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

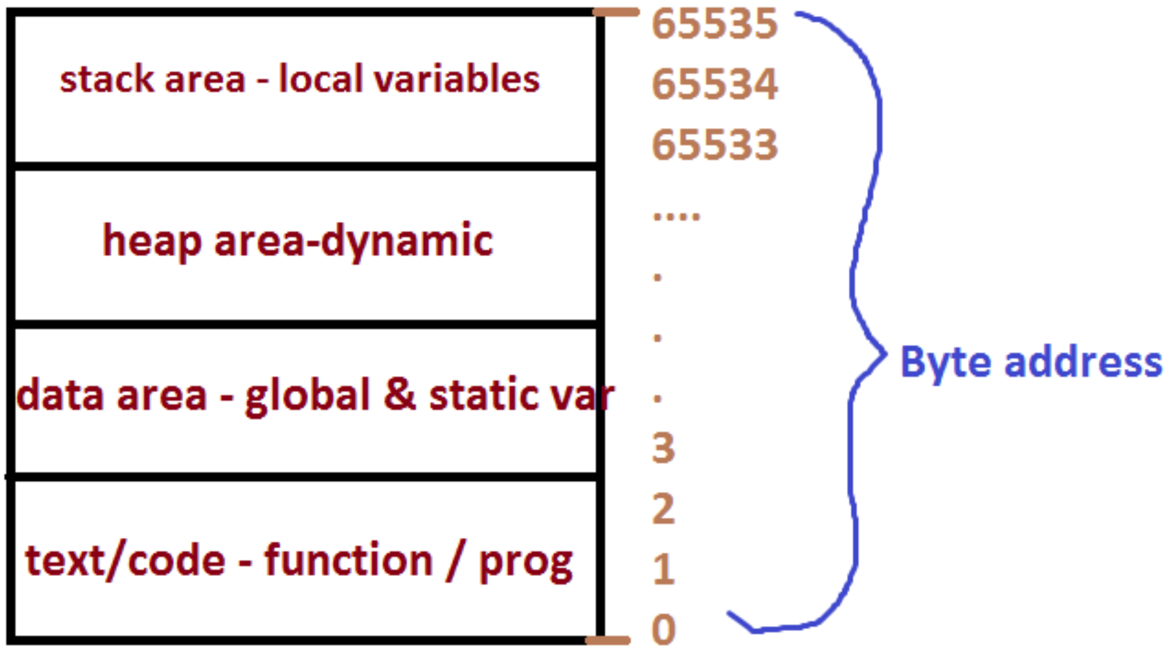
1024 kb = 1 mega byte

1024 mb = 1 giga byte

1024 gb = 1 tera byte

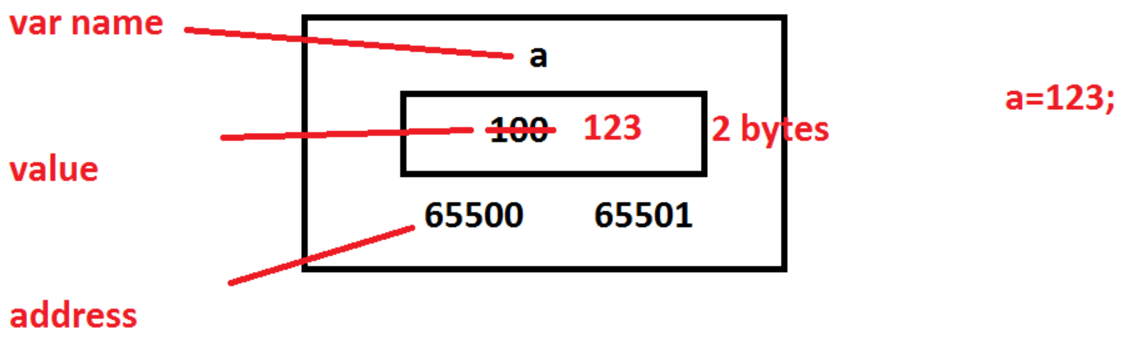
1024 tb = 1 peta byte

RAM STACK
64KB-65536 BYTES


stack area - local variables
heap area-dynamic
data area - global & static var
text/code - function / prog
65535
65534
65533
....
.
.
.
3
2
1
0
Byte address

int a=100;

