

VARIABLES

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

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

All the variable created in primary / temporary memory i.e. RAM only. Due to this after program / function execution automatically the variables are deleted from memory.

In c compiler we should have to declare the variables at the first line of any function. In c++ we can declare anywhere.

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

```
int a=100;
```

```
int A=200;
```

when the variable is defined then only memory allocated.

Eg.

```
int a; /* declaration / declared */
```

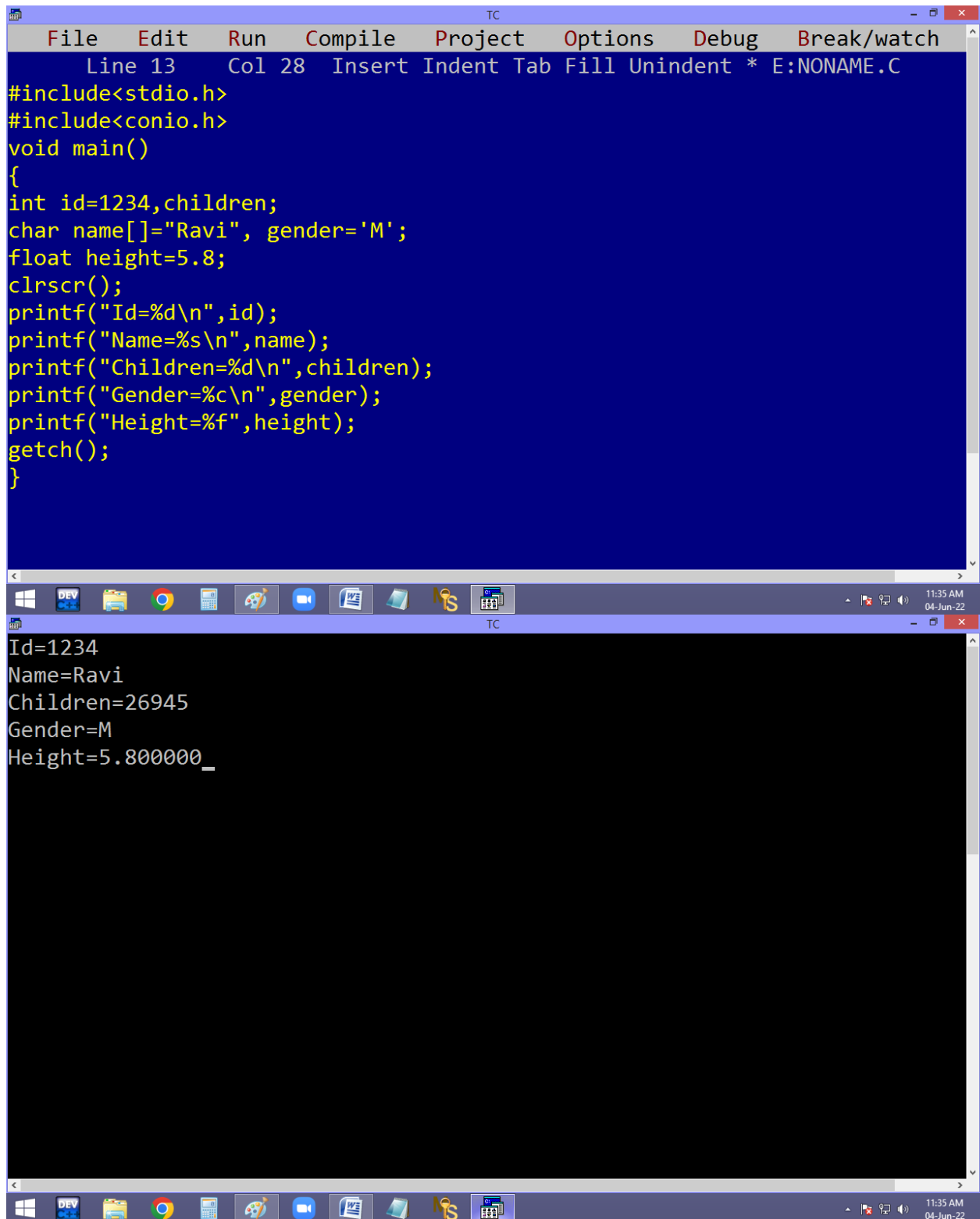
```
a=100; /* initialization or defined */
```

syntax: datatype **variable**[=value], **variable**[=value],....;

eg. int id=1234, children=2;

```
char name[ ]="Ravi", gender='M';
```

```
float height = 5.8;
```



The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 13 Col 28 Insert Indent Tab Fill Unindent * E:NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
int id=1234,children;
char name[]="Ravi", gender='M';
float height=5.8;
clrscr();
printf("Id=%d\n",id);
printf("Name=%s\n",name);
printf("Children=%d\n",children);
printf("Gender=%c\n",gender);
printf("Height=%f",height);
getch();
}
```

The bottom window shows the output of the program:

