

Variables:

Variable is a container is used to store the values in our programs.

Variable is a named memory [bytes] location where we can store and manipulate [modify] the values in our programs.

All the variables are stored in primary memory [RAM] only. Once the program execution is completed, automatically the variables are deleted from memory.

Variables are case sensitive. i.e. lower and upper are different.
eg:

```
Int a=100;
```

```
Int A=200;
```

In C compiler we should have to declare the variables in first line of any function. But in C++ we can declare anywhere. Every variable is having 2 stages.

1. Variable declaration / declared

Eg: `int a;`

2. Variable initialization / defined

Eg: `a=100;`

When a variable is initialized then only memory allocated.

4 bits = 1 nibble

8 bits = 1 byte

1024 bytes=1kilo byte

1024 kb = 1 mega byte

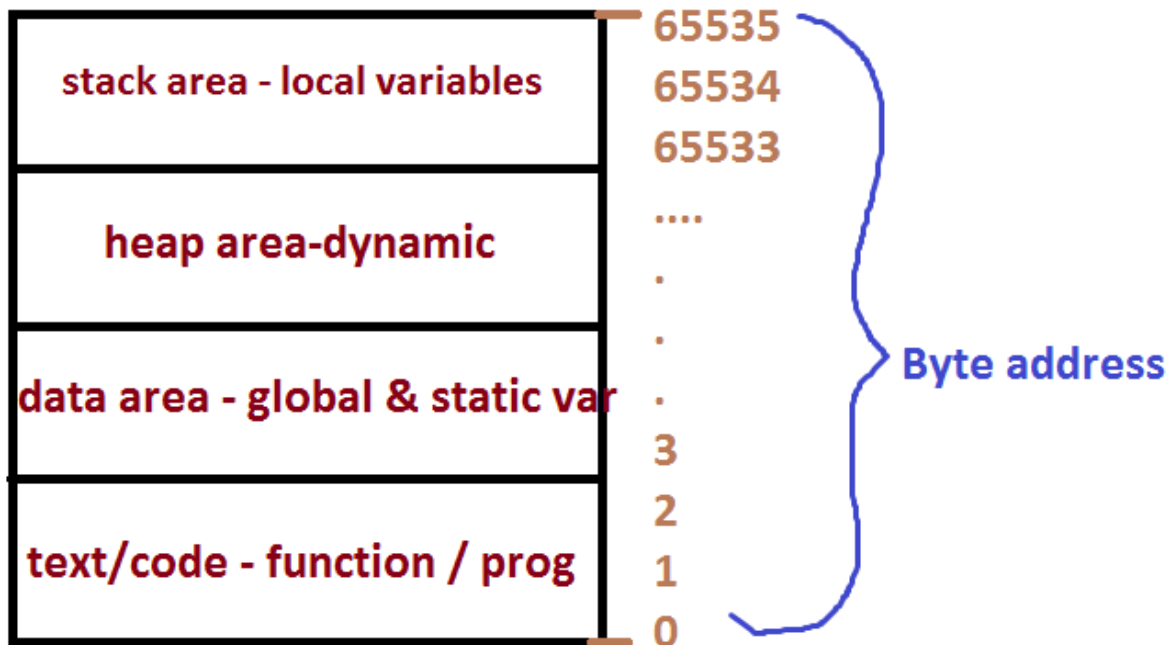
1024 mb = 1 giga byte

1024 gb = 1 tera byte

1024 tb = 1 peta byte

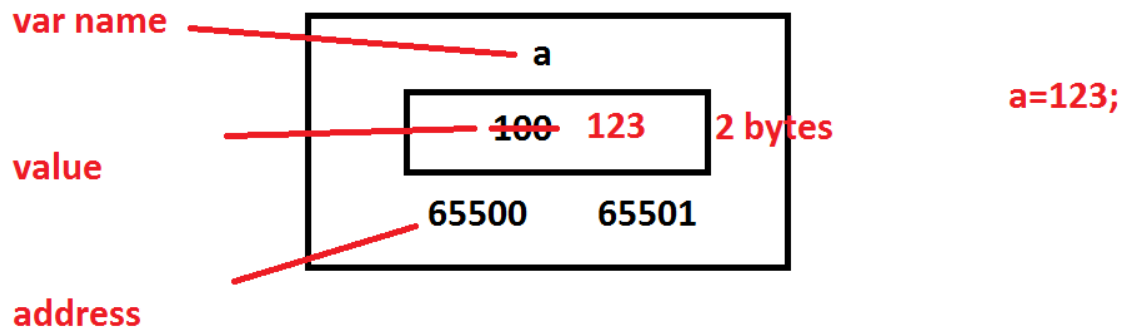
RAM STACK

64KB-65536 BYTES



`int a=100;`

RAM - BYTES

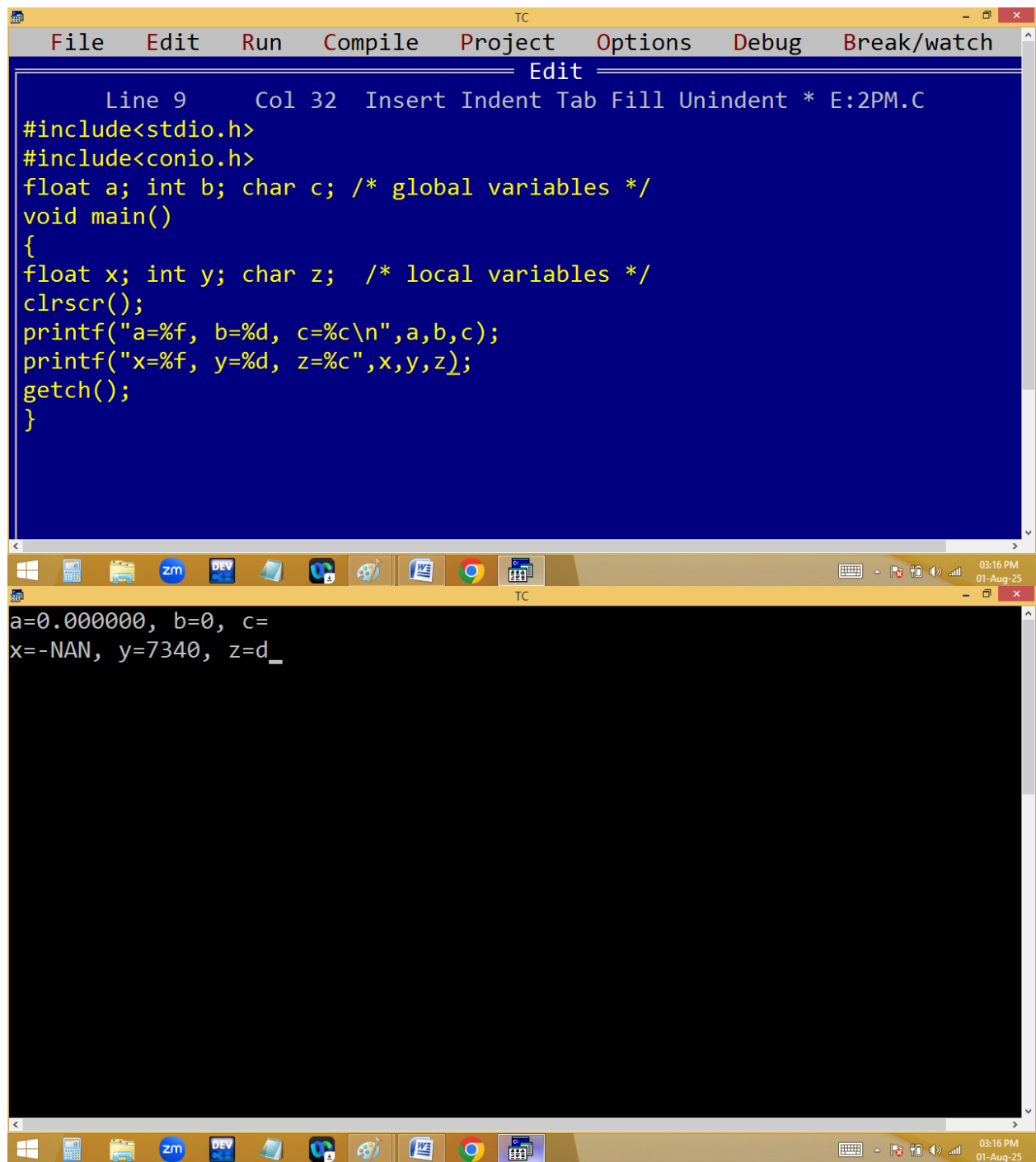


We are having 2 types of variables.

