The Tasks

- In this assignment, the task is to implement a one-variable polynomial class derived from `BST<int,double>`, where `BST` stands for Binary Search Tree.
- Actually, you should keep the `BST` class template (and its node class template) unchanged from what we have covered in the class, and put any polynomial operations in the derived classes.
- Each node of the tree represents a term of the polynomial. The `key` field corresponds to the exponent and the `element` field corresponds to the coefficient.
- Let us name the polynomial class `PolyBST`.

The Tasks

- You need to implement the following operations for your polynomial class:
  - Operator `+` for adding two polynomials.
  - Operator `–` for subtracting one polynomial from another.
  - Operator `*` for multiplying two polynomials.
  - Operator `*` for multiplying a polynomial and a double scalar.
  - Operator `<<` (to an `ostream`) for printing out the polynomial.
  - Function `setTerm(exponent, coef)` to set a term of the polynomial. It can generate a new term, update the coefficient of an existing term, or delete a term (if input `coef` is zero).

The Tasks

- One more operator required: Operator **=** for assigning one polynomial to another.
- Provide suitable constructor (creating an empty polynomial) and destructor. You can also just use the constructor and destructor of the base class, but you still have to implement them.
- Notes about text output:
  - Format example: `3*x^3 – 2*x^2 + 5`
  - When the polynomial is empty, print `0`.
  - Always print the terms in the order of large to small exponents.
  - You will need to use some type of tree traversal for this operation. Think about how you want to implement it.

The Guidelines

- Allowed environments: VS2012/2013/2015, Dev-C++. Indicate your environment at the beginning of your code.
- You need to write your own `main` function to test your permutation generation function. You do not need to include this `main` function in your submission. The instructor will provide a test `main` function for you.
- No usage of STL class templates allowed.
- Include documentation; this will be part of your grade.
- Demo: Only a randomly selected subset of students; will be announced separately after the due date.

The Guidelines

- Submission:
  - Use E3 only.
  - Submit all your code in a single header file (`.h`). Name it `P4_xxxxxxx.h`, where `xxxxxxx` is your ID. **Do not** submit your `main` function or any file that is not your code (such as the *.sln file). No compressed file (*.zip, *.rar, etc.). **Only the header file!!!**
  - Due date: **12/18/2015**. There's a grace period of 4 days with 10% deduction per day. (The deduction kicks in only when you have accumulated more than three days of delay during the semester.)