```
Id=1234
Name=Ravi
Children=26945
Gender=M
Height=5.800000_
```

The Windows taskbar at the bottom shows the time as 11:35 AM on 04-Jun-22.

Memory allocation for variables:

RAM - STACK
64KB - 65536 BYTES



65535
65534
.
.
.
.
.
.
'
1
0

1KB = 1024

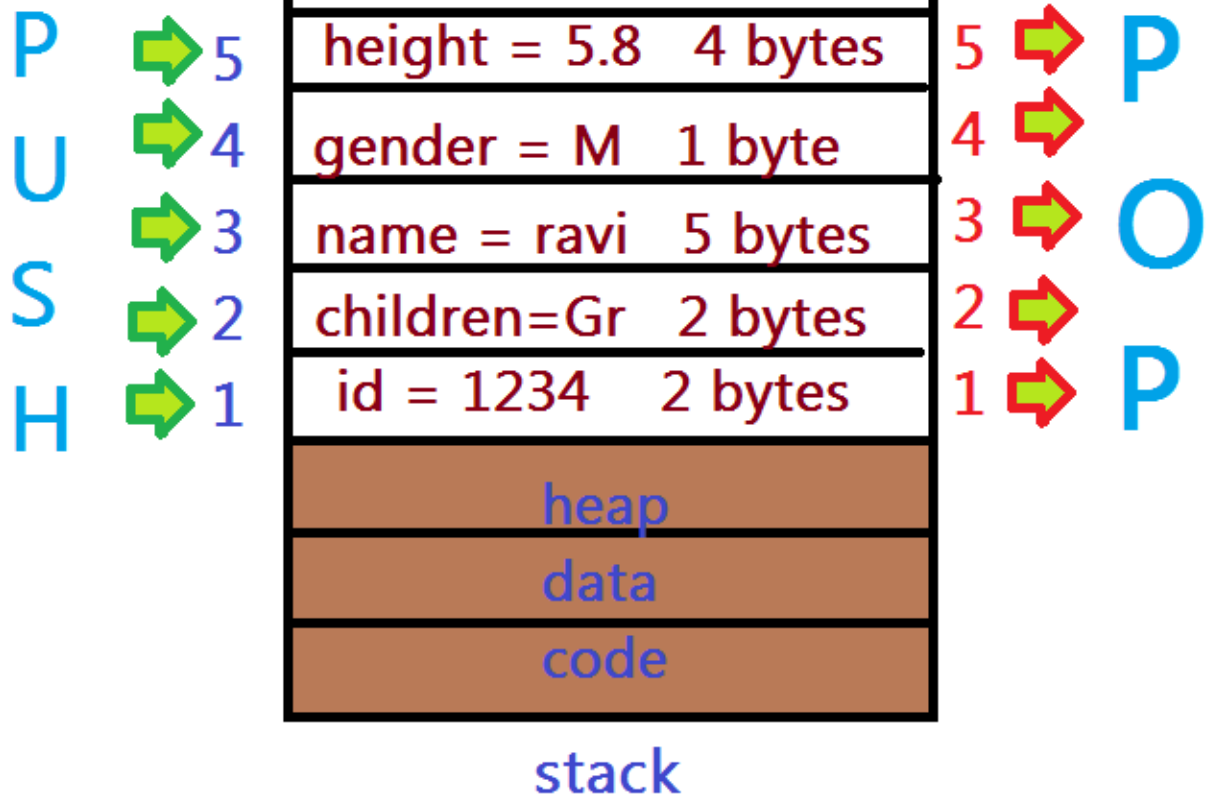
64KB = 1024 * 64 = 65536

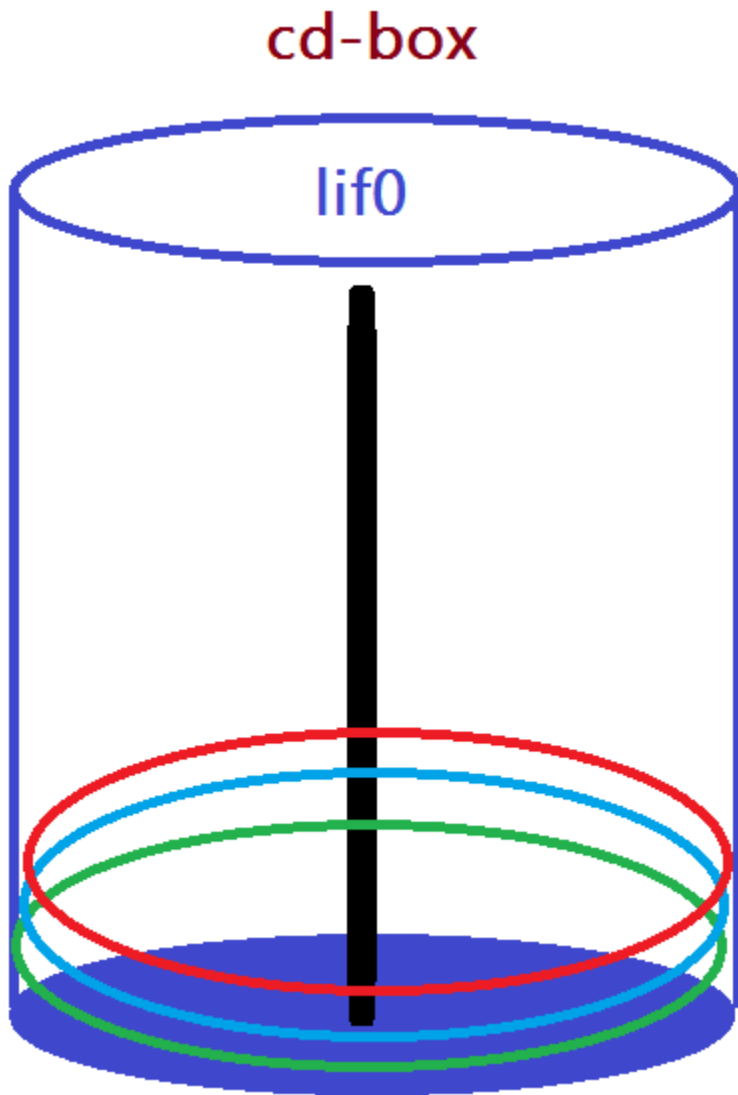
BYTE ADDRESS

unsigned int [%u]

Last In First Out - LIFO

First In Last Out - FILO





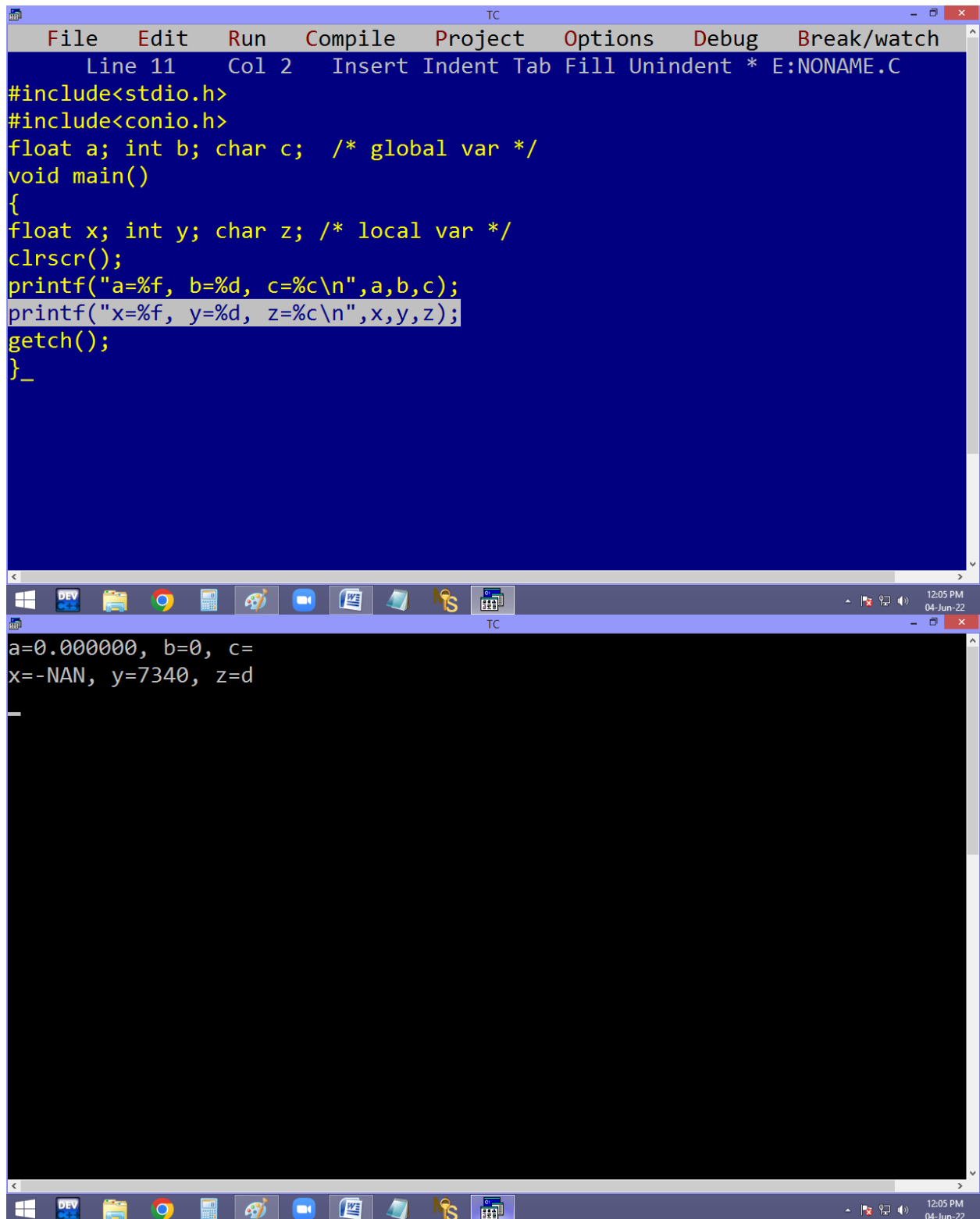
We can declare two types of variables.

1. Local variables
2. Global variables

	LOCAL VARAIABLES	GLOBAL VARIABLES
--	-------------------------	-----------------------------

declaration	Within fun / { } block	Top of program
Storage area	Stack area	Data area
Initial values	Garbage values	Int-0,float-0.000000, char-blank space
Scope-where we can access	Within fun / { }	Total program
Life time – active in memory	Until fun execution	Until total program execution

Eg. finding initial values of variables.



The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the source code editor, displaying a C program. The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. The status bar at the bottom of the editor shows 'Line 11 Col 2 Insert Indent Tab Fill Unindent * E:NONAME.C'. The code in the editor is as follows:

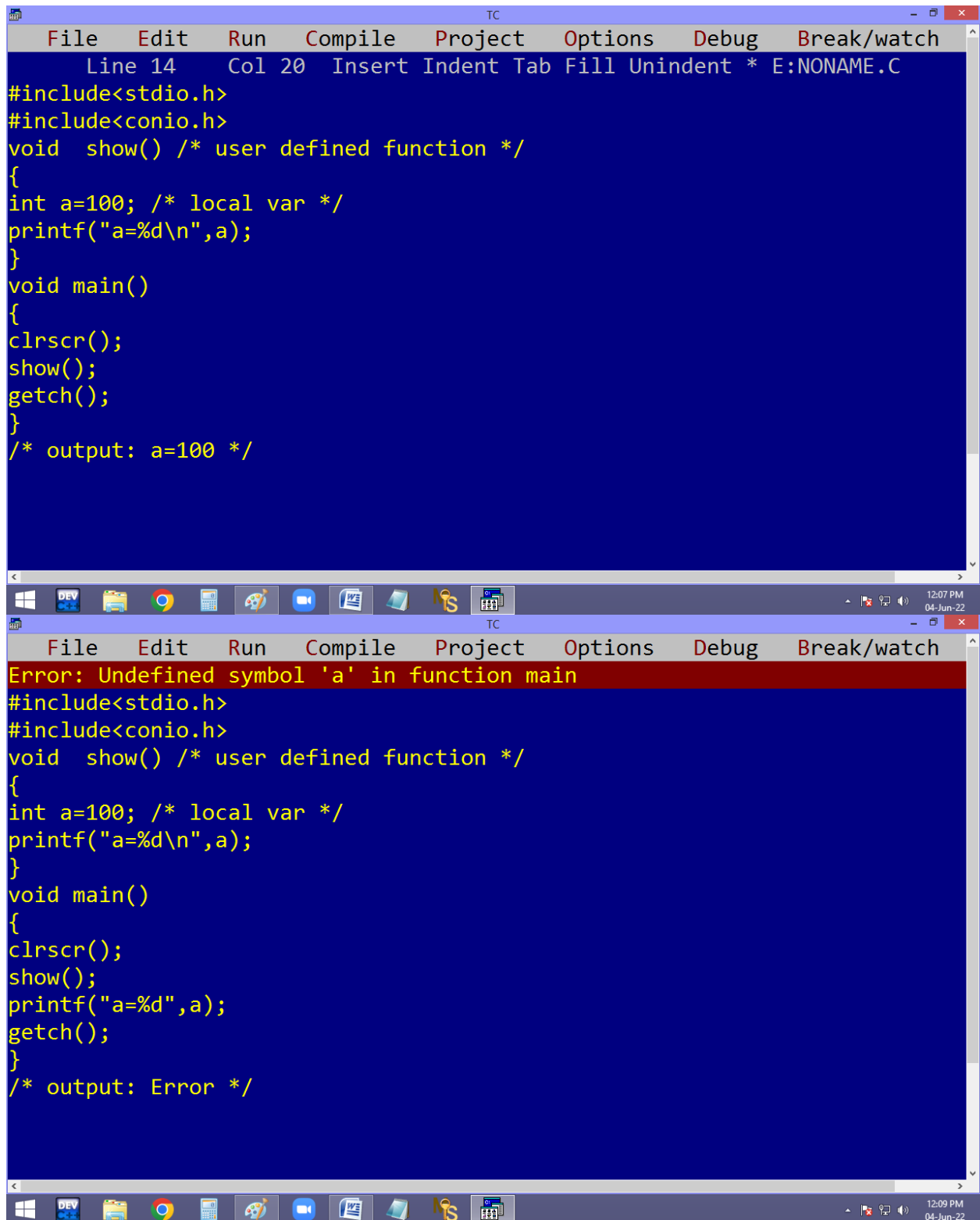
```
#include<stdio.h>
#include<conio.h>
float a; int b; char c; /* global var */
void main()
{
float x; int y; char z; /* local var */
clrscr();
printf("a=%f, b=%d, c=%c\n",a,b,c);
printf("x=%f, y=%d, z=%c\n",x,y,z);
getch();
}_
```

The bottom window is the output console, which displays the results of the program's execution:

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

The output shows that the first printf statement executed correctly, displaying 'a=0.000000, b=0, c=' followed by a newline. The second printf statement also executed, displaying 'x=-NAN, y=7340, z=d' followed by a newline. The console window has a black background and white text.

Eg. Finding scope of local variables.



The image displays two screenshots of the Turbo C++ (TC) IDE, illustrating a common programming error related to variable scope.

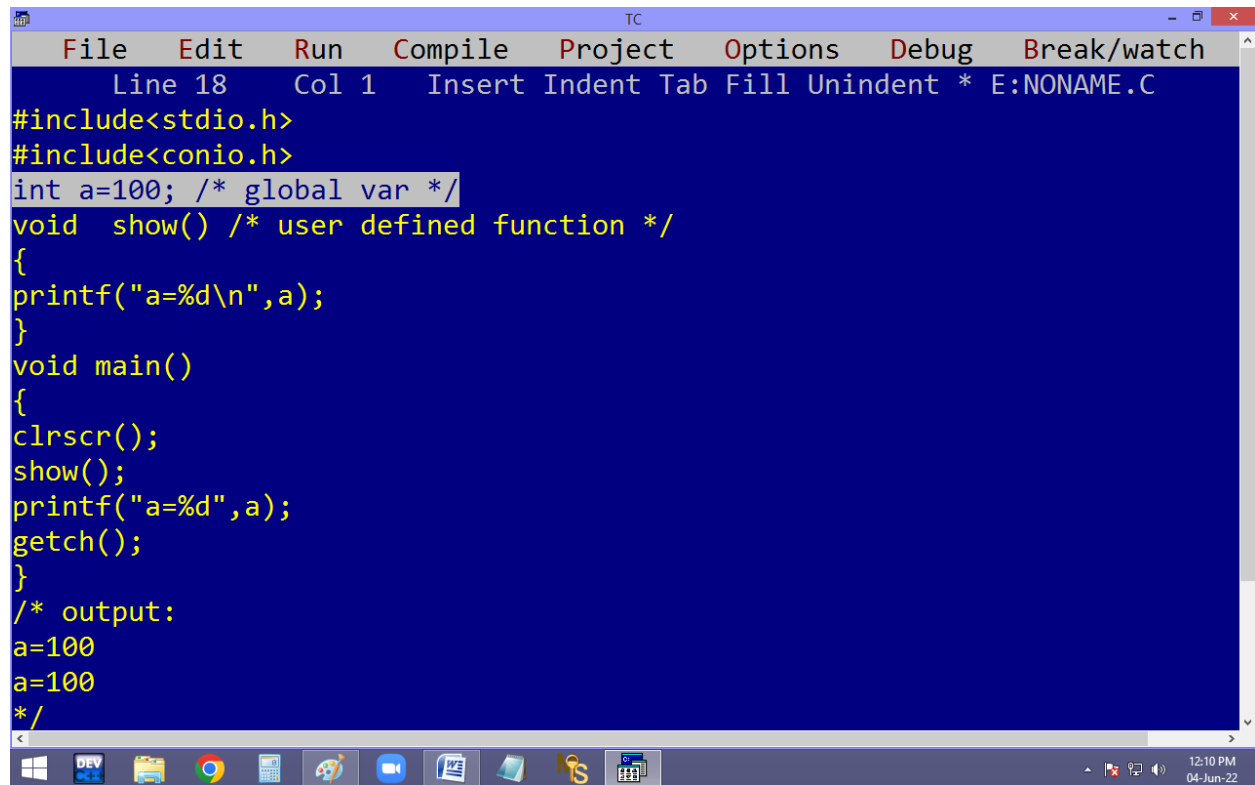
Top Screenshot: The TC window shows a C program named `E:NONAME.C`. The code defines a function `show()` and a `main()` function. In `show()`, a local variable `a` is declared and assigned the value 100. In `main()`, the `show()` function is called. The code is as follows:

```
Line 14 Col 20 Insert Indent Tab Fill Unindent * E:NONAME.C
#include<stdio.h>
#include<conio.h>
void show() /* user defined function */
{
int a=100; /* local var */
printf("a=%d\n",a);
}
void main()
{
clrscr();
show();
getch();
}
/* output: a=100 */
```

Bottom Screenshot: The same TC window shows the result of compiling the code. A red error message is displayed at the top: **Error: Undefined symbol 'a' in function main**. This occurs because the variable `a` is only defined within the scope of the `show()` function but is used in the `main()` function without being declared there. The code in the editor is identical to the top screenshot, but the `main()` function now includes a call to `printf("a=%d",a);` instead of `show();`.

```
File Edit Run Compile Project Options Debug Break/watch
Error: Undefined symbol 'a' in function main
#include<stdio.h>
#include<conio.h>
void show() /* user defined function */
{
int a=100; /* local var */
printf("a=%d\n",a);
}
void main()
{
clrscr();
show();
printf("a=%d",a);
getch();
}
/* output: Error */
```

Finding scope of global variable:



The screenshot shows the Turbo C++ (TC) IDE with a C program open in the editor. The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. The status bar at the top indicates 'Line 18 Col 1 Insert Indent Tab Fill Unindent * E:NONAME.C'. The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
int a=100; /* global var */
void show() /* user defined function */
{
printf("a=%d\n",a);
}
void main()
{
clrscr();
show();
printf("a=%d",a);
getch();
}
/* output:
a=100
a=100
*/
```

The Windows taskbar at the bottom shows various application icons, including DEV, File Explorer, Chrome, Calculator, Paint, and others. The system clock in the bottom right corner displays '12:10 PM 04-Jun-22'.

