DATA STRUCTURES & ALGORITHMS 1

BATCH – A [WEDNESDAY FEBRUARY 06, 2019: 3:30 PM – 6:30 PM]

LAB ASSIGNMENT – 4 CODE:assign04

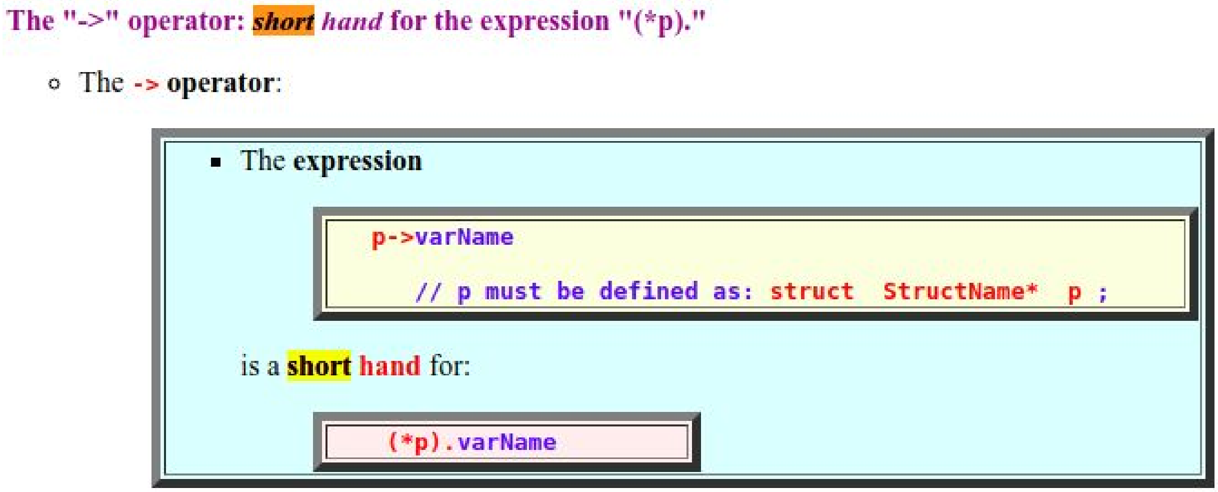
NOTES:

1. Please carefully read all assignments and there is no choice.
2. **Use the template for this assignment**
3. Each problem in this assignment has to be answered in the same c file.
4. Create a .c file following the file name convention:
    a. If your roll number is 'abc' and assignment code is 'assignXX'. Then use the following file name convention as follows: 'abc-assignXX.c'
    b. For example, if the roll number is 92 and assignment code is assign01, then the file name should be 092-assign01.c
    c. Strictly follow the file name convention. When you are ready, submit the solution via google classroom.
5. Follow naming conventions
    a. except for variables in for-loop, none of the other variables should be a single character.
    b. The variable names and function names should indicate what they are storing/computing.

PROBLEM INSTRUCTIONS:

For the following problems write functions which satisfy the following:

1. The functions **should not have a return statement** (hence its return type should be void).
2. **All** the **arguments** to the functions should either be **pointers or void**
3. Do **not use global or static** variables(**except maybe for one student pointer variable 'first-student'**)
4. Please use the '->' shorthand dereference+access operator wherever necessary



The "->" operator: *short hand* for the expression "(*p)."

- The **->** **operator**:
  - The **expression**

```
p->varName

   // p must be defined as: struct  StructName*  p ;
```

is a **short hand** for:

```
(*p).varName
```

Create a Structure **'student'** with the following details

1. Student First name
2. Student Last name
3. Rollno
4. CGPA
5. A pointer to next 'student' (struct) instance (we will use this pointer to point to the next student)

PROBLEMS [Total Marks: 20]:

1. [Marks: 4] Write a function which gets details of a student from the user and creates an instance of the structure in **the Heap**. Its ok to initialize all 'next-student' to NULL.

2. [Marks: 4] Write a function that takes as input two struct instances: student1 and student2. It populates the *next-student of student1 to point to student2 (i.e the address of student2 is stored in *next-student of student1)

3. [Marks: 4] Write a function that takes one struct instance as input and prints the content clearly. It uses the *next-student to fetch and print the next student instance. The function continues until it reaches a 'null' for the next student. You must use **recursion** for implementing this.

<4th problem on next page>

4. [Marks: 8] Use all of the above functions to achieve the following.
    a. Ask user for no-of students: ‘n’
    b. Create ‘n’ student instances, properly chain them using the next-student pointer
    c. The ‘first-student’ pointer should always point to the first student
        i. You might need to modify the function you wrote for problem-1:

```
IF  the first-student is null
        the new instance becomes the first student
ELSE
        The new instance is created
        First student’s next-student now points to new-instance
```

Please store the solutions. Future assignments might ask you to improve upon this solution