Tools for programming in MOOCs: Assess Student’s Knowledge
Under the supervision of Prof. Deepak B. Phatak

Submitted by:
Veerta Singh
M.Tech (I.T.)
(2013-2015)
Table of Contents

1. Introduction.......................................................................................................................3
2. Literature Survey ...........................................................................................................6
3. Overall description........................................................................................................17
4. Analysis Model..............................................................................................................19
5. Work done ....................................................................................................................21
6. Conclusion ....................................................................................................................21
7. Implementation and Future Work ...............................................................................21

1.1 Purpose .......................................................................................................................3
1.2 Scope ..........................................................................................................................5
1.3 Definitions, acronyms, and abbreviations ..................................................................6

2.1 References ..................................................................................................................15

3.1 Product Perspective ....................................................................................................17
3.2 Product Functions ........................................................................................................18

4.1 Use Case Diagram .....................................................................................................19
4.2 Flow Chart ..................................................................................................................20
1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Purpose

The purpose of this document is to present a detailed description of the requirements for the “Tools for programming in MOOCs: Assess Student’s Knowledge” project. The purpose and complete declaration for the development of system are illustrated. This document is primarily intended to a Student for assessing their programming knowledge in MOOCs when they start learning basic programming language C/C++ as a novice programmer. The Software Requirements Specification (SRS) captures the complete software requirements for the system or a portion of the system.

**edX:** An organization established by MIT and Harvard University in May 2012 that develop an open-source technology platform to deliver online courses. edX is a non-profit MOOC, available to the open source community as different modules such as LMS(Learning Management System), CMS(Content Management System), edx-ora (Open Response Assessment) etc. It hosts online university-level courses in a wide range of disciplines to a worldwide audience at no charge and to conduct research into learning. Aim is to provide better education to all over the world through online learning platform, from the combination of best faculty and reputed universities.

Today there are three MOOCs platform providers available: Coursera, Udacity and edX.

**Coursera:** Founded a year ago, by Stanford Computer Science professors Andrew Ng and Daphne Koller, Coursera is the most popular MOOC platform today. More than 3 million students take classes on diverse topics offered by 62 universities from various countries. What makes Coursera unique is the wide range of subjects they cover. You can take courses in Arts, Economics, Life Sciences, Law, Computer Science, Chemistry and many more, all taught by leading educators throughout the world. This great variety gives you incredible flexibility to explore as many topics as interest you, no matter how diverse your interests are. Each course runs on a fixed schedule so you’ll have to ensure that you have enough free time to study the courses that you want, but if you miss a particular course or there are too many that clash you can simply study them the next time the re-open. Some courses also offer archives and self-paced schedules

Coursera does not produce the courses on their platform themselves, but rather allows universities to create them under a set of guidelines and a common structure, generally
consisting of weekly short video lectures, content quizzes and assignments and also including a midterm and final exam but this varies between courses. Some have only quizzes and videos whilst others have only a final exam. Also, computing courses generally have automated coding assignments while humanities courses usually use peer-reviewed essay assignments. The final decisions on teaching style, course content and grading rubrics are made by the individual universities and so quality and style can vary between courses. Most courses offer free completion certificates (although not all) and some are eligible for Coursera’s Signature Track, which for a fee can give you an enhanced certificate for which Coursera verifies your identity.

**Udacity:** Udacity is another Stanford-born MOOC platform and was founded by Sebastian Thrun, David Stavens and Mike Sokolsky in February 2012. Unlike Coursera, the emphasis was mainly on Computer Science but they have since expanded to include Mathematics, Physics, Business and Psychology courses to its catalog. Unlike Coursera, Udacity courses are all self-paced; you can study as much or as little as you like each week. This can be very valuable for students that have less flexible schedules or less free time. Also unlike Coursera, which has many introductory courses, Udacity courses are divided into three levels: beginner, intermediate and advanced. Udacity create all of their courses themselves and they follow their own unique pedagogy style with strict quality control. The team of instructors, teaching assistants and video editors work closely together to create courses that are well designed and highly polished.

