
# TUTORIAL #3: Introduction to C Part II

**Faculty of Engineering and Applied Science**
**Operating Systems SOFE-3950U**

74171 Group 11
Awais Ahmad - 100785259
Mithusan Arulampalm - 100787123
Jawad Kohja - 100786085
Mohammad Mohammadkhani - 100798165

Date: February 8, 2023

**Conceptual Questions**

1. **List each of the modes for the fopen function to the perform the following operations: read, write, read and write, append to a file.**
    - **read**: FILE *fopen("file_name.txt", "r");
    - **write**: FILE *fopen("file_name.txt", "w");
    - **read and write**: FILE *fopen("file_name.txt", "r+");
    - **append**: FILE *fopen("file_name.txt", "a");

2. **Does dynamic memory use the stack or heap? What is the difference between the stack and heap?**
   Dynamic memory uses the heap. A stack is when the memory is loaded at the top and uses a LIFO (last in first out) order. You can push (add) items to the top of the stack and pop (remove) from the top of the stack. A heap uses a tree based structure that satisfies a given condition (max-heap or min heap).

3. **Explain what a pointer is, and provide examples (in C code) of how to change the address that a pointer points to and how to access the data the pointer points to.**
   A pointer is a type of variable that keeps as its value the memory address of another variable.
   **Input**

```c
#include <stdio.h>

int main () {

   int  var = 16;   /* actual variable declaration */
   int  *ip;        /* pointer variable declaration */

   ip = &var;  /* store address of var in pointer variable*/

   printf("Address of var variable: %x\n", &var  );

   /* address stored in pointer variable */
   printf("Address stored in ip variable: %x\n", ip );

   /* access the value using the pointer */
   printf("Value of *ip variable: %d\n", *ip );

   return 0;
}
```

**Output**

```
      |
      |
Address of var variable: 4ea8265c
Address stored in ip variable: 4ea8265c
Value of *ip variable: 16
```

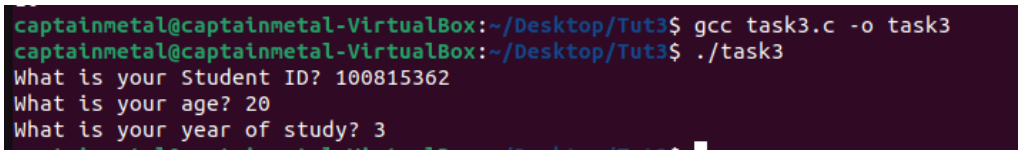
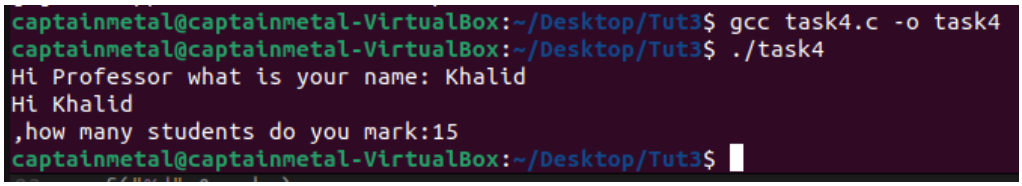
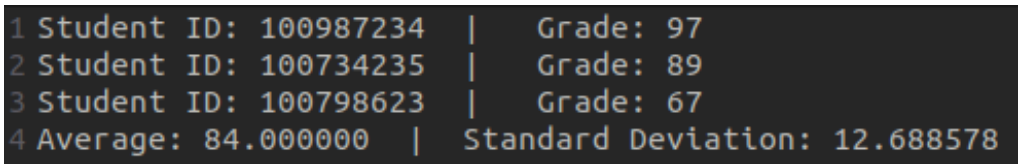
4. **Read the documentation on the malloc and free functions and explain briefly how to use malloc.**
   The malloc function's purpose is to allocate the single large block of memory with the specific size and return a pointer to the memory, but the free() function is the opposite where it deallocates the memory since malloc() function cant deallocate it. This helps free up space in the memory.

5. **What is the difference between malloc and calloc?**
   The Difference between malloc and calloc is that malloc() allocates the memory block of the given **n** size ( in bytes) vs calloc() allocates the memory and also initializes the allocated memory to 0.

---

**Application Questions**

| | |
|---|---|
| 1 | `captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ gcc task1.c -o task1`<br>`captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ ./task1`<br>`What is your First Name? Mohammad`<br>`What is your age? 20`<br>`What is your height? 190`<br><br>`Your first name is Mohammad, your age is 20, and your height is 190cm.` |
| 2 | `captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ gcc Task2.c -o task2`<br>`captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ ./task2`<br>`1`<br>`2`<br>`3`<br>`4`<br>`5`<br>`6`<br>`7`<br>`8`<br>`9`<br>`10` |

| | |
|---|---|
| 3 |  captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ gcc task3.c -o task3<br>captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ ./task3<br>What is your Student ID? 100815362<br>What is your age? 20<br>What is your year of study? 3 |
| 4 |  captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ gcc task4.c -o task4<br>captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ ./task4<br>Hi Professor what is your name: Khalid<br>Hi Khalid<br>,how many students do you mark:15<br>captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ |
| 5 |  captainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$ ./task5<br>Name of Professor:Khalid<br>Number of Students:3<br>Enter the student's Id and grade respectivly: 100987234 97<br>Enter the student's Id and grade respectivly: 100734235 89<br>Enter the student's Id and grade respectivly: 100798623 67<br>Open grades.txt for Resultscaptainmetal@captainmetal-VirtualBox:~/Desktop/Tut3$<br>  1 Student ID: 100987234 | Grade: 97<br>2 Student ID: 100734235 | Grade: 89<br>3 Student ID: 100798623 | Grade: 67<br>4 Average: 84.000000 | Standard Deviation: 12.688578 |