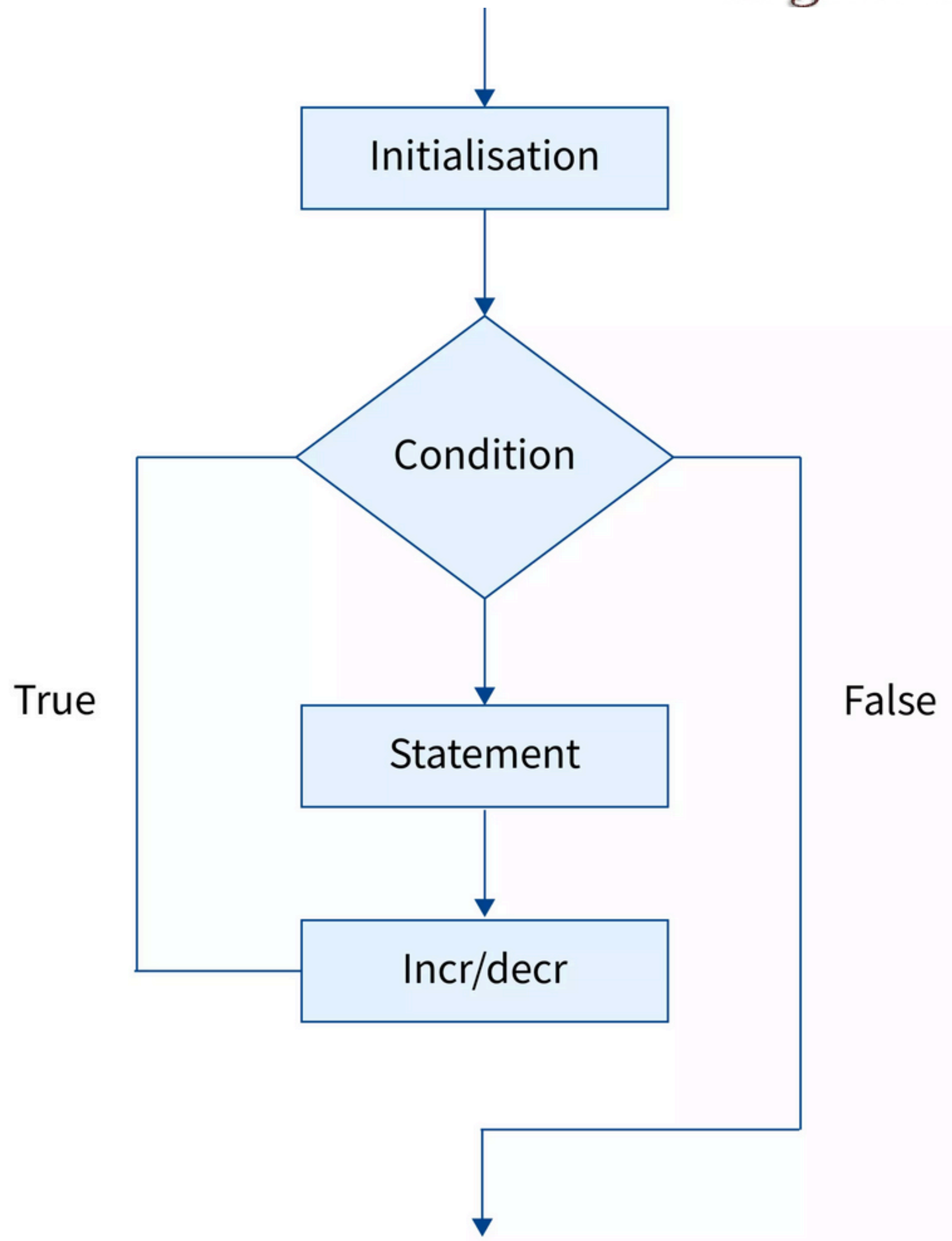


Loop body starts
Condition to continue next iteration
True
continue
False
Execute remaining part of loop body

Nested Loops

```
for( .. ) {
    for( .. ) {
        //do something
    }
}
```

```
#include <stdio.h>
int main()
{
 int n;// variable declaration
 printf("Enter the value of n :");
scanf("%d",&n);
 for(int i=1;i<=n;i++) // outer loop
 {
 for(int j=1;j<=10;j++) // inner loop
 {
 printf("%d\t",(i*j)); // printing the value.
 }
 printf("\n");
}
return 0;
 }
```



Programming with C

The End

E-Gov Cell