RAM - BYTES

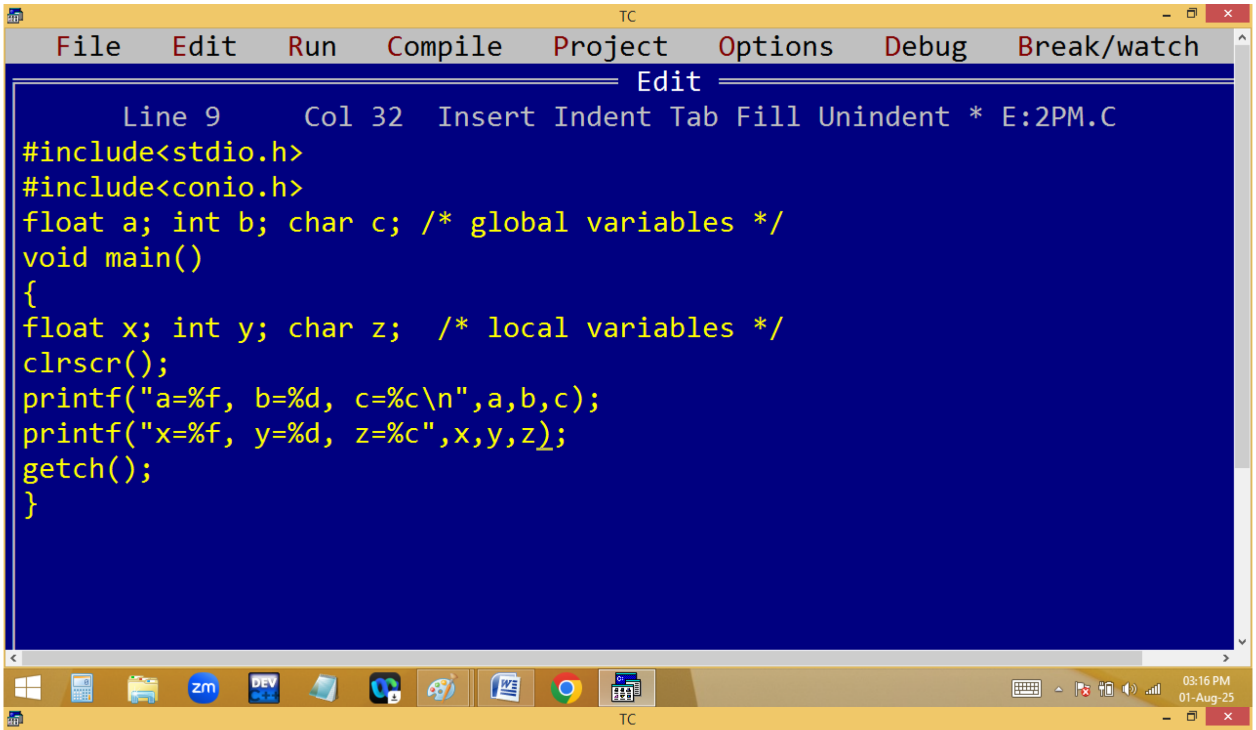


We are having 2 types of variables.