**edX:** Founded in 2012 by MIT and Harvard, edX is the East Coast answer to Stanford-initiated start-ups. Initially, edX offered Engineering, Computer Science and Science courses but they now offer courses on Humanities and Liberal Arts too. Many other universities have now joined non-profit initiative: Berkeley, The University of Texas System, McGill Australian National University, Wellesley, Georgetown University, University of Toronto, Ecole Polytechnique de Lausanne, Tu Delft and Rice.

The nature of edX courses is much more similar to offline college courses than with the other platforms but there is a feeling of great polish and quality for each one, especially with their Computer Science assignments which are very in-depth and interactive. Unlike Udacity, courses are not self-paced and unlike Coursera, the courses usually last between 10-12 weeks as opposed to Coursera’s shorter 5-8 week average.

You may find edX courses more challenging than courses of other platforms because edX’s courses are much more rigorous than Coursera’s and less tolerant in terms of deadlines than Udacity’s. However, the studying process is organized in the way most convenient to the student. It’s consistent: once you learn how to use edX, you won’t have to change your studying approaches and habits due to the new regulations or policies.
**MOOCs:** Massive Open Online Courses (MOOCs) have been one of the most recent developments in the field of online learning. A massive open online course (MOOC) is an online course aiming at large-scale interactive participation and open access via the web. In addition to traditional course materials such as videos, readings, and problem sets, MOOCs provide interactive user forums that help build a community for the students, professors, and TAs. MOOCs are a recent development in distance education.

A massive open online course (MOOC) is a free Web-based distance learning program that is designed for the participation of large numbers of geographically dispersed students. It is an online course with option of free and open registration, a publicly shared curriculum, and open-ended outcomes. A MOOC may be patterned on a college or university course or may be less structured. MOOC is ideal for independent study and users can select courses from any institution offering them. MOOCs build on the engagement of learners who self-organize their participation according to learning goals, little prior knowledge and skills, and common interests. MOOCs are short courses that are delivered online for free.

These courses:

- do not have any entry requirements - all courses can be taken by anyone from anywhere online
- are usually run two or three times each year
- are led by world-class academics and supported by teaching assistants
- typically require 1-2 hours of study each week for around 5 weeks
- are self-directed, meaning you follow the course materials, complete the readings and assessments, and get help from a large community of fellow learners through online forums
- are comparable to a standard University of Edinburgh course in terms of content and study level
- meet high academic standards and are subject to internal quality assurance processes
- Although MOOCs do not have formal university credits assigned to them, Statements of Accomplishment will be available to any learner who completes a course with us.

### 1.2 Scope

“Tools for programming in MOOCs: Assess Student’s Knowledge” aim of this project is to design a Programming tool like compiler that can compile and assess student’s source code to provide instant feedback. A compiler is a programming language processor that works as a translator used to convert high level programming language written by human into machine readable language. Compiler will be used to first compile student’s program then assess using some grading strategy to provide feedback for his programming knowledge.

In context of MOOCs student starts to learn basic programming language like C/C++, needs a programming language processors for e.g compilers, source code editors, debuggers etc. programming language processor like Compiler is used to compile source code and then shows output after successful execution of program. But in MOOCs Student also wants instant feedback for his programming efficiency. A compiler that compiles and assess student’s programming skills would be beneficial for them.

A compiler that can be used to compile C/C++ Programming Language and grade source code using some good grading strategy. Programming efficiency of any programmer
somehow depends upon programming environment or programming tools on which he/she is working. Aim of this project is to design a compiler that can compile as well as assess student’s knowledge to motivate and enhance programming skills. This will lead to successful completion of course in MOOCs if student get instant feedback for his/her source code, he/she will be able to find out his/her mistakes during first stage and try to minimize these mistakes in future.

Assessment process relies on many different conditions or approaches that an assessment tool would implement for assessment procedure. A good evaluation approach is required to assess student’s programming assignment.

