



## Lab3 – Binary Trees.

**Important!** Read the document “Information and rules about the labs” . Read through the entire lab specs before starting to write code so that you get an idea of what to do, the extent and can plan your work and time. Read each assignment carefully: it is important that you follow the instructions to get everything done correctly. If something is unclear, contact your lab assistant: Gabriele Capannini ([gabriele.capannini@mdh.se](mailto:gabriele.capannini@mdh.se)) or Abu Naser Masud ([masud.abunaser@mdh.se](mailto:masud.abunaser@mdh.se)). You need to complete all the lab assignments to be eligible to attend the final lab examination.

---

### 1. Binary Trees

#### Task 1.1 Download the skeleton

Download and extract the file `Lab3.zip` . Create a new project with a proper name and add all the three files to your project. Make sure you can compile the project without any error messages (there may be warnings: just ignore them at the moment).

#### Task 1.2 Binary Search Tree & Menu Program

You will now implement all functions that are declared in `BSTree.h`. There are also some static empty features in `BSTree.c`. You can implement these in the order you prefer. Also, note that many of the functions will use each other. One suggestion, however, is that you start with the static function `createNode()` in `BSTree.c`. This feature creates and assigns tree nodes and it is needed to insert tree nodes. In `test_BSTree.c` there is a menu function you can use, it is ok if you want to write your own menu function. Because there are the many features to be implemented: type one function at a time and check (by means of the menu) if it works properly before you pass to the next. The functions `writeSortedToArray()` and `buildTreeSortedFromArray()` are used by the balancing function and need only be used if you choose to implement balancing as suggested in the `balanceTree` comments.

#### Task 1.3 Test & Debug

Once all functions are implemented, test the tree. Run the test function in `test_BSTree.c`, if an assertion is raised, you need to find the error in your implementation (the comments in the function should help). If everything works properly, you are ready to report. Note that the traversal functions (Preorder, Inorder and Mail Order) must be tested manually!