



Programming Grader

Shubham Swami¹; Dhwani Shah²; Sofia Francis³

¹Computer Department & Mumbai University, India

²Computer Department & Mumbai University, India

³Computer Department & Mumbai University, India

¹swamishubham19997@gmail.com; ²dhwani806@gmail.com; ³francissofia2@gmail.com

Abstract— The ability to program is one of the core tools used by computer scientists, and programming competencies a recommended requirement for ABET accreditation. In our experience, students learn programming skills best by writing many programs, ranging from simple to complex. Overworked teachers can be dismayed by the prospect of grading still more programs per student as well as teaching introductory classes with large enrolments. Even the difference in the ability to check programs between the teachers is also prone to unintentional prejudice. And the most severe problem is perception of prejudice in the students whose programs are evaluated. The automatic grading approach address the above problems and offers substantial advantages and opportunities, but also some challenges. It eliminates the variability of the solution provided by different teachers. Even it explores certain factors to grade the program such as number of test cases passed number of attempts taken for successful compilation, etc. We present Programming Grader, a web-based automatic program grader that provide the robust and a reliable result to the mentioned variability and load of checking problem continuing from long time.

Keywords— Programme Grader, Web-based, Programming Skills, Test cases, C, C++ Languages

I. INTRODUCTION

From its dawn, computer technologies have seen tremendous development and growth, and in the past few recent years it is been developing even more vigorously due to the escalation in the innovative minds and new demands of the growing world. It's like now has become a major integral part of everyone's life in some or the other way and also will be playing an important role even in the future. Computers are helping in every sector whether it may be industries, services, innovation, defence, research or even medical they are been present everywhere helping the humans making their survivor easy. Basically machines have revolutionized human relationship with each other whether it may be through various communication system like different online chatting applications, mobiles phones or even the entertainment facilities like YouTube, Live TV, and even lot. So, it's like computers are ubiquitous, in this present world.

Computers being ubiquitous, everyone spends some specific duration with them in their daily routines or even might as the part of their job. Even the grading of such courses also become an important part for the motivation and make understanding better of the students.

II. LITERATURE REVIEW

Now a day's in the era of machines where computers are playing major role in every sector. Its significance increases when we talk about the role of computers in the field of education where we are developing and nurturing the future minds who will lead the next generation ahead. So computer programming is major part of this developing system where we need to make understand the new minds how to develop a logic for a given problem and provide a solution to tackle it.

Emily Stankov, Mile Jovanov, Bojan Kostadinov and Ana Madevska Bogdanova[1] together has provided the guidelines using data mining technique to evaluate the source code of the students by similarity detection as it is not a fully automated technique so the level of strictness is less and it works on modern system known as 'MENDO'. The main loop in a system it just review the test case in sense of number of points where our system will be fully automatic and checking the test case even with their syntax checking and appropriately give the feedback to the students and faculties both.

Derek S.Morris[2] contributed the design of the grader which is based on software testing. The program being tested are Java Classes, i.e., Class Under Test (CUT). The design here have three major components: - A Test Driver –that is used to provide sequence of inputs to the CUT, and evaluator of the CUT's outputs and a framework that protects the design of the grader. Here reflection class allows the student program's method to be individually found and executed. Java Inheritance allows the invalid or methods with errors to be overthrown with the better known method and regular expressions check the desirable and undesirable coding pattern of student's program. The major drawback of this technique is that it allows the evaluation process to continue in the presence of serious flaws

Milena Vujosevicic-Janicic, Mladen Nikolic, Dusan Tomic, Viktor Kuncak[3] has proposed the system for automatically evaluating the programs which is beneficial for both teachers and students. This evaluation is helpful for teachers in grading assignment and it provides immediate feedback for students in turn which is very important in process of studying specially in computer Science. The main framework of this system is based on merging the information from the three evaluation methods: Software Verification used for automated bug finding, Control flow graph (CFG) for similarity measurement, Automated Testing. But still this system some counterpoints as it plans to make an integrated web-based system along with compiling, automated testing, profiling and detection of plagiarism of student's programs. It intends to improve feedback to students by indicating the missing parts of the codes compare to the teachers solution.