1. Local variables
2. Global variables

	Local variables	Global variables
Declaration area	Within function	Top of the program
Storage area	Stack area	Data area
Initial/default values	Garbage values	Int-0,char-blank, Float-0.000000
Scope-where we can access	Within function.	Anywhere
Life time – until what time they are active in memory	After function execution	After main() / total program execution

Finding initial values of local and global variables:

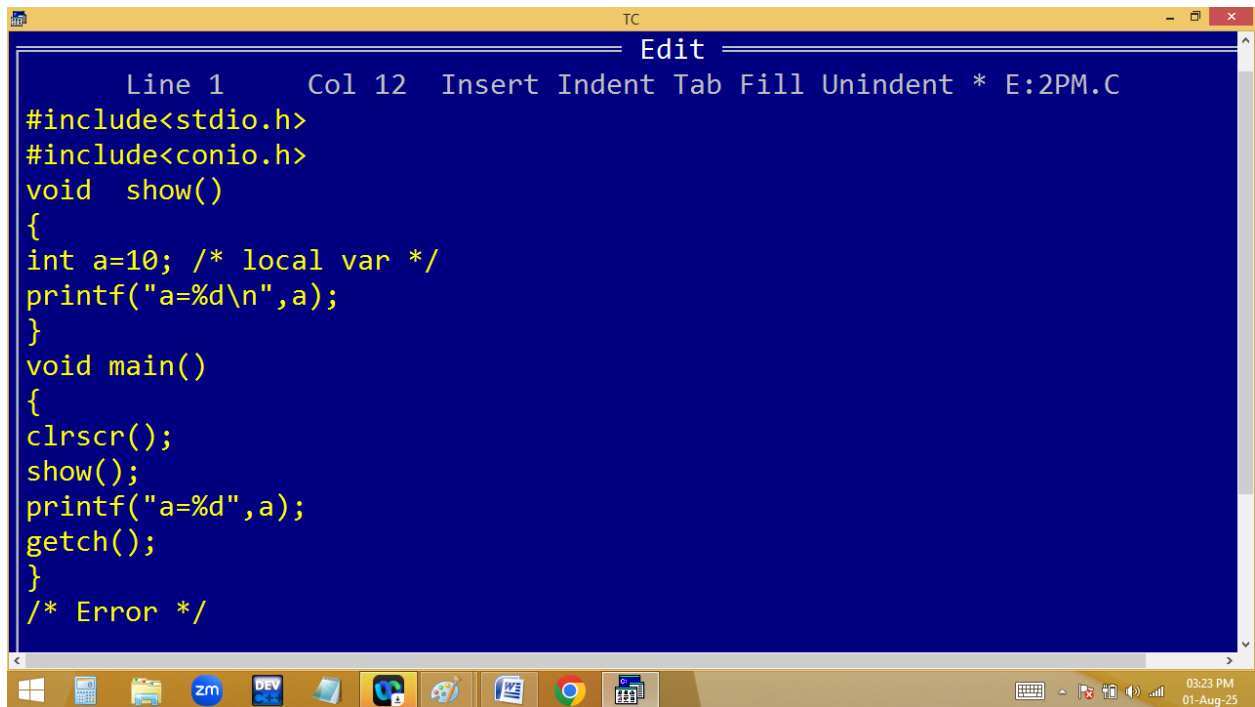


The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the 'Edit' window, displaying a C program. The program includes `<stdio.h>` and `<conio.h>`, declares global variables `float a;`, `int b;`, and `char c;`, and defines a `main()` function. Inside `main()`, it declares local variables `float x;`, `int y;`, and `char z;`, calls `clrscr()`, and uses `printf` to print the values of `a`, `b`, `c`, `x`, `y`, and `z`. It also calls `getch()` before returning. The bottom window is the 'Output' window, which shows the execution results: `a=0.000000, b=0, c=` on the first line and `x=-NAN, y=7340, z=d_` on the second line. The Windows taskbar at the bottom shows the time as 03:16 PM on 01-Aug-23.

```
Line 9      Col 32  Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
float a; int b; char c; /* global variables */
void main()
{
float x; int y; char z; /* local variables */
clrscr();
printf("a=%f, b=%d, c=%c\n",a,b,c);
printf("x=%f, y=%d, z=%c",x,y,z);
getch();
}
```

a=0.000000, b=0, c=
x=-NAN, y=7340, z=d_

Finding scope of local variable:

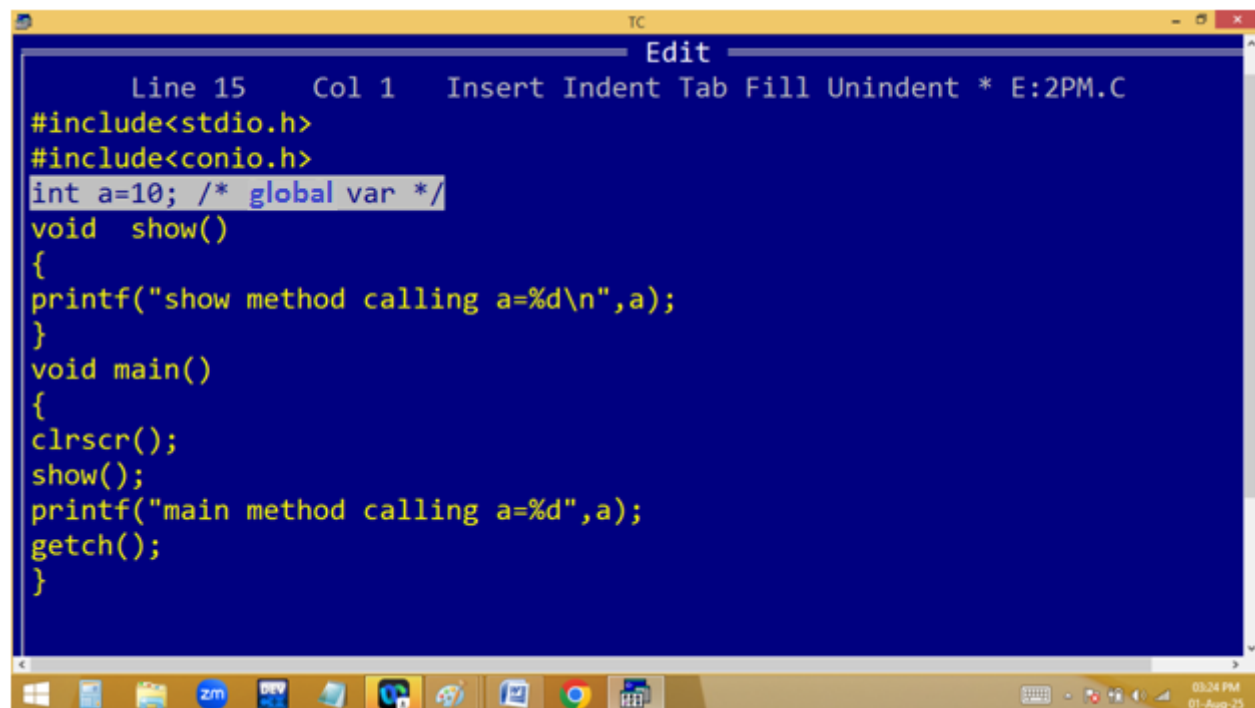


The screenshot shows the Turbo C++ (TC) editor window. The title bar says 'TC'. The menu bar includes 'Edit'. The status bar at the top indicates 'Line 1 Col 12 Insert Indent Tab Fill Unindent * E:2PM.C'. The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
void show()
{
int a=10; /* local var */
printf("a=%d\n",a);
}
void main()
{
clrscr();
show();
printf("a=%d",a);
getch();
}
/* Error */
```

The Windows taskbar is visible at the bottom with various icons and a system clock showing 03:23 PM on 01-Aug-25.

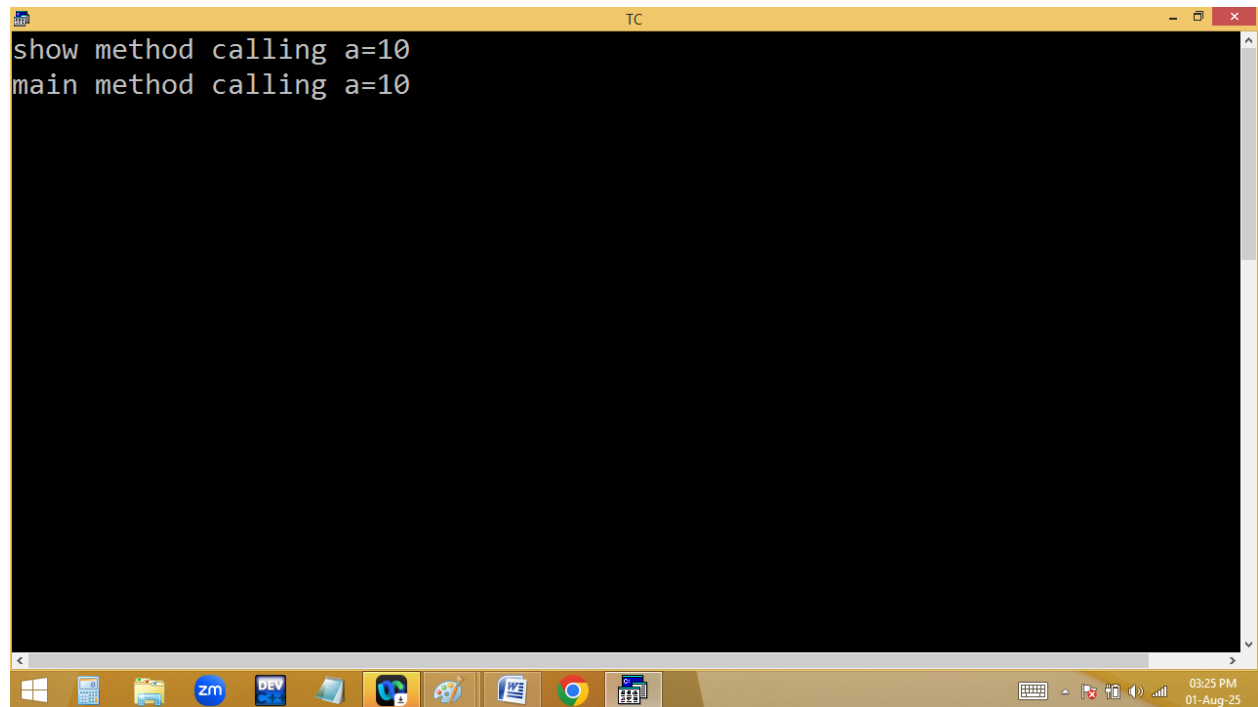
Finding global variable scope:



The screenshot shows the Turbo C++ (TC) editor window. The title bar says 'TC'. The menu bar includes 'Edit'. The status bar at the top indicates 'Line 15 Col 1 Insert Indent Tab Fill Unindent * E:2PM.C'. The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
int a=10; /* global var */
void show()
{
printf("show method calling a=%d\n",a);
}
void main()
{
clrscr();
show();
printf("main method calling a=%d",a);
getch();
}
```