1.3 Definitions, acronyms and abbreviations

<table>
<thead>
<tr>
<th>edX</th>
<th>An organisation founded by MIT and Harvard university that develop an open source technology platform to deliver online courses.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOOCs</td>
<td>A massive open online course (MOOC) is a free Web-based distance learning program that is designed for the participation of large numbers of geographically dispersed students.</td>
</tr>
<tr>
<td>Student</td>
<td>Learner starts to learn basic programming language C/C++ in MOOCs.</td>
</tr>
<tr>
<td>Compiler</td>
<td>Programming tool used for converting high level programming language into machine readable language.</td>
</tr>
<tr>
<td>Assessor</td>
<td>Assessment tool used to assess and grade student’s source code.</td>
</tr>
</tbody>
</table>

2. Literature Survey

edX is a massive open online course (MOOC) provider and online learning platform. It is an organisation founded by MIT and Harvard University in May 2012 that develop open source technology platform to deliver online courses. It hosts online courses in a wide range of disciplines to a worldwide audience at no charge. edX is a non-profit MOOC platforms provider and runs on an open-source software platform. Coursera and Udacity are two more MOOC platform provider but they are for-profit organisation [1].

MOOCs (Massive Open Online Courses)

The term MOOC was first used by George Siemens and Stephen Downes to describe a course offered in Canada in 2008. Massive refers to expand capacity of courses to large number of learners [2]. Open refers to providing open education to vast number of participants around the globe regardless of any location, age, income, level of education, ideology, without entry requirements or course fees to access high quality education [2]. Online, term refers to the accessibility of this opportunity from any location in the world via internet connection to
provide synchronous as well as asynchronous interaction between the course participants [3]. **Courses.** In MOOCs it refers to the educational courses to be delivered to the learners. It comprises of OER, learning objectives, networking tools, assessments, and learning analytics tools [4]. MOOCs represent the ultimate democratization of education, by making education more accessible to as many people as possible. Anybody could enrol in this course as it was open online course and more than 2000 people registered to access this course. MOOC didn’t provide credit to online participants, as universities give to their students [5]. Since then MOOC has gained popularity in the USA especially when Sebastian Thrun, a Stanford professor offered an artificial intelligence course for free. In 2013 Gaebal constructed MOOC as a free, credit-less online course where large amount of people can join it without any entry requirements [6]. Basically, any individual with an Internet connection can join a MOOC, access the available resources, interact with other students, reflect and share their knowledge with others [7].

There are two types of MOOCs available: cMOOCs and xMOOCs. Connectivism MOOCs (cMOOCs) is based on connectivism pedagogy. cMOOCs are not prescriptive, and participants set their own learning goals and type of engagement. In cMOOCs students were encourages to create their personal learning environment and network of co-learners. Instructors play a discussion moderator role in cMOOCs whereas in xMOOCs instructor plays a role of tutor. Extension MOOCs (xMOOCs) follow the behaviourism, cognitivist, and (social) constructivism approach, mostly focussed on instructive approach. It is similar to traditional higher education courses, to engage students instructor had a syllabus that typically consisted of readings, discussions via online forums, assignments, quizzes etc. Deadlines for completing tasks and an online form of continuous assessment allow the course administrators to assign marks and credits [5]. According to the current published literature review that focussed on the use of massive open online courses (MOOCs) by students’ and instructors’ reveal some motivations and challenges regarding MOOCs. Identified two major issues that have yet to be resolved- the quality of MOOC education, and assessment of student work [7]. A big challenge is there before MOOCs that is automated assessment of student work: Because of the large number of students participating, it is impossible to manually check the programming assignments of every student, as it is time consuming [8]. When MOOC is designed to teach a skill like programming, automatic assessment of programs is necessary as it contribute to the quality of such MOOC. It provides opportunities to practise programming, which is needed for students to be able, develop programming skills. Detailed feedback is necessary to help students recognising their mistakes, weakness and getting better in future [9]. MOOCs are a young phenomenon and they are still evolving dramatically. Several advantages of using automated assessment for programming assignments have been observed like speed, availability, consistency, and objectivity of assessment [10].

