Let’s solve it

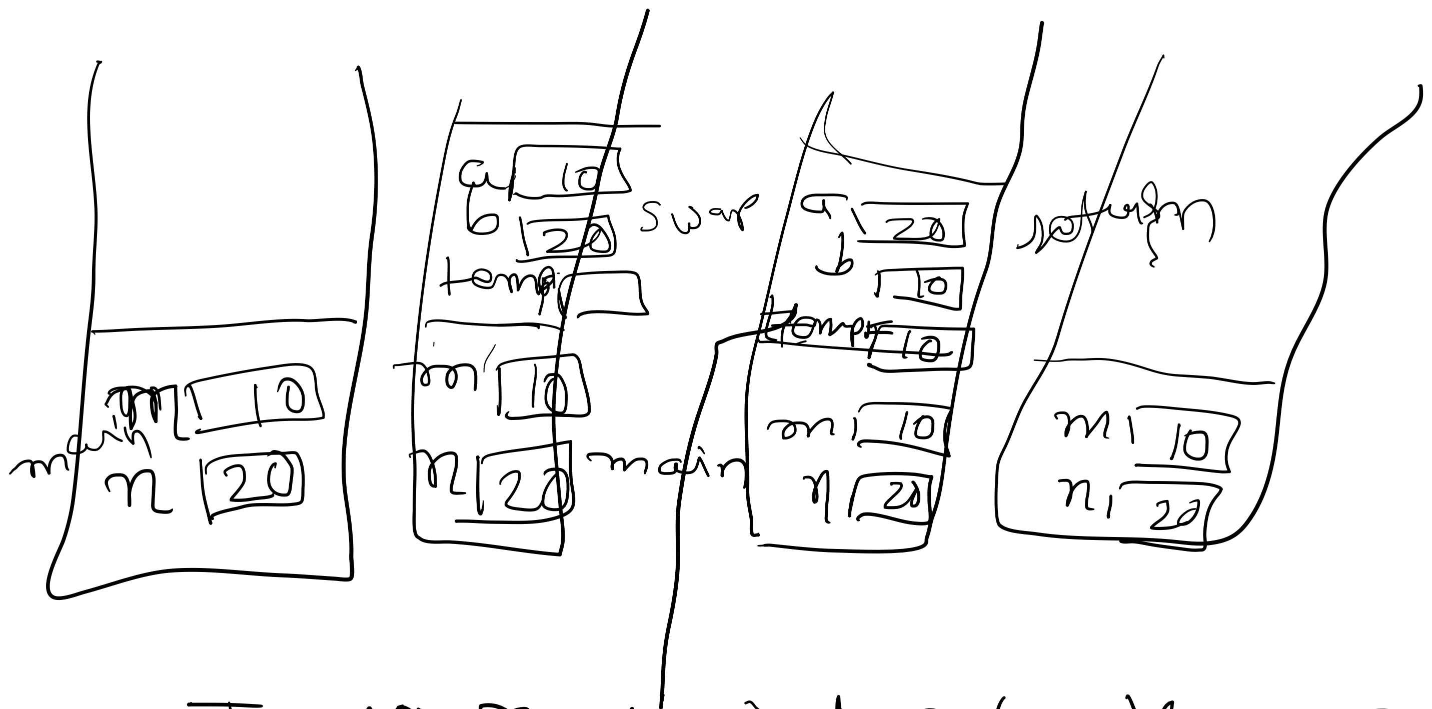
29

## Storage Classes

- Local / automatic
- global
- static
  - local
  - global
- extern

```
void swap(int a, int b);
int main()
 {
      int m =10, n=20;
       swap( m, n);

 }
void swap(int a, int b)
 {  int temp;
    temp = a;
     a=b;
     b = temp;
  }
```



If you may print a,~~b they are~~
20,10
But upon return they get
destroyed. Originally main m, n
still ~~displays~~ 10 and 20 only.

Here, m and n are local to main while a, b and temp are local to swap function.

Local variables are automatically getting memory when function execution starts and destroys when function finishes or executes 'return' keyword.

Note that their scope ~~is~~ within the f^n and lifetime too.

```
int main()
{
  for(int i=0; i<10; i++)
  {    if (~~a[i] < 0~~)
            break;
  }
  if (i==10)
  {      printf("All positives");
  }
  else
   {  printf("Negative present");
   }
}
```

Note that i variable has scope within loop block only.

Not accessible outside.

Correction

```
int i;
for(i=0; ...
{ ...
  ...
}
```

## Global variable

declared/defined outside all functions.

It get memory when program starts and destroys ~~when progr~~am finishes.

- Accessible by all functions and hence very much like public property. Not widely used. less secure.

Mostly, in st~~ructured~~ programming data is passed back and forth between functions~~. to ex~~ecute proper logic.

This is also referred as message passing using global variables.

global variables lifetime is whole program ~~and also~~ scope/visibility is whole program( all files).

If we want, certain variable to be accessed within this file only, global ~~term~~ we have to make it

static int ~~e~~gglobal;

most global variables are initialized to zero by default. while local are garbage.

static autollocal to function

Imagine you want to find out number of visitors thus far or generate a sequence numbers, may use static locally.

They get memory and initialized once when created. Rest of time of 2nd function call onwards, the memory with data retained.

Scope/visibility is within the function where declared but livetime is from 1st $f^n$ call till end of Program

## extern keyword

you can use variable of file2.c into file1.c

define in file1.c globally without static.

redone it file2.c with extern keyword.

extern keyword tells compiler while compiling file2.c ~~that~~ so and so variable is ~~present at a program~~ level. You will get it when linked.