Short-term Hands-on Supplementary Course on C programming

Session 9:
Pointers

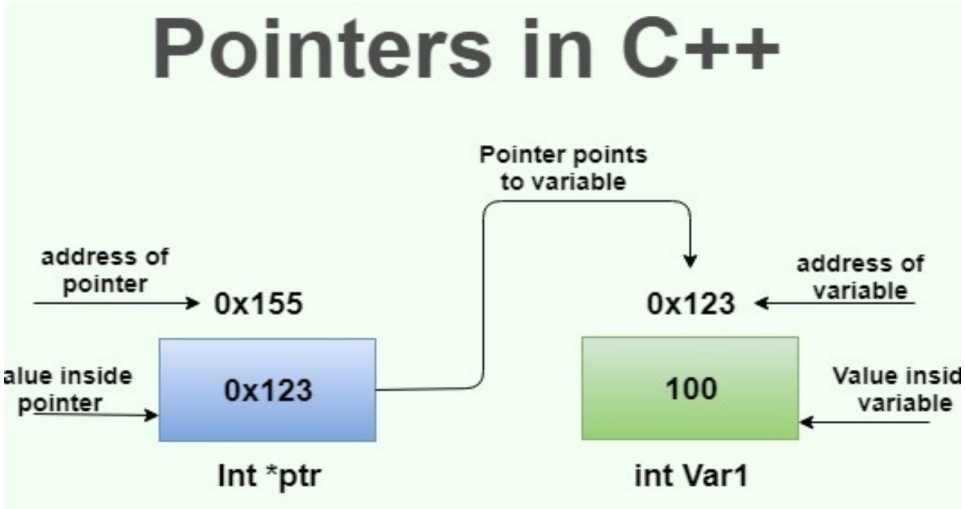
Nandakishor. V
Nitheesh Kumar. N
Sanjay Rojar U

Agenda

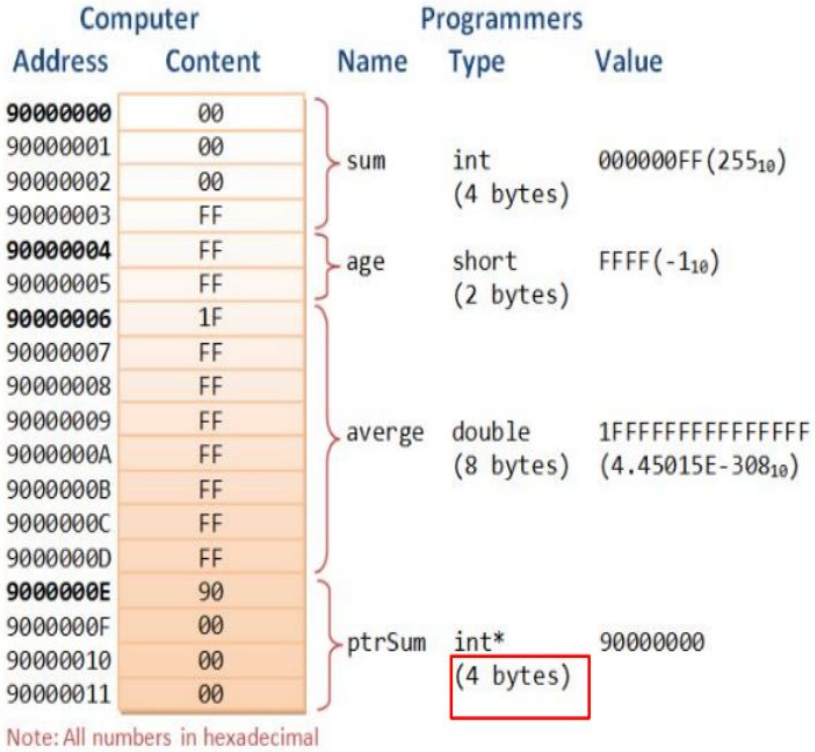
1. What are pointers ?
2. Declaring and using pointers
3. Pointer arithmetic
4. Double Pointers
5. Pointers and Arrays
6. Static vs Dynamic Memory Allocation
7. Dynamic Memory Allocation in C
    a. Primitive Types
    b. Arrays and Strings
    c. Functions
8. Tutorial: Arrays and Pointers

Pointers

Pointers in C language is **a variable which stores the address of another variable**.