Abhishek Varat, Mayur Vetel, Pooja Bawadkar, Shubham Shinde, Varsha Naik[4] proposed a system that make use of Interactive technology in Learning for the students. They have designed a system that provides an interface to teacher through which they can create new programming assignments and assign it to a particular class. It provides a detailed report of class for each programming assignment. This project has two working ends: - the client end and the cloud end for verification of assignments. At the client end teacher can create the programming assignments and can store them on the database and students can also send their programs to the compiler at the cloud end. Database and the compiler are the components of cloud end where the compiler creates error log and sends them to the students at the client end as well as evaluates the assignment and stores the result. This project doesn't offer Human interaction. If user submits malicious code then it is possible that all the information saved on the server can be lost. Even there are many security attacks in this system such as eavesdropping, data modification, and denial of service which can put the system at risk.

III. EXISTING SYSTEM

The existing system that provides an interface to teacher through which they can create new programming assignments and assign it to a particular class. It provides a detailed report of class for each programming assignment. This system doesn't offer Human interaction. If user submits malicious code then it is possible that all the information saved on the server can be lost. Even there are many security attacks in this system such as eavesdropping, data modification, and denial of service which can put the system at risk.

IV. PROPOSED SYSTEM

The Proposed system concentrates on grading the students programming assignments and grades their assignment on their source code, which in short also help the assistant staff to check out the assignment with pen and looking around individual student's assignment and rectifying their logic and giving grads to them. The proposed system aims to check the student's assignment by checking the source code submitted by them on the site and allotting them the grades according to their code. This system works like the teaching staff first has to enter the assignment for the students like according to which they have to first give the description of the problem, then the staff has to give the input test cases along with the output test cases to match for the code submitted by the students.

I. Grading Model

Our project presents the grading model developed in this study, the core of our work. We explain the theoretical idea in an incremental fashion to help users understand with the development derivation of the grading model of our programming grader. For grading programming assignments of various programming forces universities deploy teaching assistants. These assistants follow a set of grading policy. This grading policy is quite objectively defined at least for courses like introductory programming.

Using the fact that programming assignments have objective grading policies, we attempt to capture these policies and develop a grading model that performs similar task as of teaching assistant. The grading policy depends on the following factors:

- The grades are directly proportional to fraction of test cases passed
- The grades are inversely proportional to the time taken to solve a problem

II. Working Of System

The grading system consists of the following components which play the major role of the system are:

- Admin,
- Staff
- Student (User).

The Admin is the first major component of the system which registers the new student users on their request into the system, without which the student cannot access the system. The Staff is the second most important person of the system which login into the system with their own Login Id and password. Then the next step is that the staff has to upload the new Assignment into the system with its Title and Description along with the input test cases and the output test cases which are used to verify the code submitted by the student.

The main User of the system for which the system is completely made is the Student which using its own Account details given to him by the Admin, and then Login into its account. On Login the student can check for the New or previously uploaded Assignment which is needed to solve as per the requirement.

The student then write the source code for the given problem in any one of the two language as C or either C++ as per the option of the two compilers given to user, then finally the user has to submit the code and the result will be shown to them, if there is any mistake in the code or it does not satisfy any test case it will state to the user after submitting, the line or the part in which error is there and finally if the code is submitted successfully then it will show result as successful and will show the output to the user. And the significant part the grades allotted to each student will be displayed to the staff which then they can show to the students and help them to improve their programming skills

The staff will first login into the system using its own Login ID and password. Staff will upload the new assignment with test cases for the students along with the starting and ending dates and time. Starting and ending dates defines the period of time for which the assignment will be available for the students. Staff assigns the results and grades to the students.

V. SYSTEM ARCHITECTURE

In this proposed system we have main Grading system. Grading system is the main processor on which the whole system works. The student get themselves registered first with the help of the Admin. Then the method of checking the code is done by the evaluation process of the system with the help of the test cases provided to the system. The result then generated is stored as the program files of each student in the main system code files, and the grades are assigned to every student on the basis of their code and test cases submitted by the staff.

