Abstract—Programming contests such as International Olympiad for Informatics (IOI) and International Collegiate Programming Contest (ICPC) are very effective to encourage the young and bright programmers. Generally those programming contests require for contestants to a few(3-9) tasks related on algorithmic problems within a limited time3 to 5 hours in a day;) In this paper we are concerned to characterize the structural features of codes in terms of the correctness of task solved and the structural code complexity depending on the task difficulties. For this work, we have collected a set of 2,400 programming codes submitted to KOI (Korea Olympiad for Informatics) in year 2011, 2012. Also we collected collegiate source codes which are submitted at the preliminary contest session for ICPC 2009, 2011 and 2012 for East-Asia regional contest. We try to reveal the relationship among the task difficulties, personal coding styles and school grade(elementary, middle and high school) with data mining tool WEKA. We generally believe the software complexity of good codes is relatively lower than that of erroneous codes. In contest codes, we take a set of right answer codes and wrong answer codes to use the learning data. Main concern of this paper is to show the correlation or causality between the correctness of codes and the software complexity. We know there are so many different software complexity measure such as cyclomatic measure. In this paper we propose a more simple and straightforward complexity measure based on the block tree structure. Experiment showed that the proposed software metric shows somewhat more meaningful dependence in difficult tasks than in easier and trivial tasks. And we found that a set of outlier codes in terms of code complexity are almost wrong answer code. In ICPC codes, we discovered that the good collegiate programmers(who get the higher scores) try to keep their code more compact in a lower structural complexity. But these are not confirmed in KOI(elementary, middle and high school) codes, which confirms the coding style gets determined in a more concrete discipline on the higher education system. We have used WEKA to predict the code correctness using only the the proposed code complexity, which are described quantitatively in detail in the final experiment Section.

I. MOTIVATION

The study on quality evaluation for program using software metric is old concern of software engineering. Typical measured value of metric are coupling, cohesion, length of code, McCabe Cyclomatic Complexity, Halstead software science metric, etc[1], [2]. Apart from these static measured value, there are another attempt to find new metric with graph theory[3], [4], [5]. These measured value are utilized as a means to forecasting the location[6], [7] of program bugs. Moreover it can be use to limit the complexity of source code for homework in school. It can induce the student to write the source code concisely. Also, as the one way of maintenance[8], [9], the metrics can be used to classify the project into modules for making easy to understand the composition of project[10].

In this paper, we analyze source codes obtained from programming contests and finds out two issues using data mining technique. One is analysis the correlation of the structural and lexical features. The other is to find a correlation between scores and features. Ultimate goals is analysing correlation between static feature and correctness(in this case, score) of program and, by extension, forecasting the correctness of program(score) from static structure if possible.

We use the data that is source codes submitted by student who take part in the KOI. The contest problems are classified into elementary, middle, high school. Each grade consist of 100 student who passed the first regional preliminary stage of this contest. About four various level of difficulty questions are given, examination hour is limited to 5 hours. During this limited time, student must submit their source code to solve the question. This source code is evaluated based upon 20 blind test data approximately. If the source code makes correct result for each case of the data, the submitter score on the question. Lastly, the sum of the score of each question is total score of each student. In case of hard problem, the level of difficulty is up to International programming contest. So, only 2-3% students can get the perfect score. In more difficult case, there is no one get the perfect score. On the other hand, in case of easy problem, the tendency of questions is too easy, so the 80% of students can get the perfect score. Figure 1 is difficulty of problem. Elementary school hard problem is up to middle school easy problem. Middle school hard problem up to high school easy problem. The question deal with computational problem solving method such as algorithms, data structures and so on. Since each question has CPU process time limit, if the source code doesn’t make the result in process time, then that source code is regarded as wrong, even though it makes right answer. So student’s efficient programming capability is directly connected to the score.

II. APPROACH TO MEASURE SOFTWARE QUALITY

Software metrics obtained from static analysis are usually used to evaluate a source code quality. Traditionally there are many kinds of static software metrics. Such as code coverage, cohesion, coupling, function point, number of classes and interfaces etc. Recently, people presents new metrics to evaluate the source code quality. Such as exploring the influence of identifier names on code quality[11]. In the case of analysis of