**Tools for programming in MOOCs: Assess Student’s Knowledge**

In today’s world learning computer’s language is necessary. Practice is very crucial step in learning any computer programming course. All computer science students need to learn effective and good programming skills. In universities and in MOOCs (open source, can be accessed anywhere in the world) so huge amount of students enrolled and it’s becoming very tedious tasks for instructors or teachers to assess programming assignments of all students manually. It becomes viable to do manual grading of programming assignments. To address this problem of assessment there is a requirement of an Automated Assessment Tool or
Automated assessment had already been suggested by Hollingsworth, 1960 [12]. Using the system student needs to submit program written in assembly language. This Grading system will run the program and produce two different outputs either “wrong answer” or “program complete”. Advantage of automatic evaluation and instant feedback for novices is they can benefit from early disambiguation and misconception. Automatic assessment tools can be used to support students with automatic feedback for their programming as well as to help teachers in grading tasks. A student takes responsibility for their own learning and learns independently programming language outside the class using self-assessment tools. These tools would give a feedback to orientate student’s knowledge and learning. Many automated tools have been built since first appearance for assessing programming assignments. Automated assessment systems are used and have been proved to be beneficial. Three generation of automated tools have been identified by Douce et.al [13], first one refers to when assessment was made only considering right or wrong answer. Second generation refers to work with tools, which came with operating system, to build new tools. In Third generation the main improvement was done with orientation of using web-based technologies. Automated assessment systems like Quiz-Pack [14], BOSS [15], ALOHA [16], Pass [17], Web-CAT [18], ProgTest [19]. In context of computer science course an automated assessment system has to perform many tasks as submission of program, compilation; any errors at this stage have to be noted and considered during assessment. Integration of Auto grader and e-learning program is quite useful and worthy. An auto assessment tool enriches functionality of e-learning courses like MOOCs by grading programming assignments automatically [20].

Key Features of Automated assessment tools are:

- Supported programming language defines the use or not of the tool. It is important feature when it is considered to make a quick implementation.
- Programming language used to develop the tool. This would be a valuable feature to choose a tool.
- Logical Architecture is a useful feature, when modification of the tool is to be considered. This architecture will show modularity, scalability, and flexibility level. It shows how system could connect with other systems and how different modules work.
- Deployment Architecture shows how the hardware over which tool works is. It shows if current environment is suitable for implementation of a tool.
- Work mode indicates if the tool can work alone; or works as a plugin, when it is supposed to work with any other system like MOOCs.
- Evaluation Metrics or assessment approach would consider which metrics are considered inside the assessment process.
- Technologies used by the tool helps to deploy or build a new tool and for future maintenance as well.

Over the past few years, there are studies that offer solutions for automatic marking of programming codes. Each solution has its own characteristic and mechanism. Assessment process can be done by just looking into structure of code using Structural Testing (White Box), or simply analysing the functional behaviour of a program using Functional Testing (Black Box). But in evaluating program code would require more dynamic methods. Tools use different assessment strategies to grade programming assignments such as: Static and Dynamic Approach. Assessment can be done using different automatic assessment methods, three general automatic assessment methods are discussed in [21] are Software quality metric method, Dynamic testing method and Static analysis approach. To assess student’s
programming ability quickly, and accurately and achieve automatic scoring of programming assignments is involved in acknowledge as artificial intelligence, pattern recognition, natural language understanding, etc.

1. **Software quality metric method** is based on software complexity metrics, or program characteristics (such as numbers and type of variables, code lines, cyclomatic complexity etc.). This method is not good enough as it can’t understand the program structure and program syntax. Software metrics like coupling and cohesion using reference code can also be used to assess student’s program [21]. Coupling and cohesion are certain aspects of a program that measure its relatedness and dependency rather than size and complexity of the program [22].

![Diagram](image)

**Figure 1: Proposed approach.**

2. **Dynamic Testing** method compares students program output with standard answer or model provided. This is the most accurate and direct way to grade the program function. But this does not accord with the traditional manual grading method as there may be some semantic or syntactic errors in student’s program that may score to zero because of no running result [21].