The line containing 'int a=10; /* global var */' is highlighted. The Windows taskbar is visible at the bottom with various icons and a system clock showing 03:24 PM on 01-Aug-25.



The image shows a screenshot of a Turbo C++ (TC) window. The window has a yellow title bar with the text "TC" in the center. The main area is black with white text. The text consists of two lines: "show method calling a=10" and "main method calling a=10". The window has a standard Windows taskbar at the bottom with various icons and a system clock showing "03:25 PM" and "01-Aug-25".

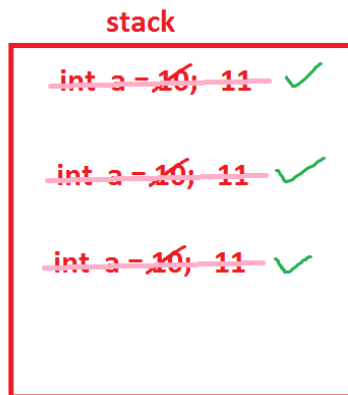
```
show method calling a=10
main method calling a=10
```

Finding life time of a local variable:

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window, titled "Edit", contains the source code for a C program. The code defines a function `show()` that prints the value of a local variable `a` and increments it. The `main()` function calls `show()` three times. The bottom window shows the output of the program, which is three lines of `a=11`. The Windows taskbar at the bottom indicates the time is 03:33 PM on 01-Aug-23.

```
Line 16   Col 1   Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void  show() /* fun definition */
{
  int a=10; /* local var */
  printf("a=%d\n",++a);
}
void main()
{
  clrscr();
  show(); /* fun calling */
  show();
  show();
  getch();
}
```

a=11
a=11
a=11



```
#include<conio.h>
void show() /* fun definition */
{
    int a=10; /* local var */
    printf("a=%d\n",++a);
} /* a deleted */
void main()
{
    clrscr();
    show(); /* fun calling */
    show();
    show();
    getch();
}
```

Finding global variable life time:

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the 'Edit' window, displaying a C program. The code is as follows:

```
Line 7      Col 18  Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
int a=10; /* global var */
void show() /* fun definition */
{
printf("a=%d\n",++a);
}
void main()
{
clrscr();
show(); /* fun calling */
show();
show();
getch();
}
```

The bottom window is the 'Output' window, which displays the output of the program:

```
a=11
a=12
a=13
```

The IDE's status bar at the bottom indicates the time as 03:53 PM on 01-Aug-25. The taskbar shows various application icons, including Windows Explorer, Zm, DEV, and Google Chrome.

stack

int a = ~~10~~ ~~11~~ ✓
12 ✓
13 ✓



```
#include<conio.h>
int a=10; /* global var */
void show() /* fun definition */
{
    printf("a=%d\n",++a);
}
void main()
{
    clrscr();
    show(); /* fun calling */
    show();
    show();
    getch();
}
```