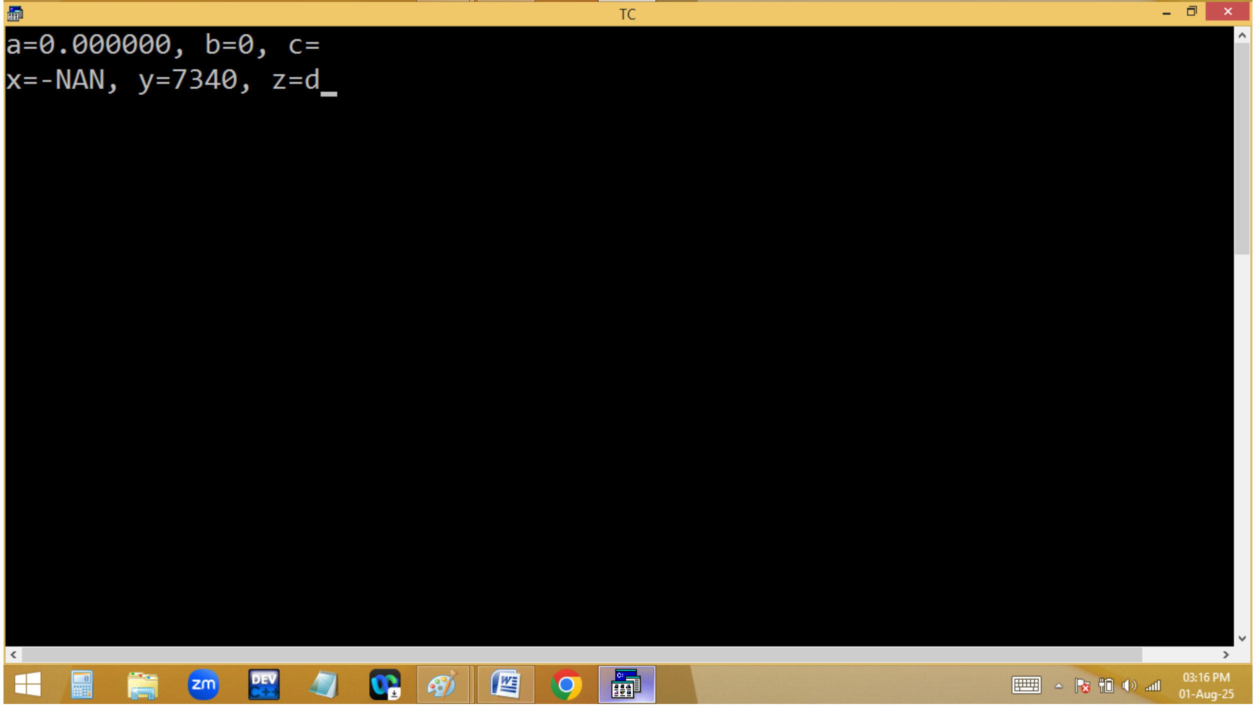
1. Local variables
2. Global variables

	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:

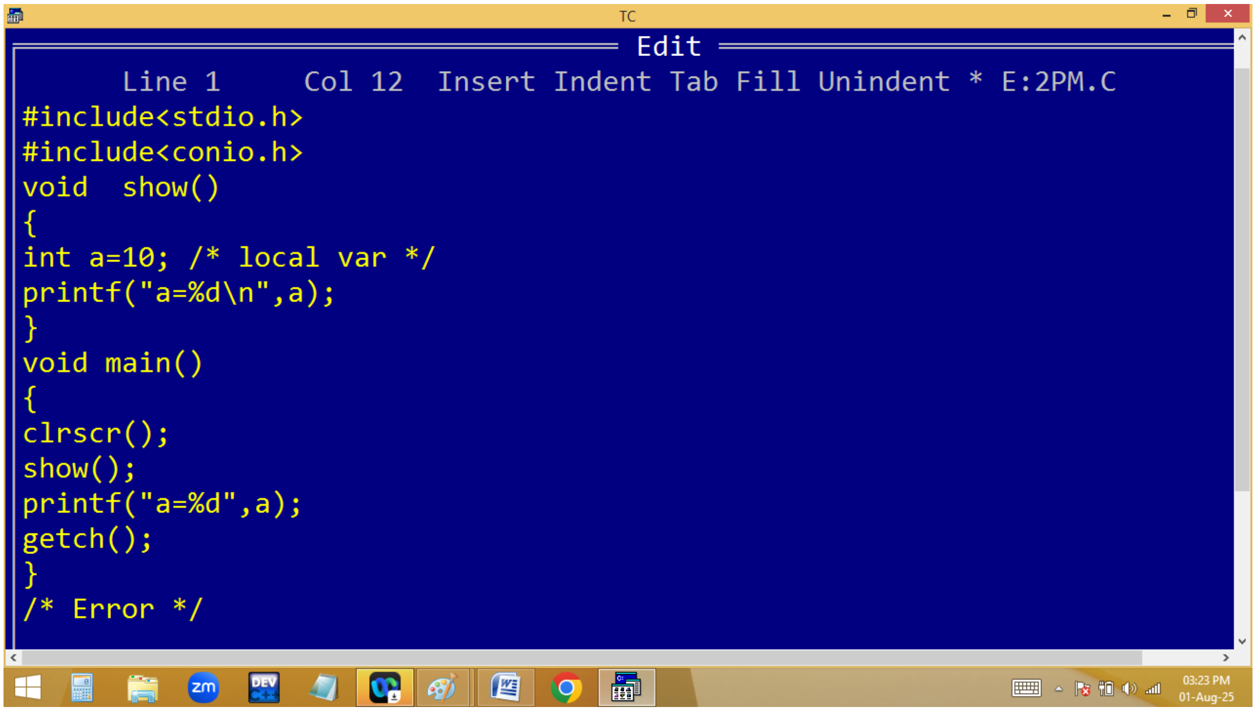


```
#include<stdio.h>
#include<conio.h>
float a; int b; char c; /* global variables */
void main()
{
float x; int y; char z;  /* local variables */
clrscr();
printf("a=%f, b=%d, c=%c\n",a,b,c);
printf("x=%f, y=%d, z=%c",x,y,z);
getch();
}
```