3. **Static analysis** method is based on the basic strategy of program understanding with the manual grading process. It assesses the student’s program according to the degree of the program achieved based on program syntax and semantic. It uses the pattern matching process [21].

Novel marking scheme is the use of combination of Dynamic testing and Static analysis method to assess student’s program code as it will be more close to manual assessment process. It can give reasonable marks to programs with syntactic and semantic errors. AutoLEP an automated learning and examination environment for programming is an automatic assessment tool used combination of dynamic testing and static analysis method. It has been used for C programming language since 2004 at Harbin Institute of Technology [23]. It evaluates by examine how close the source code of student’s program is to correct solutions. It can give reasonable marks and improves student’s learning experience by statically analysing the program. AutoLEP provides an instant feedback and assessment results to submitted programs.
- The “syntactic errors” part points out the syntactic errors in students program.
- The “dynamic testing” part points out whether student’s program passed the test suites or not.
- The “semantic analysis” part outlines the matching results of the student’s program with each model program on size, structure, statement and key

PASS (Program Assessment using specified solutions) is another automated assessment tool used to assess C Programs of students [24]. It compares submitted program with provided solution plan rather than using script based method which verify correct output or program metrics, such as cyclomatic complexity.

Web-CAT is one of the most popular automated assessment tool used by many institutes to assess students program source code [25]. It is plug-in based and evaluation is based on how well student test their own code. Student construct their own test suites to test their code for different test cases and then WebCAT marks students program code on the basis of success factor of test cases that are passed by student’s source code. The tool Supports java, C++, Scheme and many other Programming languages. It also supports manual assessment process.

Web-CAT, the Web-based Center for Automated Testing, is a new tool that grades student code and student tests together. Most importantly, the assessment approach embodied in this tool is based on the belief that a student should be given the responsibility of demonstrating the correctness of his or her own code.

In order to provide appropriate assessment of testing performance and appropriate incentive to improve, Web-CAT should do more than just give some sort of “Correctness” score for the student’s code. In addition, it should assess the validity and the completeness of the student’s tests. Web-CAT grades assignments by measuring three scores: a test validity score, a test completeness score, and a code correctness score.

Web-CAT is an open source automated grading system developed by Stephen Edwards. It grades students on how well they test their own code. Web-CAT utilizes a web interface to provide all of its capabilities. With a web browser, the user can control all submission activities; give feedback, and view results and grading activities. Web-CAT Grades assignments by coming up with three scores, namely the code Completeness, test Completeness, and test validity scores.

Web-CAT offers the ability to detect erroneous student programs and protection from malicious code. It also provides data integrity through the use of system security policies and a relational database. It was written to have a high degree of portability. Packaging under a Java servlet application, it will run under any compatible servlet container, for instance, Apache Tomcat. Besides, Web-CAT itself is completely language neutral. And it has been used to handle submissions in Java, Prolog, Scheme, C++, Standard ML, and Pascal. A downside of Web-CAT is that it is highly geared towards unit testing and thus every assignment is required to have a JUnit test. This makes it very time consuming for instructors to create weekly assignments.
Figure: Web-CAT feedback on a program assignment submission.
**CourseMarker** [26] is a tool developed in Nottingham University. Evaluation metrics of this tool is typography (indentations, comments, etc.), functionality through test cases, programming structure use, and verification in design. The programming languages are java and C++ and it has been built using java. Scalability, maintainability and security are its main advantages.

**CodeLab** [26] works with Python, Java, C, C++, C#, JavaScript, Visual Basic, and SQL and it is also a web based system like Web-CAT. According to the vendor, it was first offered in 2002 to help with teaching multiple programming languages, it is now a seasoned system that has been used in over 350 institutions worldwide and has analysed over 60 million exercise submissions from more than 150,000 students.

