Max Marchuk

CS202

Karla Fant

Assignment 4 Java Debugging

For this program I used an IDE for coding in Java. The IDE I used was eclipse. The cool thing about eclipse is that is basically compiles while you are typing and tells you about any errors while you are typing. This is great for small applications like this assignment, but I can see how it would be bad if it was a huge program that could take a really long time to compile. This was really useful however, in this program. The only time I really had to debug was when I was having scope errors and my root pointer wasn’t being modified. This was due to the fact that you can’t modify things that are passed through functions. Whenever I modified the root pointer in a function, it would change that root object and it would display fine, until I returned from/reached the end of that function. This was hard to work around because I was used to C++’s way of passing by reference and being to directly modify something anywhere regardless of where it was in the program. My work-around for this was to simply have the insert function traverse to where it needed to with the data being inserted, and to eventually find that location, and return the node at where the data needs to be inserted. This way, I was modifying root in the BST class, so that it would stay the same in the scope of that BST object until that BST object was destroyed in the main class’s main function. Debugging this little part of the program took me a really long time, because I was so used to using pass by reference in my previous programs. I can definitely see the benefit of this thing in Java, but it really does make things more difficult for this particular program and the way I began to code it. The benefit of this method of modifying objects is that, in a way, it makes the program more object oriented. Things can only be modified in the scope of themselves, and the way to get new data is to retrieve it from the functions of other objects. I don’t really see any benefit to that in most applications that I could think of, but I’m sure it has its positive and negative sides. In conclusion, debugging in Java is great because the compiler works while you type, and you catch your errors before you continue to make things worse for yourself in the long run.