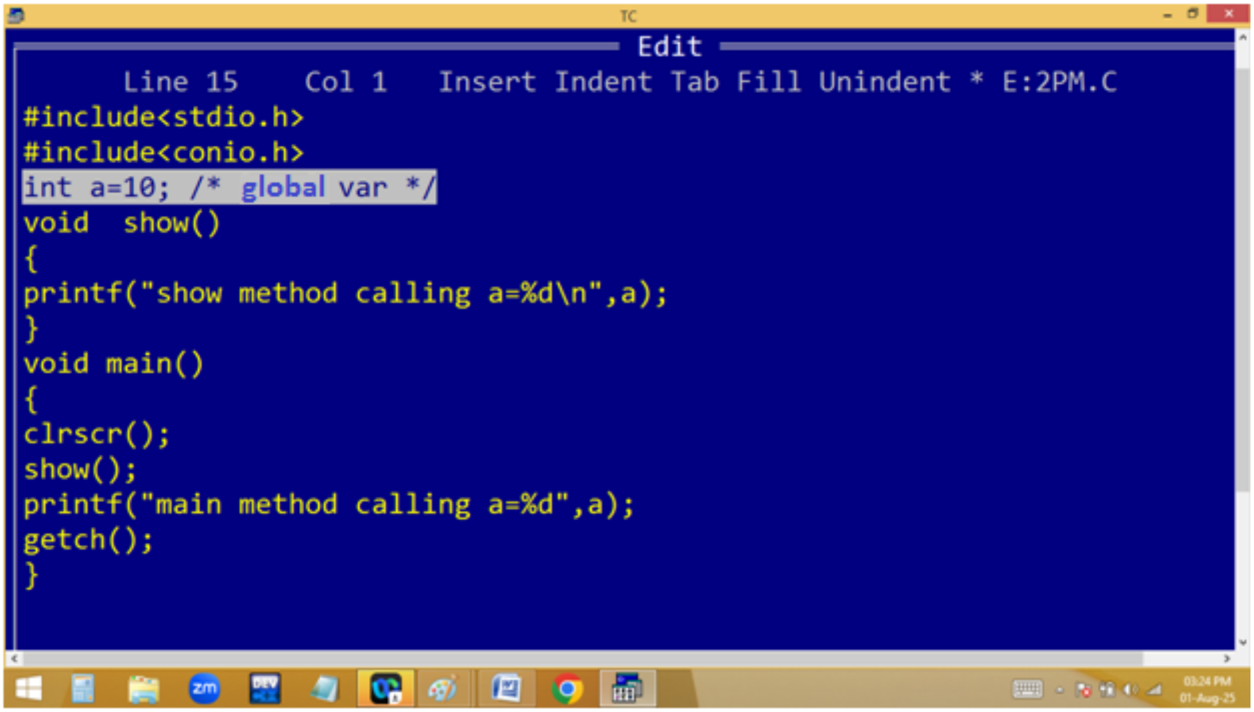
```
a=0.000000, b=0, c=
x=-NAN, y=7340, z=d_
```

Finding scope of local variable:

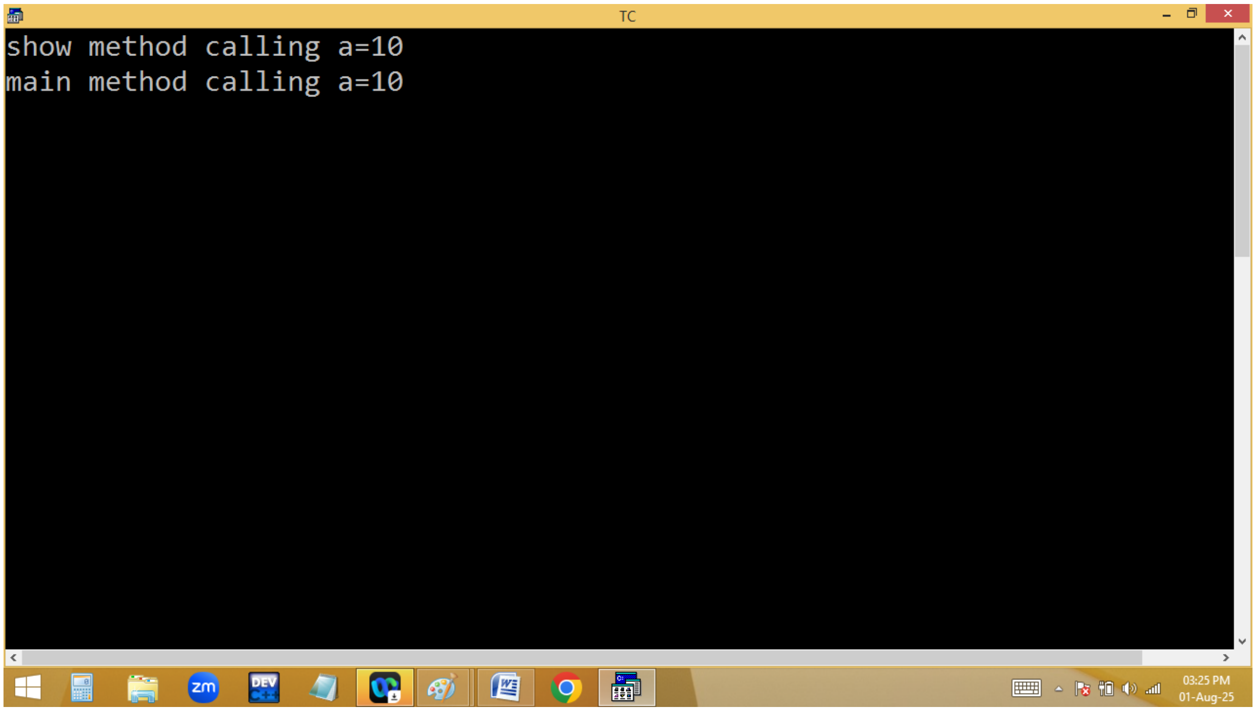


```
#include<stdio.h>
#include<conio.h>
void  show()
{
int a=10; /* local var */
printf("a=%d\n",a);
}
void main()
{
clrscr();
show();
printf("a=%d",a);
getch();
}
/* Error */
```

