CSP 509 - PG SOFTWARE LAB
Lab Test 2 Total Weightage: 12%
Submission Deadline: Oct 29 2018 17:30

General Instructions:

- All specifications must be strictly followed. Failure to do so may lead to substantial loss of points.
- **Upload your submission on Moodle before the deadline. Prefix the filenames of all your code files with your roll number.**
- **If you miss the Moodle deadline, then you would get a 0 for the lab test.**
- **Email submissions will not be accepted.**

Allowed Programming Languages and Library Use:

- Python, Java, C/ C++
- Only primitive data-types such as int, float, char and string (in case of Java and Python).
- Beyond the primitive data-types, you are allowed to have user defined data-types such as array, structures and objects.
- You can also use library calls for File reading.

Question 1 (80 points)

For this question, you are required to build a Binary Search Tree over Employee records. Assume that the Employee records are stored in a text file (input.txt) which contains the records in following format:

<Employee-Num>, <Age>, <Salary> <newline>
…..
…..

All attributes in an employee record are integers.

Following specifications need to be followed:

1. Create a user-defined data-type (e.g., structure or a class) for storing the employee records.
2. Data should be read from a text file (input.txt).
3. Your code should have an **Insert_into_BStree()** function. This function should take one instance (at a time) of the user-defined data type in item 1 and insert them into an already existing (initially NULL) globally maintained Binary Search tree. **Employee-Num** is the ordering field of this Binary Search Tree. **Employee-Num is an integer.**
4. Your code should have an **Print_Inorder()** function. This function should conduct an in-order transversal of the tree and print all the records of tree in that order.
5. Your code should have an **Delete_from_BStree()** function. This function takes a given employee-num and searches of the corresponding record in the Binary Search Tree. If the search is successful, then corresponding node is deleted from tree (and the tree is adjusted accordingly).
6. Appropriate data structures for managing the binary search trees need to be created.

Note: Your code should be generic enough to take any number of records. Grossly inefficient techniques (e.g., assuming a fixed number of records in the input file) would lead to loss of points.

Things to be submitted:

(a) All the code developed.
(b) Put your name and Roll number in all the code files.
(c) Name of the file should have your roll number in it.