Declaring and using Pointers



Computer		Programmers		
Address	Content	Name	Type	Value
90000000	00			
90000001	00	sum	int (4 bytes)	000000FF(255_{10})
90000002	00			
90000003	FF			
90000004	FF	age	short (2 bytes)	FFFF(-1_{10})
90000005	FF			
90000006	1F	averge	double (8 bytes)	1FFFFFFFFFFFFFFF ($4.45015E-308_{10}$)
90000007	FF			
90000008	FF			
90000009	FF			
9000000A	FF			
9000000B	FF			
9000000C	FF			
9000000D	FF			
9000000E	90	ptrSum	int* (4 bytes)	90000000
9000000F	00			
90000010	00			
90000011	00			

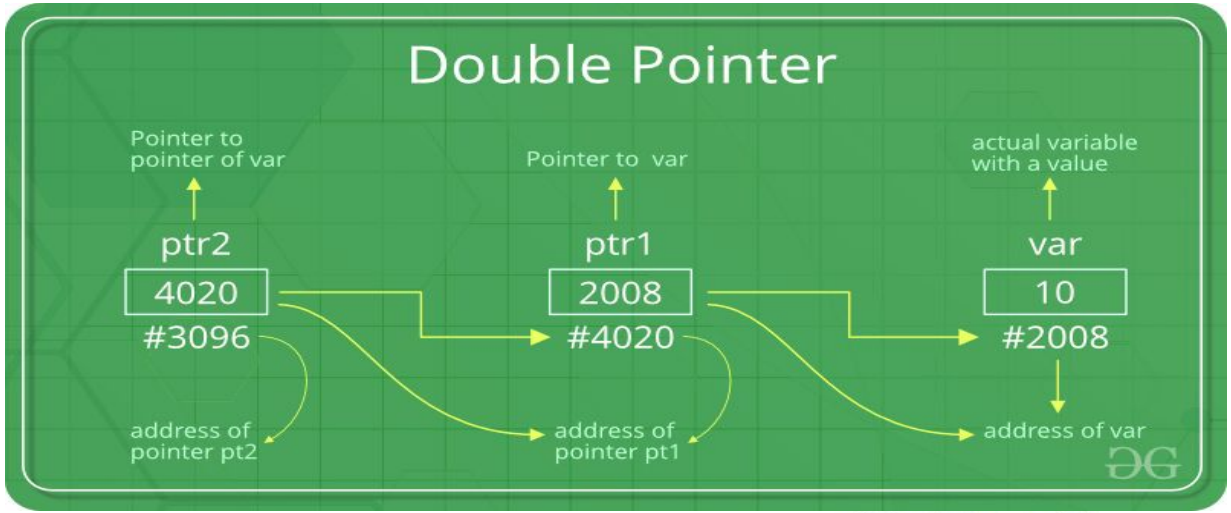
Note: All numbers in hexadecimal

```
type *ptr;
// or
type* ptr;
// or
type * ptr;
```

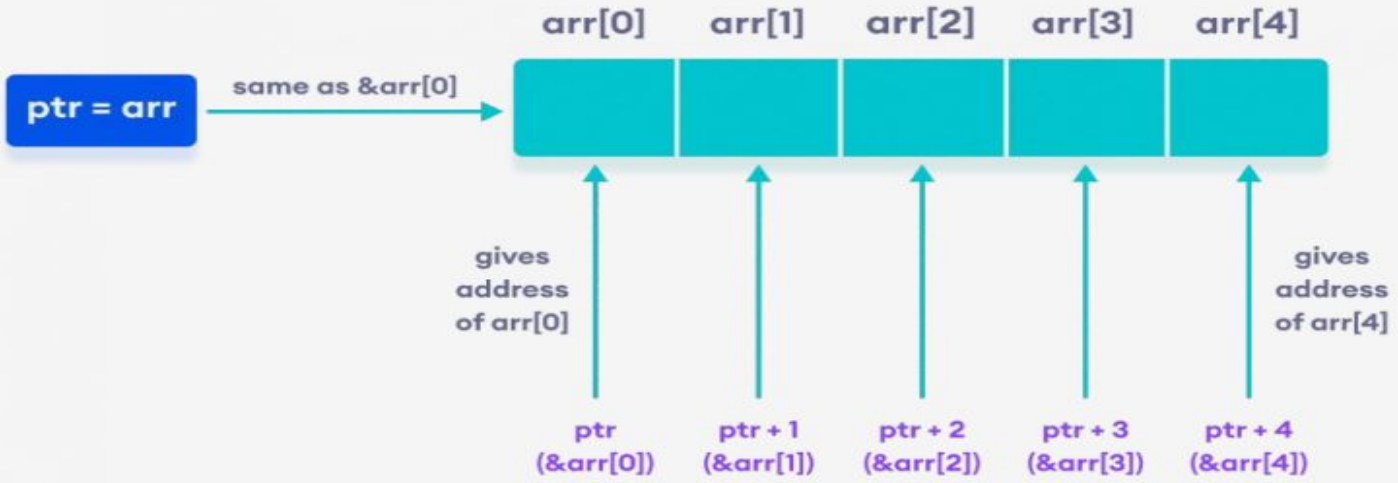
```c
#include <stdio.h>

int main(void) {
    int sum = 255;
    short age = -1;
    double average = 4.45015E-308;
    int* ptrSum = &sum;
}
```

Double Pointers

The **first pointer** is used to **store the address of the variable**. And the **second pointer** is used to **store the address of the first pointer.**