**Codecheck** [26] is a “convention over configuration” unit test library for automatic evaluation of student programs. It helps minimize the authoring work of an instructor. Codecheck was developed by Cay Horstmann of San Jose State University. For the instructors, codecheck provides an easy way to make an assignment. Specifically, the author of a problem provides a zip file that consists of an optional problem.html file and two folders named student and solution. In the student folder, codecheck expects an incomplete source code of the problem to display to the student in a textbox on a web page and files that are used for testing the student work (input files, tester programs, library files). And in the solution folder, codecheck expects the solution file. After an assignment is submitted to codecheck, the instructor will receive a link pointing to the assignment. The instructor then distributes this link to the students.

**CodingBat** is a free site of live coding problems to build coding skill in Java and Python. It was created by Stanford computer science lecturer Nick Parlante. The problems on CodingBat are short and provide immediate feedback in the browser. A limitation to CodingBat is that it is only available for Java and Python, not C. There are currently 13 Java problem groups and 8 Python problem groups available on CodingBat. Each problem group has a couple dozens exercises.
List of some former and recent tools developed since 2005[26] given below:

<table>
<thead>
<tr>
<th>Tool's name</th>
<th>Main Features</th>
<th>Supported Languages</th>
<th>Work Mode</th>
<th>Grading Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaSrat</td>
<td>Use of services. LMS integration</td>
<td>Java, Scala</td>
<td>Moodle plugin. Sandalone.</td>
<td>Correctness.</td>
</tr>
<tr>
<td>Autd.LP</td>
<td>Static and dynamic analysis to grade</td>
<td>Sandalone</td>
<td></td>
<td>Static analysis. Dynamic analysis.</td>
</tr>
<tr>
<td>Petcha</td>
<td>Coordination among existing programming support tools. Use of technology for interoperability.</td>
<td>Languages supported by Eclipse and Visual Studio</td>
<td>Sandalone</td>
<td>Based on test cases.</td>
</tr>
<tr>
<td>Jazzess</td>
<td>Moodle integration.</td>
<td>Java</td>
<td>Moodle plugin</td>
<td>Compilation</td>
</tr>
<tr>
<td>Robol.IFT</td>
<td>Grading mobile applications. GUI grading.</td>
<td>Java</td>
<td>Sandalone</td>
<td>Unit testing (public and private)</td>
</tr>
</tbody>
</table>

Table1. List of Former Auto assessment tools
Table 2. List of Recent Auto assessment Tools

<table>
<thead>
<tr>
<th>Tool's name</th>
<th>Main Features</th>
<th>Supported Languages</th>
<th>Work Mode</th>
<th>Grading Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course Marker</td>
<td>Scalability, maintainability, Security, configurability, Plagiarism detection, Work with levels of feedback</td>
<td>Java, C++</td>
<td>Standalone</td>
<td>Typology, Functionality, Structures use, Objects design, Objects relations</td>
</tr>
<tr>
<td>Marmoset</td>
<td>Scalability, maintainability, Security, configurability, Plagiarism detection, Work with levels of feedback</td>
<td>Any language</td>
<td>Standalone</td>
<td>Dynamic and static analysis.</td>
</tr>
<tr>
<td>Virtual Programming Lab</td>
<td>Moodle integration, Customizable grading mode, GNU GPL license, Plagiarism detection, Configurable activities, Jail environment</td>
<td>Ada, C, C++, C#, Haskell, FORTRAN, Java,Octave, Pascal,PHP, Prolog, SQL, Ruby,Python, Scheme, Vhdl.</td>
<td>Moodle Plugin</td>
<td>Correctness based on test cases. Open for new methods.</td>
</tr>
<tr>
<td>Grading Tool (Magdeburg University)</td>
<td>Use of services, Configurable evaluation process.</td>
<td>Haskell, Scheme, Erlang, Prolog, Python, Java</td>
<td>IMS extension.</td>
<td>Compilation. • Execution. Dynamic tests.</td>
</tr>
</tbody>
</table>

In MOOCs a programming tool which can be used to assess students programming knowledge would be beneficial. An automated assessment tool for MOOCs based on dynamic testing and static analysis approach would help students to self-assess program code. Instant feedback is required by students on their programs for their improvement. Assessment provides both means to guide student learning and feedback for both the learner and the teacher about the learning process from the level of a whole course down to a single student on some specific topic being assessed. Students often direct their efforts based on what is assessed and how it affects the final course grade. Continuous assessment during a programming course ensures that students get enough practice as well as get feedback on the quality of their solutions.
2.1 References

