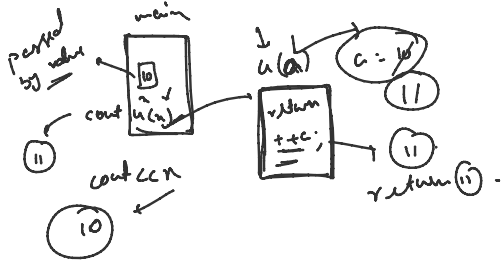
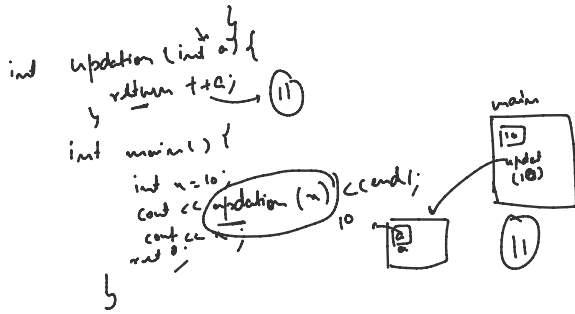
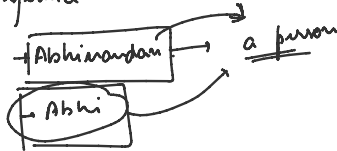


scope of variables in functions.

{ → this means a new scope.



reference means nickname in C++.



```

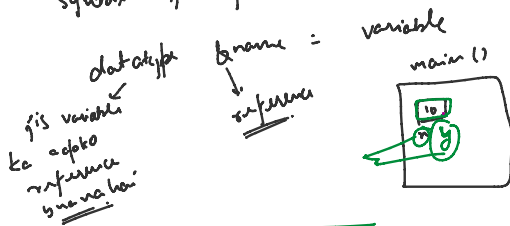
// x = updation(x)
updation2(x)
cout << x << endl;
    
```

→ changes reflect.

```

int x = 10;
int &y = x;
    
```

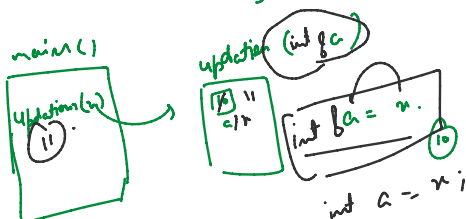
Syntax of reference.



updation2(ba)

```

{ return ++a; }
    
```



Call by reference:-

when you pass value to a, it does in that

call by reference:-

when you pass value to a function, then the changes it does in that value are also reflected inside the function from which it was called, then it is called pass/call by reference.

Arrays are always passed by reference in C++;

int x; → declaration
x = 10; → definition.
int x = 10; initialization



declaration
declaration of a function → return type ↓ function name (parameters)
↓
compiler doesn't provide any memory to function

return type function name (parameters)
{
 body
}
↓
definition

I can declare a function any no. of times but I can define it only once.

All functions should be either declared or declared & defined above main function, then only they will ever get executed.

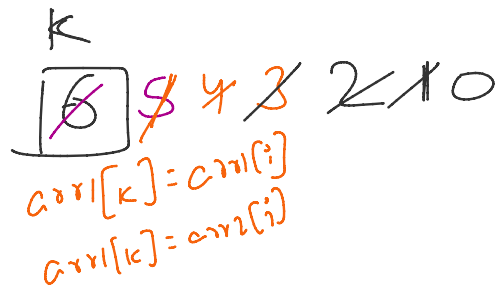
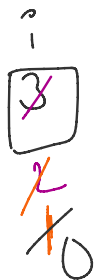
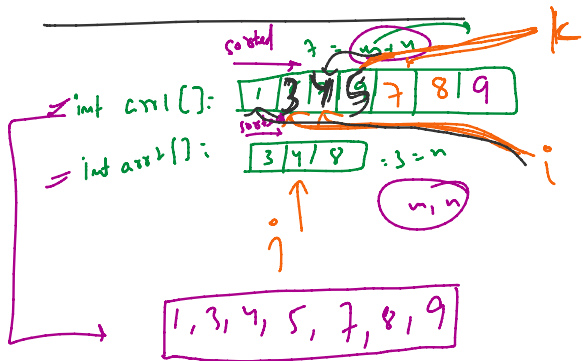
invoking / calling of a function

if function has r.t of void.
functionname (parameters);

else
cout << func(param) << endl
→ value = func(param); → u.c.r

declared / defined
{
 A();
}

How to pass an array
 → just like normal variables.
 # also pass their size with them
 in a separate variable.



7

	$arr2[i]$		$arr1[i]$
1	8	<	9
	8	>	7
	4	<	7
	4	<	5
	4	>	1
	3	>	1

comparisons
 6