Finding global variable scope:

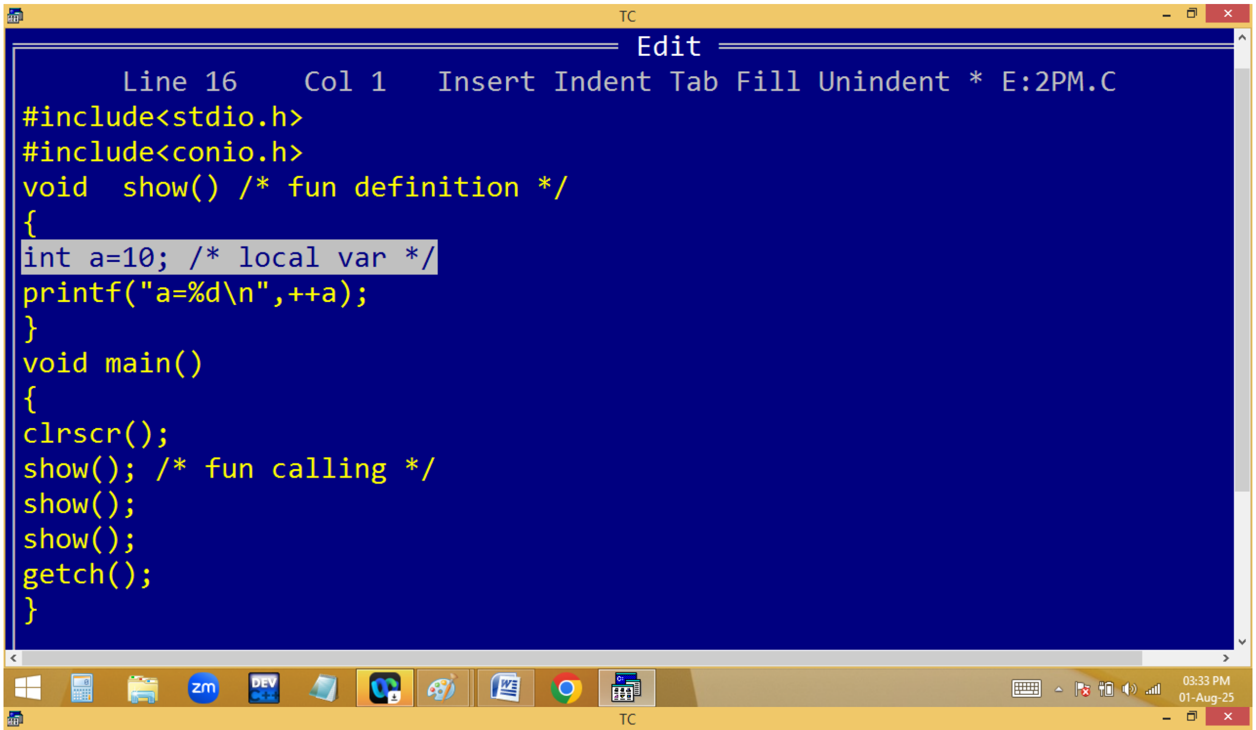


```
#include<stdio.h>
#include<conio.h>
int a=10; /* global var */
void  show()
{
printf("show method calling a=%d\n",a);
}
void main()
{
clrscr();
show();
printf("main method calling a=%d",a);
getch();
}
```

