**Algorithms assignment 1**

For parenthesis {} validation I have made use of Stack.

Made use of String Tokenizer for this purpose. Since the test files given had the “ {” format I made use of this as the delimiter to split the file into tokens.

Whenever a “{“ is encountered it is pushed into the Stack and whenever the token is “}”

Whenever the Stack is not empty or there was nothing to pop but pop was encountered then the validate class will be false which means the class is not valid.

Whenever the Stack is empty meaning push and pops are equal in no then the validate class will be true and this mean the class is Valid.

While doing this the tokens are also checked for /\* and \*/. If these tokens are encountered, then the tokens in between them are skipped since it is multi line comment. So if {} are encountered between the multiline comment it will not affect the overall code validity.

For finding the keywords I have made an Array which will have all the keywords of Java stored in it. The String Tokenizer will tokenize the inputfile and each keyword is matched with the token. If they match for the first time it is placed in the Hashmap using put(key,value=1). If it is found more then once then put(key,value++) the value will be incremented.

For the identifiers I found all the words before “=” and also all the words after keywords and similarly used hashmaps to save them. I have not found all the identifiers by this but quite a lot of them have been picked up rightly. I have made use of Regex to find the identifiers starting with letters(uppercase/lowercase) or $.