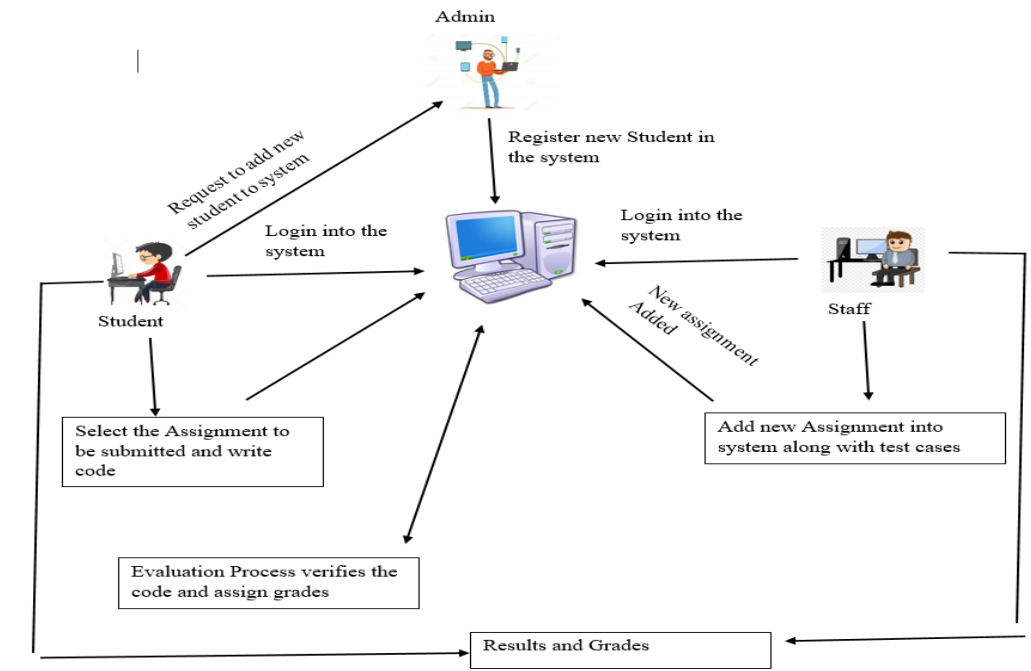


Fig. 1 System Architecture

VI. CONCLUSION

The aim of our project was to come up with the grading model for programming courses and develop a tool implementing the grading model. We have presented the model which evaluates individual student's source code and will generate the grades as scores for the students. This will significantly reduce the workload of instructors by reducing the time taken to manually grade the assignments. Thus, time can be utilised for analysing and identifying strengths or weaknesses of the students. After performing various experiments and judging our programming grader on factors that are more related to the purpose of grading it would be safe to say that models performance is comparable to that of the teaching assistants.

ACKNOWLEDGEMENT

Owing deeply to the supreme, we extend our sincere thanks to God almighty that made all things possible. We are grateful to Dr.Gyanappa Walikar, Head of the Department of Computer Engineering, for his support and encouragement. We also extend our sincere thanks and our sense of gratitude to our guide Ms Prajкта J Akre with her timely motivation and help make the work possible. Finally we thank our parents for their constant support and blessings.

REFERENCES

- [1] Emil Stankov, Mile Jovanov, Bojan Kostadinov and Ana Madevska Bogdanova. *A New Model for Collaborative Learning of Programming Using Source Code Similarity Detection*. 2015 IEEE Global Engineering Education Conference (EDUCON).
- [2] Derek S. Morris'. *Automatically Grading Java Programming Assignment via Reflection, Inheritance and Regular Expressions*. 32nd* ASEEnEEE Frontiers in Education Conference. November 6 - 9.2002. Boston. MA.
- [3] Milena Vujošević'-Janic'ic' ^{a,†}, Mladen Nikolic' ^a, Dušan Tošic' ^a, Viktor Kuncak. *Software verification and graph similarity for automated evaluation of students' assignments*. *Information and Software Technology* 55 (2013) 1004–101.
- [4] Abhishek Varat, Mayur Vetel, Pooja Bawadkar, Shubham Shinde, Varsha Naik. *Online Interactive Educational System for submission and Evaluation of programming Assignment*. 2017 International Conference on Intelligent Computing and Control (I2C2'17).

- [5] Shashank Srikant and Varun Aggarwal. *Automatic Grading of computer programs: - A Machine Learning Approach*. 2013 12th International Conference On Machine Learning And Applications.
- [6] Jong-Yih Kuo and Louisa Chu. *Intelligent code analyzer for online course management System. Proceedings of the 2005 Third ACIS Int'l Conference on Software Engineering Research, Management and Applications (SERA '05)*.
- [7] Susilo Veri Yulianto and Inggriani Liem. *Automatic grader for programming assignment using source code analyzer*. 978-1-4799-7996-7/14/\$31.00 ©2014 IEEE