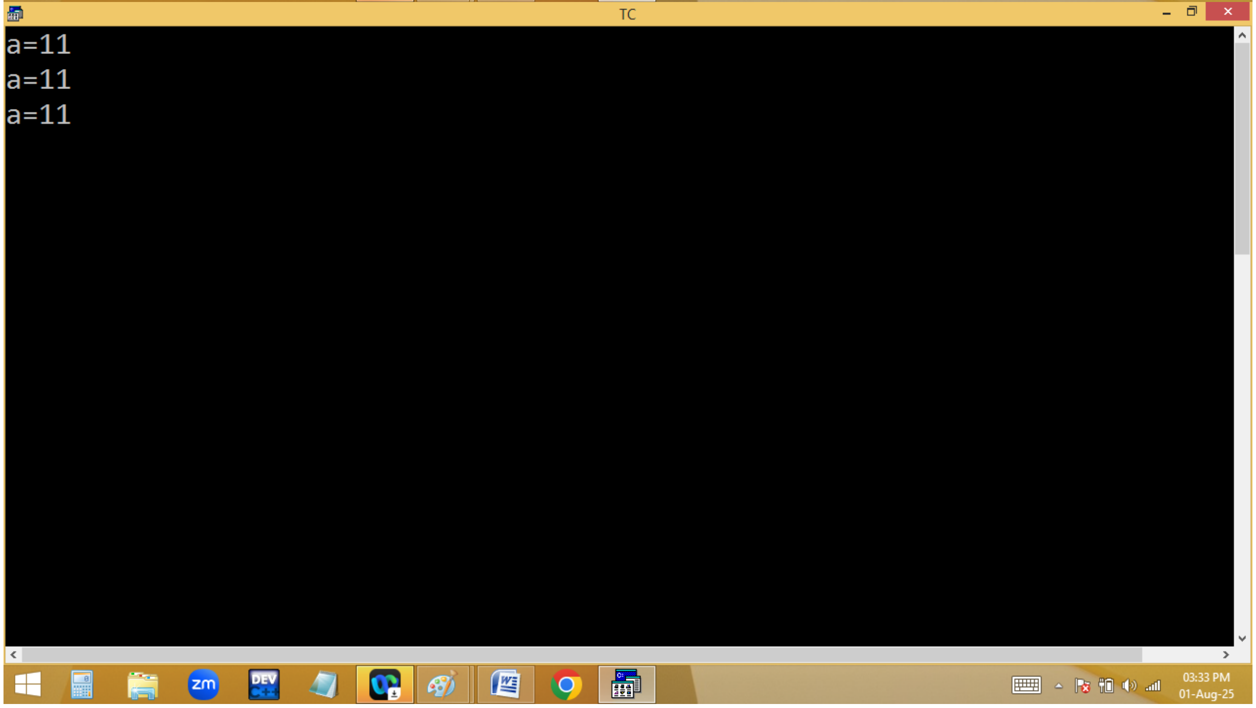


```
show method calling a=10
main method calling a=10
```

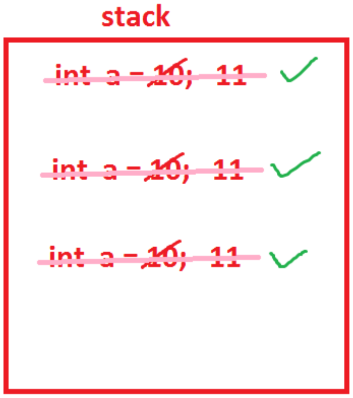
Finding life time of a local variable:

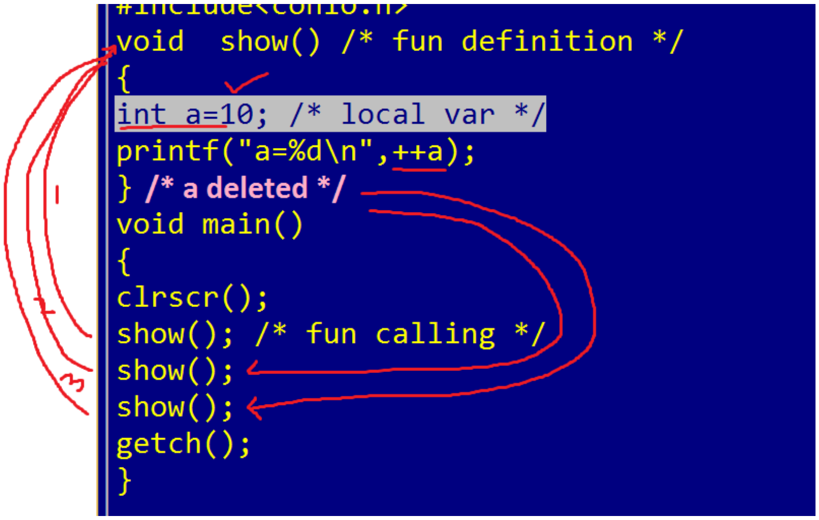


```
      Line 16    Col 1   Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void  show() /* fun definition */
{
int a=10; /* local var */
printf("a=%d\n",++a);
}
void main()
{
clrscr();
show(); /* fun calling */
show();
show();
getch();
}
```



