

# Computer Programming Assignment 5

Due date: 2017-12-15 23:59:00

## Checkpoints

1. You should do the assignment in your own. You are not allowed to share code with others and/or copy code from other resources. If you are caught, as in the syllabus, you will get a failing grade.
2. Grading will be done in Linux environment using java 1.8 and GNU g++ 7.2.0 with -std=c++11.
  - a. If you program under clang, please write comment on top of your program that your program wrote under clang environment.
  - b. There will be no guarantee that your program will run if it is wrote under Visual Studio
3. Program failed to compile/run will result 0.
4. Do not infinite loop your program to repeat unless you are told so.
5. Do not change input/output format unless you are told so.
6. Do not color console.
7. Write your name and student number at top of program as a comment

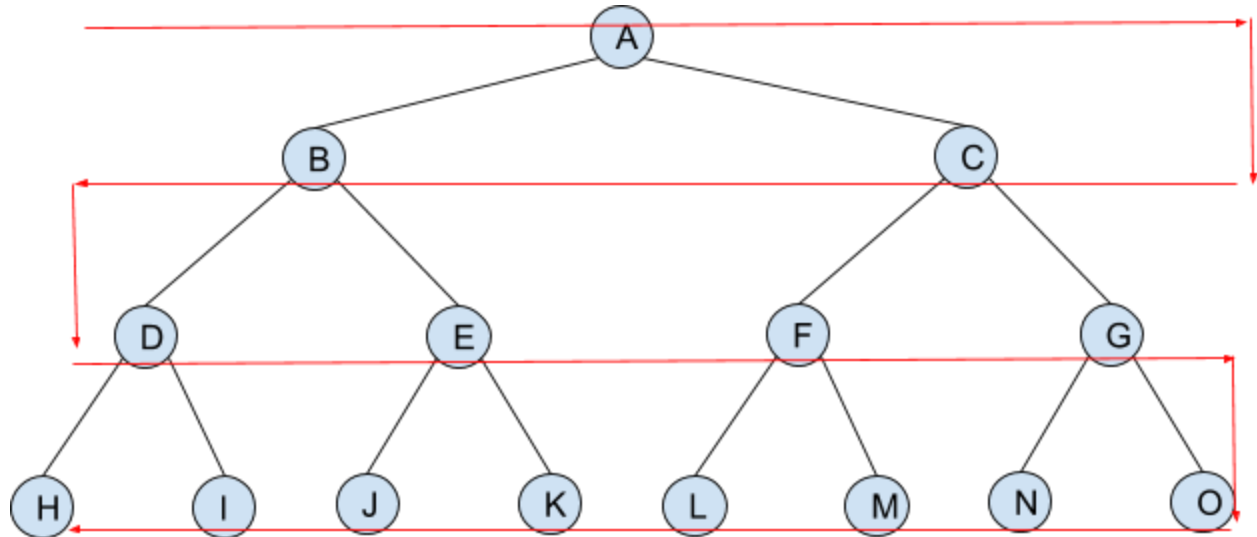
## Submission

1. Submit your assignment on eTL.
2. Zip your file (or tar) as '<Student ID>-assign5.zip'
  - a. Do not create subfolder
  - b. Do not include package
3. No late submission is allowed

## Tree

- Write a class Assignment5\_1.java
- Constructor takes two string preorder and inorder
  - Given two string build a tree
- Write a String report()
  - Returns the label of nodes that travels levelorder from left to right, right to left, left to right, and so on.
- Do not modify Node5\_1 get()
  - Root will be accessed to check if tree is constructed correctly.
- Do not import any classes.
- You may write your own classes to help Assignment5\_1.java, but you may not import any classes on those files as well.
- Do not modify Node5\_1.java

ex.) "ABDHIEJKCFLMGNO" "HDIBJEKALFMCNGO"



String report() returns "ACBDEFGONMLKJIH"

## Operations between Scalar and Vector(C++)

Your program should provide “+”, “-”, “\*”, “/” operators for Scalar and Vector objects.

The order of computation matters here, and you should print error message and exit program when an operator cannot be applied to.

Error message format : “[operator] cannot be applied to [Entity1] and [Entity2]”.

e.g.) “+ cannot be applied to (1, 2) and (2, 3, 4)

- Scalar : a numeric value ( [double] )

- Vector : Math entity that represents direction and length in N-dimensional space. It consists of a sequence of N Scalar elements.

Each entity's print format follows form of the below.

Scalar : [Number output from cout << number << endl]

Vector: [(num1,\_num2,\_ ... ,\_numN)]

e.g.) Vector vec1 = {1,2,3,4,55} => (1, 2, 3, 4, 55) //Spaces should be precisely matched

**Definition of each operation is listed below.**

(Scalar = s, Vector = v[n] = (v\_1, v\_2, v\_3, ... , v\_n) )

### [Scalar and Scalar]

1. Scalar [+ , - , \* , / ] Scalar = just regular mathematical result of numbers.

When division by zero happens, you should print error message as mentioned before.(Error message format)

### [Vector and Vector]

2. Vector(vec1) [+] Vector(vec2) = ( vec1\_1 [+] vec2\_1, vec1\_2 [+] vec2\_2 ... )

Size of both vectors must be the same.

e.g.) (1, 2,3) + (4,5,5) = (5,6,8)

3. Vector(vec1) [ - ] Vector(vec2) = (vec1\_1 [-] vec2\_1, vec1\_2 [-] vec2\_2 ... )

Size of both vectors must be the same.

4. Vector(vec1) [ \* ] Vector(vec2) = (v\_1, v\_2, ... v\_n) \* (x\_1, x\_2, ... , x\_n) = 
$$\sum_{i=1}^n v_i * x_i$$

Dot product of two vectors.

5. `Vector(vec1) [ / ] Vector(vec2) : Not defined.`

**[Scalar and Vector] (result type = Vector)**

6. `Scalar [+, *] Vector = Vector [ + , *] Scalar = (s [+, *] v_1, s [+, *] v_2 , ...)`

7. `Scalar [-, /] Vector = not defined, Vector [-, /] Scalar = (v_1 [-, /] s, v_2 [-, /] s, ...)`

Scalar should be located after Vector for `[-, /]` operator.

8. `Vector [ / ] Vector = not defined.`

MyVector.h MyVector.cpp files will be provided to you.

You can make more cpp files to support MyVector.h and MyVector.cpp.

Write more codes into these files to calculate results of the entities.

Format of constructors will be in the cpp files. For grading, I will run testing functions where each type of objects are generated by constructors. Utilize operator overloading so that  $v3 = v1 + v2$  becomes addition of v1 and v2.

Please refer to those files for more information.

[Input Range]

Scalar(double) :  $[-1e-15 \sim 1e15]$

Vector Size :  $[1 \sim 4096]$

## Submission

Make a zip file having "MyVector.h", "MyVector.cpp" and supporting .cpp, .h files into "Assign5\_2.zip".