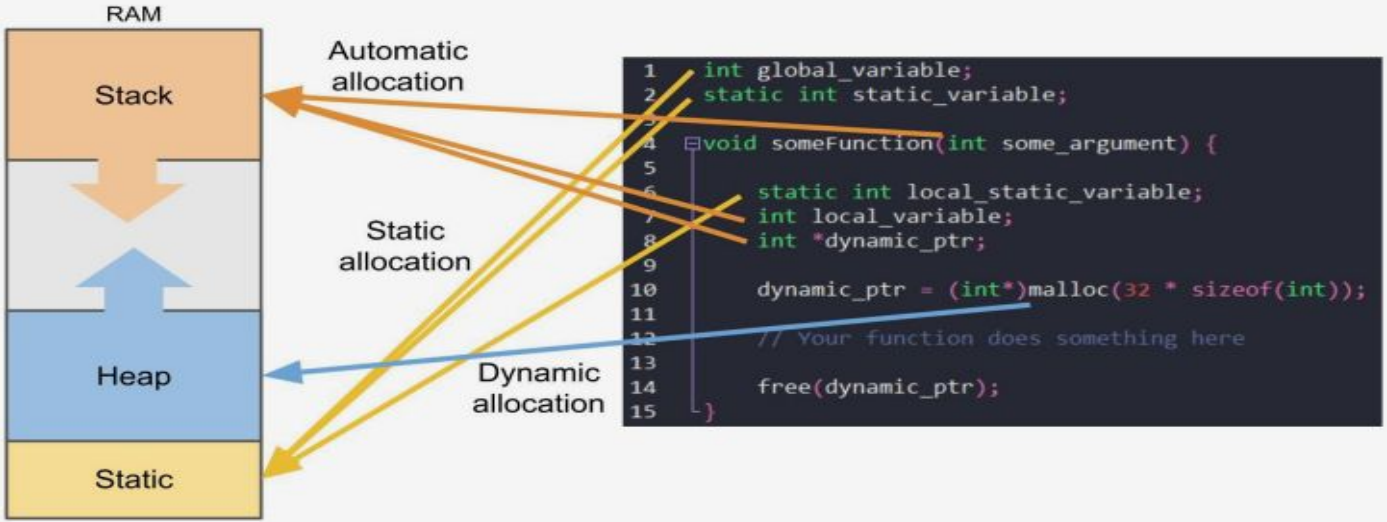


Pointers & Arrays



SSN

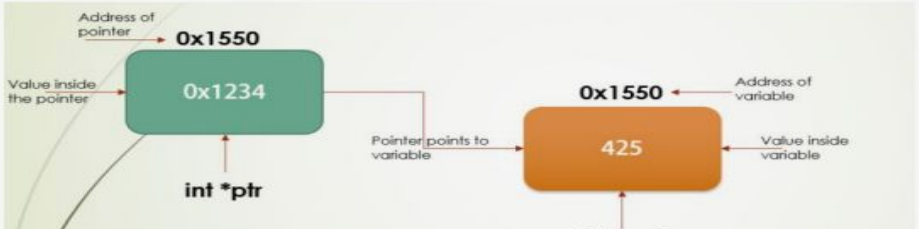
Memory Allocation in C



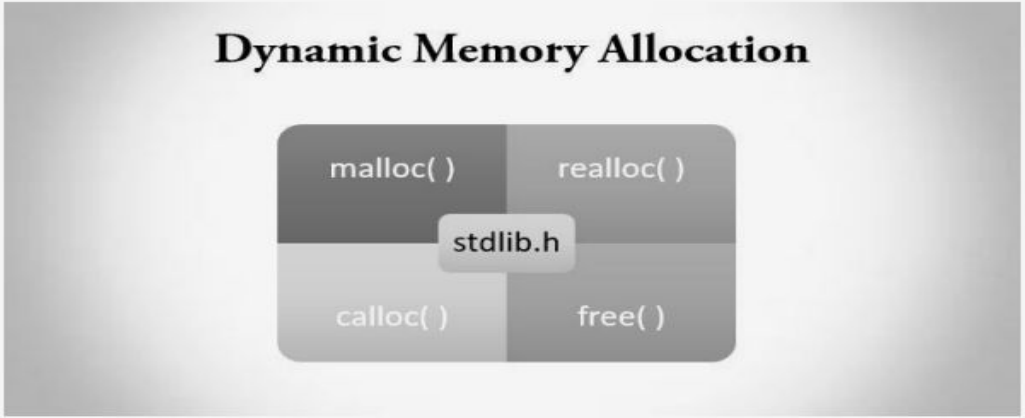
Static vs. Dynamic Memory Allocation

Dynamic Memory	Static Memory
Allocated at run time	Allocated at compile time
Memory can be altered during program execution	Memory cannot be altered during program execution
Example: Linked list	Example: Array

- The **heap** is often called **unnamed variable space**



Dynamic Memory Allocation in C



Syntax:

- void *malloc(size_t size);
- void *calloc(size_t num, size_t size);
- void *realloc(void *ptr, size_t new_size);
- void free(void* ptr);

Any Queries !?

Thank You for attending!

Contact us regarding any questions through email

nandakishor2010608@ssn.edu.in
nitheesh2010343@ssn.edu.in
sanjayrojar2010085@ssn.edu.in