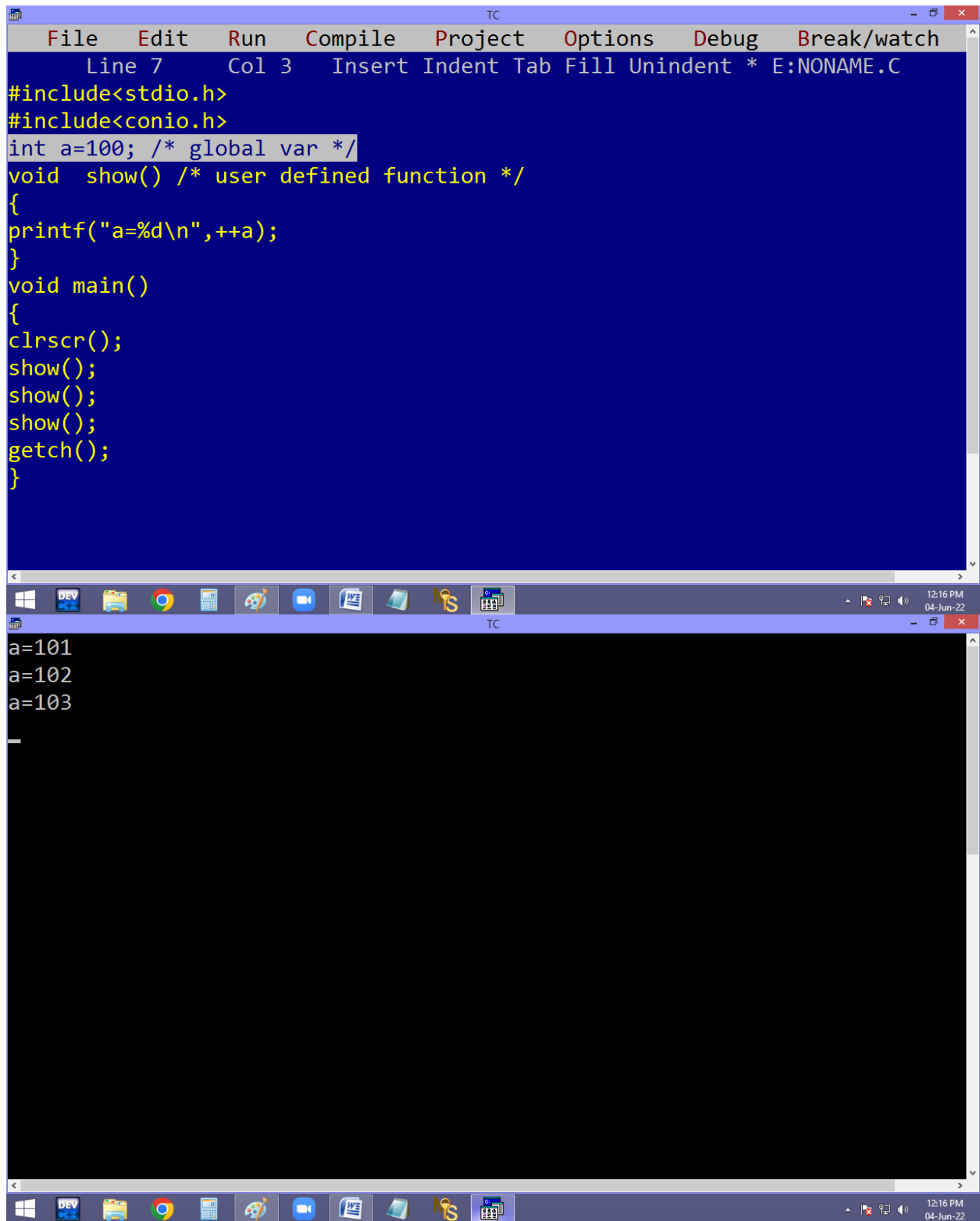
Eg. finding lifetime of local variable.

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 18 Insert Indent Tab Fill Unindent * E:NONAME.C
#include<stdio.h>
#include<conio.h>

void show() /* user defined function */
{
int a=100; /* local var */
printf("a=%d\n",++a);
} /* a deleted */_
void main()
{
clrscr();
show();
show();
show();
getch();
}

a=101
a=101
a=101
_
```

Eg. finding life time of global variable:



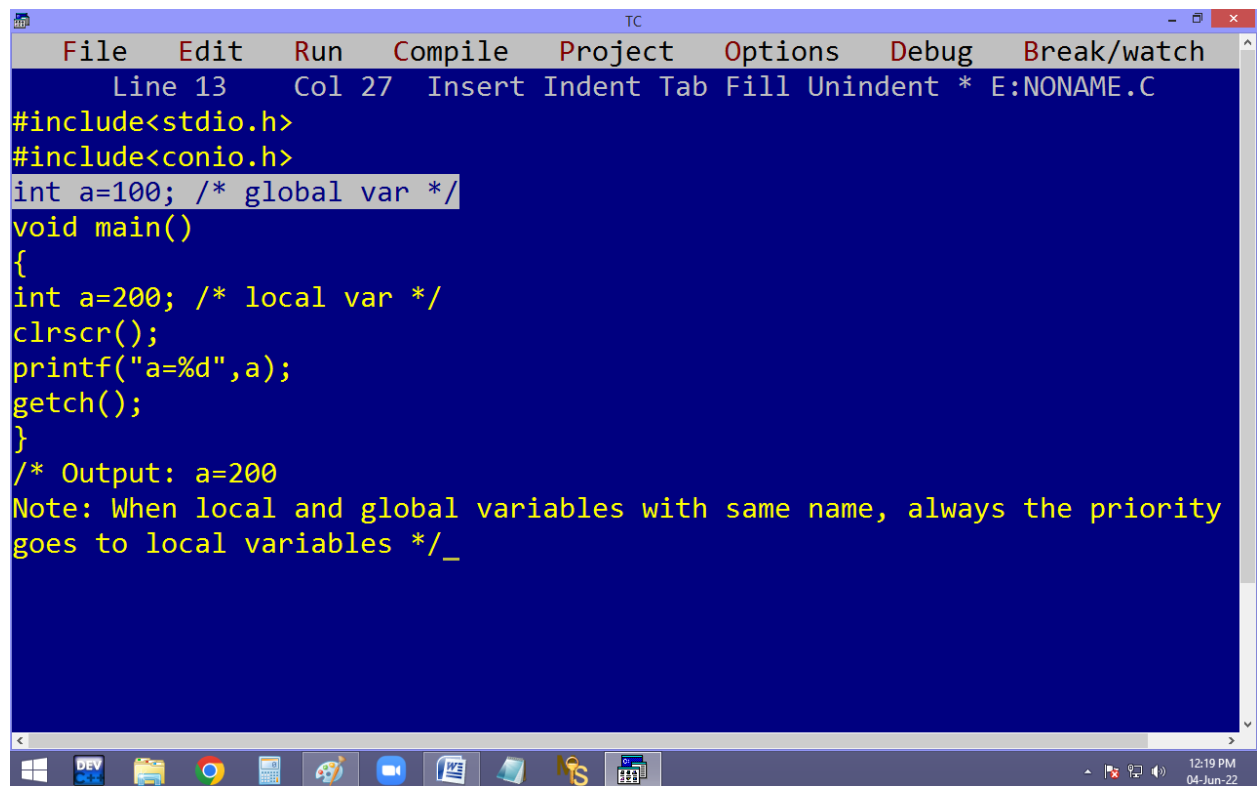
The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 7 Col 3 Insert Indent Tab Fill Unindent * E:NONAME.C
#include<stdio.h>
#include<conio.h>
int a=100; /* global var */
void show() /* user defined function */
{
printf("a=%d\n",++a);
}
void main()
{
clrscr();
show();
show();
show();
getch();
}
```