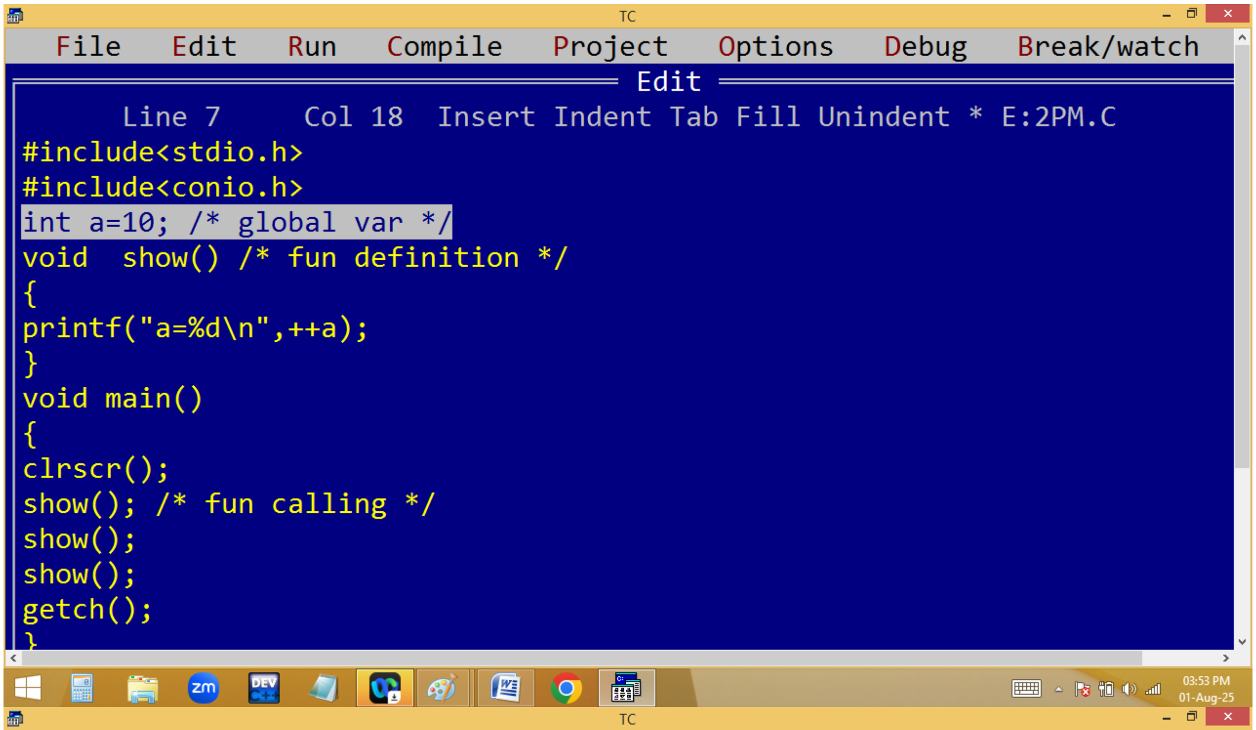
```
a=11
a=11
a=11
```



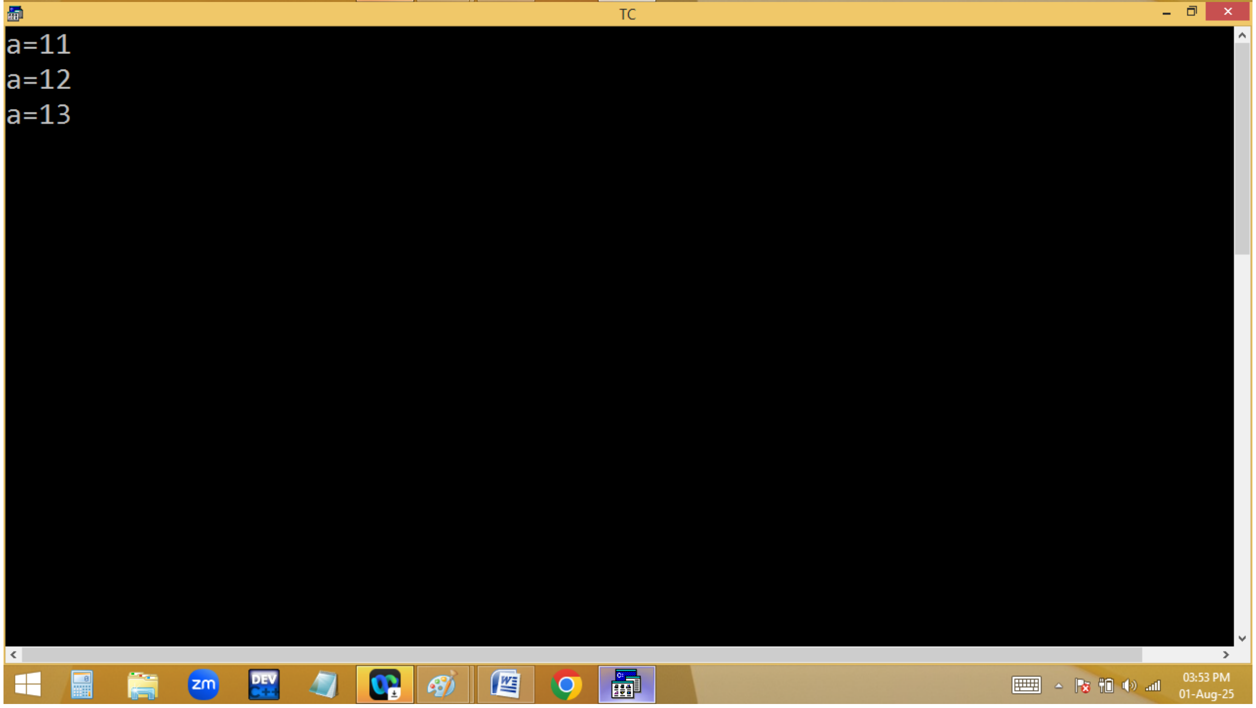


```
#include<conio.h>
void  show() /* fun definition */
{
int a=10; /* local var */
printf("a=%d\n",++a);
} /* a deleted */
void main()
{
clrscr();
show(); /* fun calling */
show();
show();
getch();
}
```

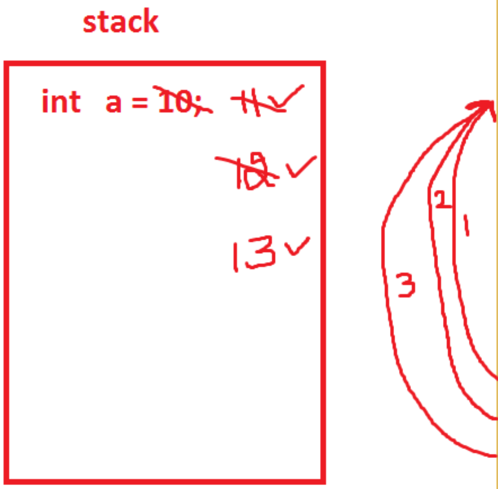
