

# Lab 1: Object Oriented Programming

## 1 Introduction

Our urban forest is important for many reasons; trees provide shade, store carbon and prevent runoff, and their presence is known to improve our collective mental health. In this lab and several others we will be looking at information about trees that has been gathered by our city's policy makers and forestry researchers. As a first step, we will be representing trees as software Objects in Java and associating them with attributes and methods. Along the way, we will be overriding some methods from the Object class and exploring the Comparable interface.

We've provided some tests to help guide your software implementation, but remember that we will be testing your submissions on other tests as well. So, make sure to test comprehensively.

To begin, log into MarkUs and click on the assignment called "Lab 1". Pull the starter files so that you can edit them on your local machine. You should find a folder within your Lab01 folder and five java files therein: *Main.java*, *WhiteAsh.java*, *WhiteAshTest.java*, *Forest.java* and *ForestTest.java*.

## 2 Programming Task

The code for this assignment relates to White Ash Trees, which can be found around Mississauga and, among other things, are used to make baseball bats. Ash trees are carbon rich and the amount of carbon in each tree can be estimated based on its diameter using allometric equations. Allometric equations provide biomass estimates for specific species of tree from tree measurements such as diameter at breast height (DBH). The percentage of carbon in a given tree can then be calculated based on the tree's biomass.

Your first task is to implement a WhiteAsh class to represent the White Ash species of tree. You will include the following methods in the WhiteAsh class:

- The method *grow*. This will increment the diameter of the tree based on time that has elapsed (in years). The growth rate for an Ash tree has been provided as a default attribute, and is 0.3 cm in diameter per year.
- The method *carbonContent*. This will calculate the carbon content in the wood of any given White Ash tree. The formula to calculate this is based on an allometric equation; the parameters for this equation have been provided in your starter code.

- The method *fellTree*. A tree that is felled (or chopped down) will have its 'diameter' attribute set to zero, and its 'living' attribute set to false.
- The method *equals*. This will compare two trees. If the two trees are both living and have the same diameter, you can consider them to be 'equal'.
- The method *hashCode*. Two equal trees should have the same hashCode!! Unequal trees should not. Create a hash function that guarantees 'equal' trees will land in the same bucket when hashed.
- The method *compareTo*. This method is derived from the Comparable interface and implemented by the WhiteAsh class. Consider one tree to be "greater" than another if it is alive and the other is dead. If both are alive or dead, call the one with the larger diameter "greater".

Once you've completed the tasks above, implement the following methods in the class Forest:

- The method *currCarbonContent*. This will return the sum of all the carbon content in all the wood of all trees in a given forest of White Ash trees.
- *harvest(WhiteAsh target)*. This will 'fell' every tree in the forest with a diameter greater than or equal to a target input tree (i.e. all trees that are "greater" than whatever target is supplied).

### 3 Testing your Classes

A small number of test cases have been provided to help you test your implementations. Make sure you pass these tests! And, as ever, make sure you write your own tests, too.

### 4 What to Submit

1. WhiteAsh.java
2. Forest.java

HAVE FUN AND GOOD LUCK!