**Question 2:**

There is a compiler error since the Coord class does not have a != operator function defined. It does not know on what bases to make comparisons between two Coord objects. When the insert function is called, it calls the doInsertOrUpdate function, which calls the find() function. The find function checks to see if p->m\_key != key. In the case of Map<int, double>, C++ already has != operators defined for int and doubles. However, it does not know on what bases to compare two Coord objects, since p->m\_key and key are two coord objects in this map. Therefore, the compiler generates a compiler error.

**Question 3b**

If we didn’t have the path variable, then we wouldn’t be able to keep track of our updated string since the function is a void function and does not return anything. Therefore, it would not be able to keep track of all the classes that come before it.

**Question 4a**

**The time complexity is O(N^3)**. Ignoring each assignment operation since it only has a time complexity of O(1) (lower order), the first for loop runs N times. The nested for Loop also runs a worst case of N times and the inner most for Loop also runs a worst case of N times. Therefore, the worst case scenario of total number of computations is N\*N\*N which is N^3. So the time complexity is O(N^3).

**Question 4b**

**The time complexity is O(N^3)**. Ignoring each assignment operation since it only has a time complexity of O(1) (lower order), the outer loop runs a worst scenario of N times. The first nested loop is variable and dependent on the variable of the outer loop. However, since the outer loop may run a worst case scenario of N times, and since i < N, j < i , j will also run N times. This makes the total number of computations N\*N. The inner most loop runs a worst case scenario of N times. Therefore, the total number of computations is N\*N\*N, leading to a time complexity of O(N^3).

**Question 5**

**The time complexity is O(N^2)**. Map res(\*bigger) runs N computations since \*bigger has N elements and it is being copy constructed. The for loop runs N times, leading to N computations. The res.insert function within the for loop calls the doInsertOrUpdate function, which then calls the find function. In a worst case scenario, the find function will traverse through N nodes. This leads to an overall time complexity of O(N^2 + N). Since we only consider the most significant term and drop the coefficient, the overall time complexity is O(N^2).