Finding global variable life time:

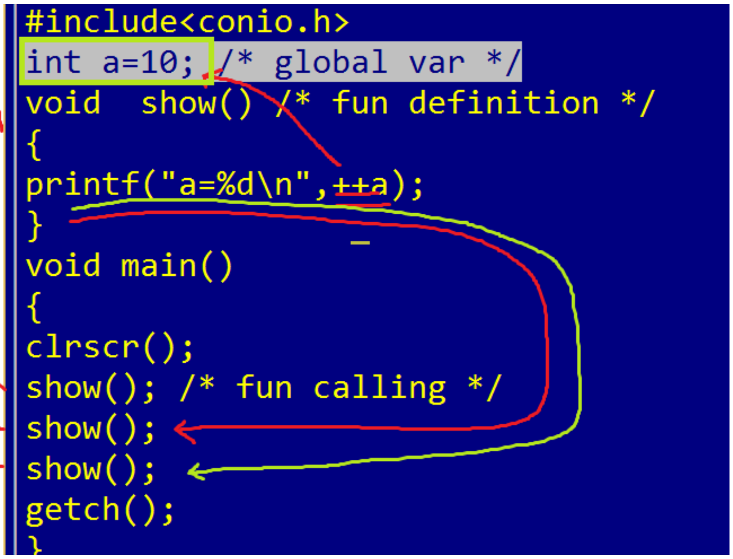

TC
File Edit Run Compile Project Options Debug Break/watch
Edit
Line 7 Col 18 Insert Indent Tab Fill Unindent * E:2PM.C
03:53 PM
01-Aug-25

```
#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();
}
```



```
a=11
a=12
a=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();
}
```