The bottom window shows the output of the program:

```
a=101
a=102
a=103
_
```

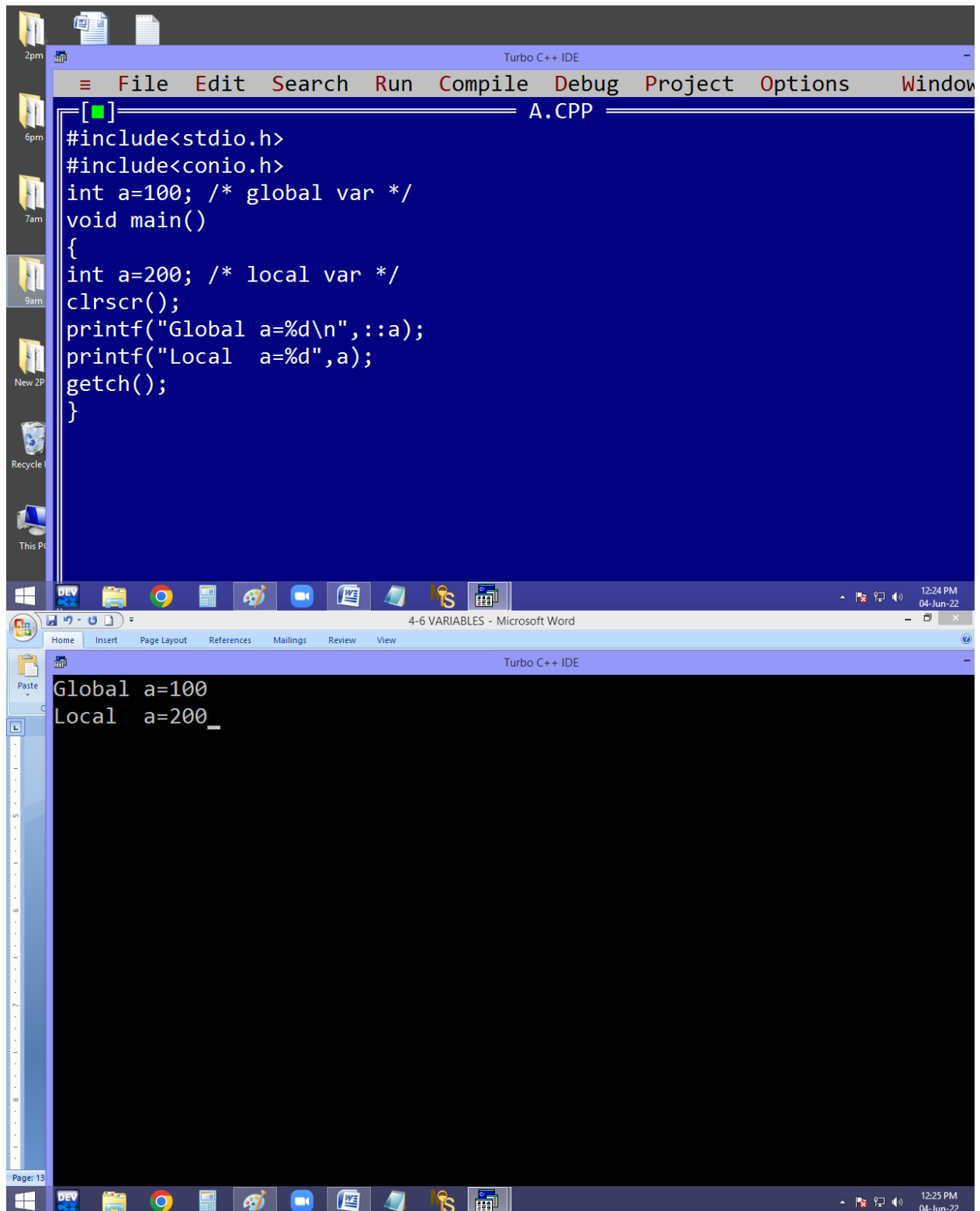
The Windows taskbar at the bottom shows the time as 12:16 PM on 04-Jun-22.



The image shows a screenshot of a Turbo C++ (TC) IDE window. The title bar reads "TC". The menu bar includes "File", "Edit", "Run", "Compile", "Project", "Options", "Debug", and "Break/watch". The status bar at the top indicates "Line 13", "Col 27", and "Insert Indent Tab Fill Unindent * E:NONAME.C". The main text area has a dark blue background with yellow text. The code is as follows:

```
#include<stdio.h>
#include<conio.h>
int a=100; /* global var */
void main()
{
    int a=200; /* local var */
    clrscr();
    printf("a=%d",a);
    getch();
}
/* Output: a=200
Note: When local and global variables with same name, always the priority
goes to local variables */_
```

The Windows taskbar is visible at the bottom, showing icons for Windows, DEV C++, File Explorer, Google Chrome, Calculator, Paint, VS Code, and a folder named "NS". The system clock in the bottom right corner shows "12:19 PM" and "04-Jun-22".



TC

File Edit Run Compile Project Options Debug Break/watch

Error: Redeclaration of 'a'

```
#include<stdio.h>
#include<conio.h>
int a=100; /* global var */
int a=500;
void main()
{
int a=200; /* local var */
clrscr();
printf("a=%d",a);
getch();
}
```

