(Lab 8) C Programming - Recursion, Struct, Dynamic Memory Allocation

CS2013 Systems Programming

Department of CSE, IIT Palakkad

IIT Palakkad

Sep 17, 2025

Quiz 7 (15 minutes, **Do not copy the question**)

1. Consider a character array `char array[100]`. To read a string into the variable `array`, you should use

   ```
   scanf("%s",____)
   ```

   The blank is one of `array` or `&array`. Which one ? and why ?

2. `int` is 4 bytes and `double` is 8 bytes in a given machine. Write down which element does `iptr+2` and `dptr+2` point to ?

   ```
   int array[100];          double values[100];
   int* iptr = array;       double* dptr = values;
   ```

3. `"East or west home is the best"`

   Write a C code that stores each word of the sentence above in an array named `proverb`. For example, `proverb[0]` should contain "East", `proverb[1]` should contain "or" and so on ....

Plan (Demo)

- Function declaration and definition
- Scope of variables
- Calling conventions - value and reference
- User-defined types - struct
- Function call / activation stack
- Dynamic memory allocation and structs
- Library usage, CLI arguments: argc, argv

Memory structure during a function call

- Demo using `pythontutor`
- Stack and stack frame
- Every function including the `main()` function has stack
- Space given for every variable defined in a function
- Stack structure and space for variable allocated by code generated by compiler.

Dynamic memory allocation

- Demo using `pythontutor` website
- Dynamic memory allocation used when space can be used up (and freed) based on demand.
- Allocation done via `ptr = malloc()`.
- Allocated memory must be freed after use. Done using `free(ptr)`
- Otherwise, this leads to memory leaks (memory is available but cannot be allocated)

Quick Summary

- Functions and calling mechanism
- Scope of variables
- User defined types
- Dynamic memory allocation
- Library usage

Lab Exercise

Questions (Do the following in your repo)

- Do `$ git switch lab08`. Commit all the changes.
- Do `$ git push -u origin lab08`

Lab Exercise

Questions (Do the following in your repo)

- Do `$ git switch lab08`. Commit all the changes.
- Do `$ git push -u origin lab08`
- Do `$ git fetch && git merge`
- Do `$ git switch lab08` to see the `questions.md` in `lab08` folder.
- **To push changes: do `$ git push -u origin lab08`**

Class repo (for in-class demo)

- Accessible via
  - `git clone git@gitserver:class_repo`
- To see latest changes, cd to the class_repo and do
  - `git fetch && git merge`
  - This does a git pull

Class has ended

- No more pushes to `gitserver`.
- Complete the exercises during off-lab hours.

Humble Request

Please keep the chairs in position before you leave.
(as a token of respect for our CFET staff)