

Comprehensive Question for

CSC 183 - Programming C

What is an Identifier?	2
Data types	2
What is constant?	2
What is variable?	2
Expressions	3
Operators	3
1. Arithmetic Operators	3
2. Assignment Operator	3
3. Relational Operator	3
Control Statement	4
String	4
Array	4
Two dimensional arrays	4
Function	4
Local, Global and Static variable	5
Local variable:	5
Global variable	5
Static variables	6
Recursion	6
Pointer	7

What is an Identifier?

Identifiers are user defined words used to name entities like variables, arrays, functions, structures etc.

Rules for naming identifiers are:

- 1) Name should only consist of alphabets (both upper and lower case) or digits or underscore (_) sign.
- 2) First character should be the alphabet or underscore.
- 3) Name should not be a keyword.
- 4) Since C is case sensitive, the upper case and lower case are considered differently, for example code, Code, CODE etc. are different identifiers.
- 5) Identifiers are generally given in some meaningful name such as value, net_salary, age, data etc. An identifier name may be long, some implementations recognize only first eight characters, most recognize 31 characters. ANSI standard compiler recognizes 31 characters. Some invalid identifiers are 5cb, int, res#, avg no etc.

Data types

1. Int
2. Float
3. Double
4. Char
5. Pointer
6. Array
7. Void



What is constant?

Constant is any value that cannot be changed during program execution.

What is variable?

Variable is a data name which is used to store some data value or symbolic names for storing programs. Declaration of variables specifies its name, data types and range of the value that variables can store depending upon its data types.

Syntax:

- int a;

- char c;
- float f;

Expressions

An expression is a combination of variables, constants, operators and function calls. It can be arithmetic, logical or relational for example:-

- `int z = x + y` // arithmetic expression
- `a > b` // relational
- `a == b` // logical
- `func(a, b)` // function call

Operators

1. Arithmetic Operators

These operators are used for numeric calculation. These are either Unary arithmetic operator, Binary arithmetic operators. Where Unary arithmetic operator required 25 *under revision only one operand such as +, -, ++, --, !, tiled. And these operators are addition, subtraction, multiplication, division.

2. Assignment Operator

A value can be stored in a variable with the use of assignment operator. The assignment operator(=) is used in assignment statements and assignment expressions. For example,

- `int x = y;`
- `int Sum = x + y + z;`

3. Relational Operator

It is used to compare the value of two expressions depending on their relation. Expression that contains a relational operator is called relational expression. Here the value is assigned according to true or false value.

- `(a >= b) || (b > 20)`
- `(b > a) && (e > b)`
- `(b != 7)`

Control Statement

Control statement defines how the control is transferred from one part to the other part of the program. There are several control statements like if...else, switch, while, do....while, for loop, break etc.

Loops in C

It is a block of statements that performs a set of instructions. There are three types of loops in c:

- [While loop](#)
- [do while loop](#)
- [for loop](#)

String

Array of characters is a string.

Array

Array is the collection of similar data types or collection of similar entities stored in contiguous memory locations. Array of characters is a string. Example: `int arr[100];`

Declaration of an array :

Its syntax is :

Data type array name [size];

`int arr[100];`

`int mark[100];`

`int a[5]={10,20,30,100,5}`

Two dimensional arrays

Two dimensional array is known as a matrix. Example: `int a[2][3];`

Function

A function is a self-contained block of codes or sub programs with a set of statements that perform some specific task or coherent task when it is called.

Local, Global and Static variable

Local variable:

variables that are defined within a body of function or block. The local variables can be used only in that function or block in which they are declared. Same variables may be used in different functions such as

```
#include <stdio.h>

int sum(){
    // local variable of sum()
    int a = 10, b = 20;
    printf("%d + %d = %d", a, b, a+b);
}
```

```
int tryToAccessLocalVariable(){
    // It will create an error
    printf("%d", a);
}
```

```
int main(){
    sum();
    tryToAccessLocalVariable();
}
```

Run

Global variable

The variables that are defined outside of the function are called global variables. All functions in the program can access and modify global variables. Global variables are automatically initialized at the time of initialization.

Example:

```
#include <stdio.h>

int globalVariable = 10;

int fun1(){
```

```

        printf("%d\n", globalVariable);
    }

int fun2()
{
    printf("%d\n", globalVariable);
}

int main()
{
    fun1();
    fun2();
}

```

Run

Static variables

A static variable is a type of variable that retains its value between function calls. In the context of object-oriented programming, a static variable belongs to the class rather than any instance of the class. Static variables are initialized only once, and they have a single memory location. In C and C++, static variables are often used for counting function calls, maintaining state, or sharing data across function calls.

```

#include <stdio.h>

void countCalls() {
    static int count = 0; // static variable initialization
    count++;
    printf("Function called %d times\n", count);
}

int main() {
    for(int i = 0; i < 5; i++) {
        countCalls();
    }
    return 0;
}

```

```
}
```

Run

Recursion

When a function calls itself (inside the body) again and again then it is called a recursive function.

Ex:- /*calculate factorial of a no.using recursion*/

```
fact (int num)
```

```
{
```

```
    If (num == 0 || num == 1)
```

```
        return 1;
```

```
    else
```

```
        return(num*fact(num - 1));
```

```
}
```

```
void main()
```

```
{
```

```
    int num;
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &num);
```

```
    f = fact(num);
```

```
    printf("factorial is = %d\n", f);
```

```
}
```

Pointer

A pointer is a variable that stores a memory address or that contains the address of another variable where addresses are the location number and always contains the whole number.

Syntax-

Data type *pointer name;

Example:

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

```
int main()
```

```
{
```

```
int a = 10;
```

```
int *p = &a;
```

```
printf("Value of a = %d\n", a);
```

```
printf("Value of a using pointer = %d\n", *p);
```

```
printf("\nMemory address of a = %d\n", &a);
```

```
printf("Memory address of a using pointer = %d\n", p);
```

```
}
```

Run