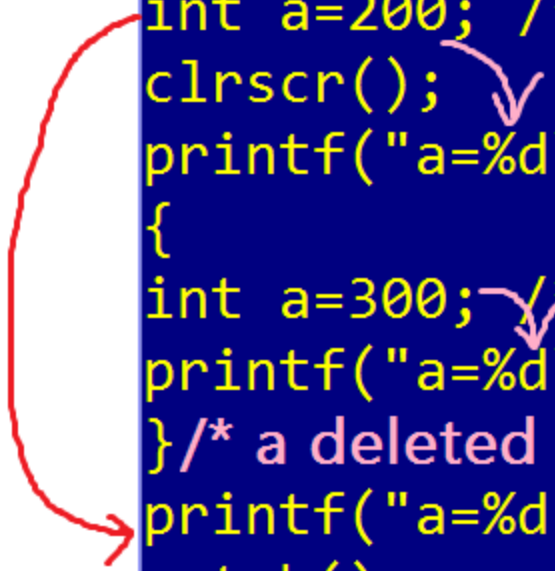
TC

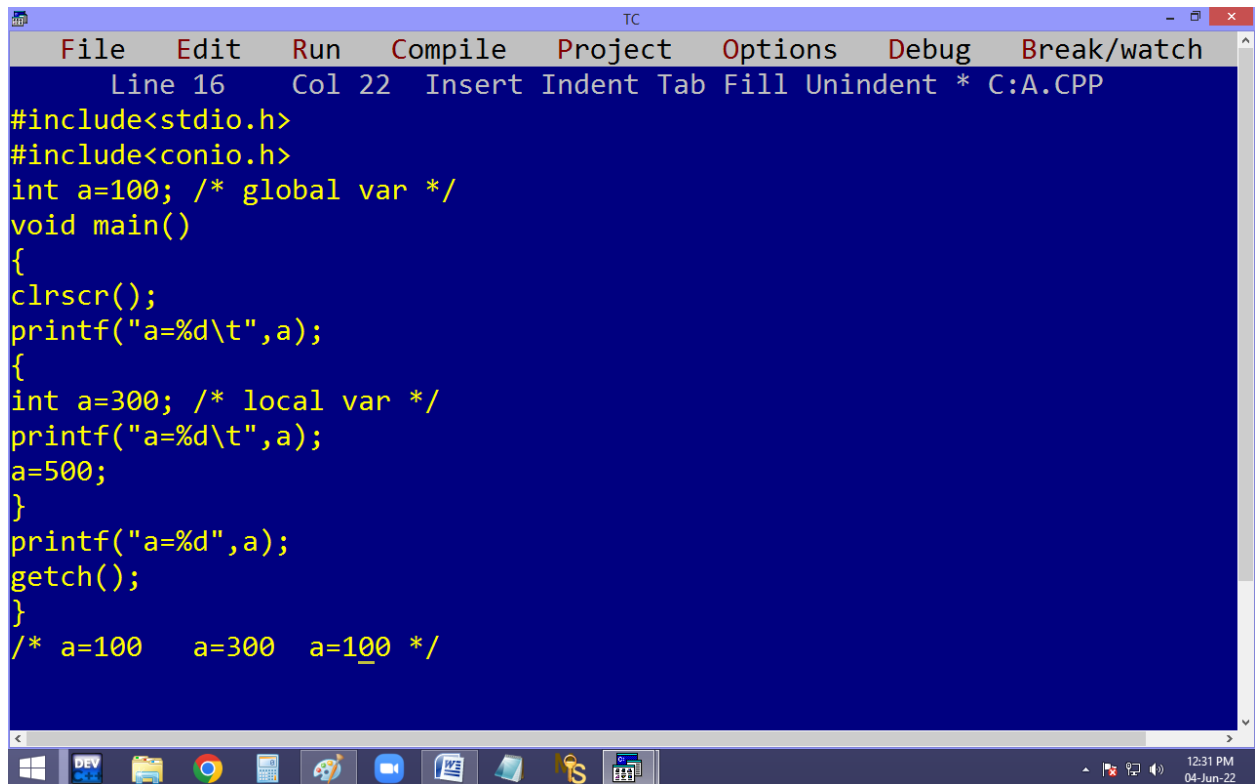
File Edit Run Compile Project Options Debug Break/watch

Line 16 Col 27 Insert Indent Tab Fill Unindent * C:A.CPP

```
#include<stdio.h>
#include<conio.h>
int a=100; /* global var */
void main()
{
int a=200; /* local var */
clrscr();
printf("a=%d\t",a);
{
int a=300; /* local var */
printf("a=%d\t",a);
}
printf("a=%d",a);
getch();
}
/* a=200    a=300    a=200 */
```

```
int a=100; /* global var */
void main()
{
int a=200; /* local var */
clrscr();
printf("a=%d\t",a); 200
{
int a=300; /* local var */
printf("a=%d\t",a); 300
}/* a deleted */
printf("a=%d",a); 200
getch();
}
/* a=200    a=300    a=200 */
```





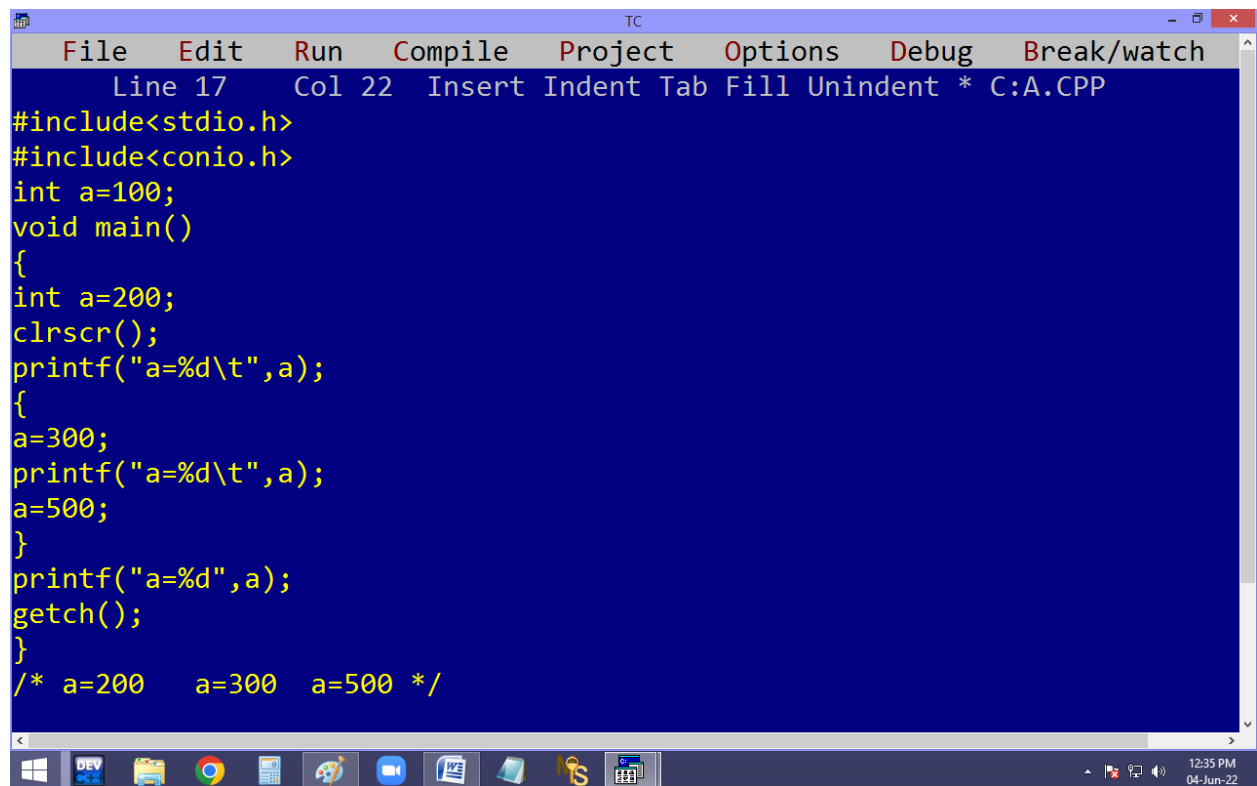
The image shows a screenshot of a Turbo C++ (TC) IDE window. The title bar reads "TC". The menu bar includes "File", "Edit", "Run", "Compile", "Project", "Options", "Debug", and "Break/watch". The status bar at the top indicates "Line 16", "Col 22", and "Insert Indent Tab Fill Unindent * C:A.CPP". The main editing area has a dark blue background with yellow text. The code is as follows:

```
#include<stdio.h>
#include<conio.h>
int a=100; /* global var */
void main()
{
    clrscr();
    printf("a=%d\t",a);
    {
        int a=300; /* local var */
        printf("a=%d\t",a);
        a=500;
    }
    printf("a=%d",a);
    getch();
}
/* a=100    a=300    a=100 */
```

The Windows taskbar is visible at the bottom, showing icons for Windows, DEV C++, File Explorer, Google Chrome, Calculator, Paint, VLC, Telegram, Notepad, and a folder icon. The system clock in the bottom right corner shows "12:31 PM" and "04-Jun-22".

```
#include<conio.h>
int a=100; /* global var */
void main()
{
clrscr();
printf("a=%d\t",a); 100
{
int a=300; /* local var */
printf("a=%d\t",a);
a=500; 300
}
printf("a=%d",a);
getch(); 100
}
/* a=100    a=300    a=100 */
```

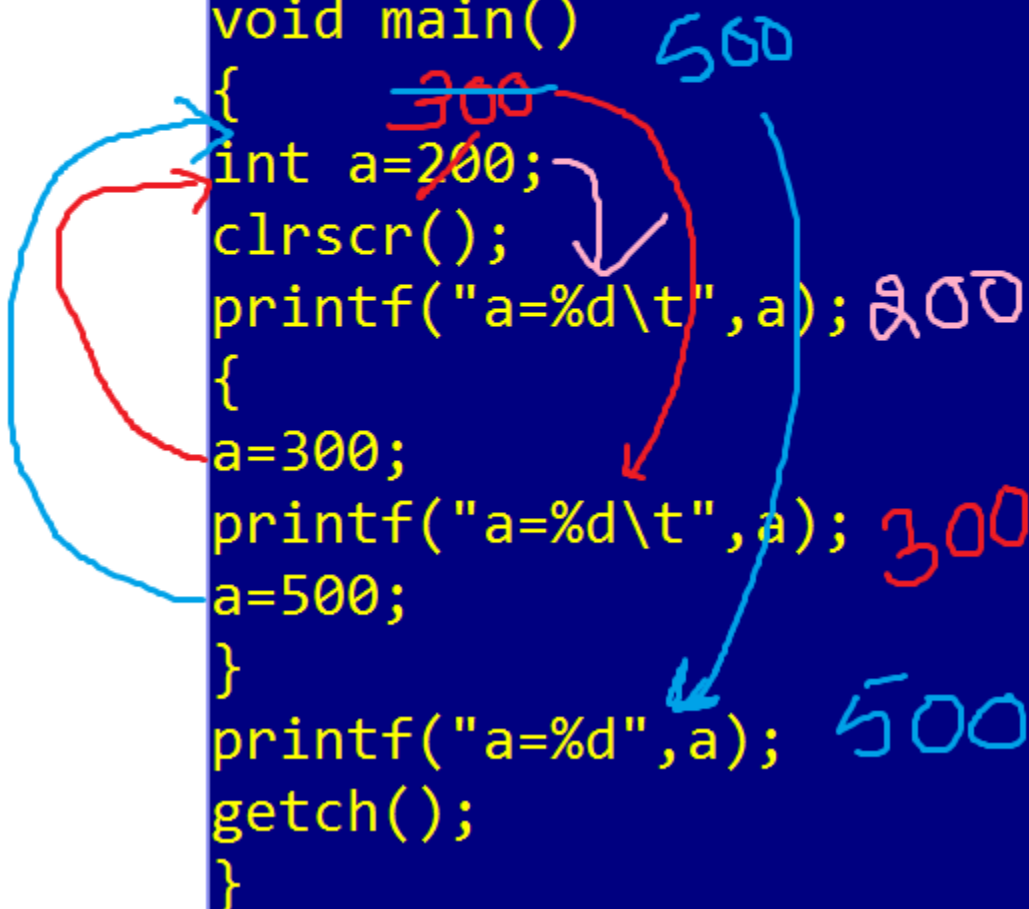
The diagram illustrates the scope resolution for the variable 'a'. A red arrow originates from the 'a' in the first printf statement and points to the global variable 'a=100'. Another red arrow originates from the 'a' in the second printf statement and points to the local variable 'a=300'. A third red arrow originates from the 'a' in the final printf statement and points to the global variable 'a=100'.



```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 17 Col 22 Insert Indent Tab Fill Unindent * C:A.CPP
#include<stdio.h>
#include<conio.h>
int a=100;
void main()
{
int a=200;
clrscr();
printf("a=%d\t",a);
{
a=300;
printf("a=%d\t",a);
a=500;
}
printf("a=%d",a);
getch();
}
/* a=200 a=300 a=500 */
```

Line 17 Col 22 Ir

```
#include<stdio.h>
#include<conio.h>
int a=100;
void main()
{
int a=200;
clrscr();
printf("a=%d\t",a);
{
a=300;
printf("a=%d\t",a);
a=500;
}
printf("a=%d",a);
getch();
}
```



TC

File Edit Run Compile Project Options Debug Break/watch

Error: Undefined symbol 'c' in function main

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10,b=20;
clrscr();
{
int c=30;
printf("sum=%d\t",a+b+c);
c=50;
}
printf("sum=%d",a+b+c);
getch();
}
/* Error */
```

TC

File Edit Run Compile Project Options Debug Break/watch

Line 16 Col 19 Insert Indent Tab Fill Unindent * C:A.CPP

```
#include<stdio.h>
#include<conio.h>
int a=1,b=2,c;
void main()
{
int a=10,b=20;
clrscr();
{
int c;
printf("sum=%d\t",a+b+c);
c=100;
}
printf("sum=%d",a+b+c);
getch();
}
/* Sum=Gr Sum=30 */
```

```
int a=1,b=2,c;  
void main()  
{  
  int a=10,b=20;  
  clrscr();  
  {  
    int c;  
    printf("sum=%d\t",a+b+c);  
    c=100;  
  }  
  printf("sum=%d",a+b+c);  
  getch();  
}  
/* Sum=Gr    Sum=30 */
```

10+20+gr=gr

10+20+0=30