20. Draylson Micael de Souza, Jose Carlos Maldonado, and Ellen Francine Barbosa “ProgTest: An environment for the submission and evaluation of programming assignments based on testing activities” In Software Engineering Education and Training (CSEE T), 24th IEEE-CS Conference on. Institute of Electrical and Electronics Engineers, Piscataway, NJ, 1–10, 2011


3. Overall Description

3.1 Product Perspective

Programming tools like compilers can be used as assessment tool to assess student’s knowledge. Compiler is a software translator which accepts, as input, a program written in a particular high-level language and produces, as output, an equivalent program in machine language for a particular machine. In context of MOOCs, student starts to learn programming language obviously needs programming tool like compiler for compilation of program code. Efficiency of any programmer and programming language is always somehow depends on programming environment and compiler or ide being used at the time of programming. This automated system is primarily intended to a Student for assessing their programming knowledge in MOOCs when they start learning basic programming language C/C++ as a novice programmer. This programming tool will be used as compiler and assessment tool to assess student’s program code in MOOCs. It should be platform independent and easy user interface. This system will consist of three main parts: one is Programming tool (compiler), second assessor and third MySQL database. Compiler is used to translate student’s source code written in human readable language into machine language. If compilation is successful then assessment process will start and finally result with feedback on student’s program source code is returned.
3.2 Product function

This programming tool is used to assess student’s knowledge will help novice learners who starts to learn programming language through MOOCs. This automated system is primarily intended to a Student for assessing their programming knowledge in MOOCs when they start learning basic programming language C/C++ as a novice programmer. This programming tool will be used as compiler and assessment tool to assess student’s program code in MOOCs.

The primary goal for this is to have the system where users can create Courses, assignments, submits the assignments. The system should automatically grade the submitted assignments by taking some parameters required for it e.g. compilation command, test-cases etc. from teacher. In IITB, currently, Moodle is used for creating courses, creating assignments, submitting assignments; it lacks the feature of automated grading. Also, event handling requires the whole page to refresh, which makes the process less user friendly. Our proposed system needs to take care of all these issues.

The ability to write a workable program using high-level programming languages is a definite requisite for college students who take computer science courses. Paperless exams, in which students write answers to programming questions and submit their answers online, are an ideal testing method. There are a number of testing systems, such as the systems used in ACM’s Intercollegiate Programming Contest, Online Judge used for competitive programming and programming courses in the National University of Singapore, TRY system, Scheme- Robo system implemented in the Helsinki University of Technology for automated assessment of exercises in the Scheme functional language, PASS, and NCRE system implemented currently in a nationwide computer ranking examination in China, etc.

Manually grading programming assignments is a difficult and time consuming task. It is difficult to score fairly and easy to make mistakes. The automated grading approach can not only overcome these barriers but is also a key feature to implementing paperless exam systems.

A grading system checks on whether a program can execute and give the correct output in any legal input case. The student’s program should be able to perform its basic functions first before other components can be verified.
4. Analysis Model

4.1 Use Case Diagram
4.2 Flow Chart

This flow chart shows the complete process of assessment of student’s programming assignment. Two processes will be performed first student submit his/her program source code in a compiler. Compiler will convert human readable programming language into machine readable language; if compilation would be successful then source code will go for evaluation process.

Source code will be tested against suite of test cases i.e. dynamic testing if output of student’s source code and given test cases are same then grading would be done but it is not necessary that output style of student’s source code will be same as standard output so dynamic testing will lead to zero marks for student’s source code. To escape from this condition we will go for static analysis method, which does not require actual execution of the program source code. In static analysis method we will use different approaches like count of number of variables, program structure, pattern matching etc. These strategies will be used for grading student’s source code and according to the amount of matching grading will be done. My approach of evaluation would be combination of both dynamic testing and static analysis method.

The primary goal for this is to have the system where users can create Courses, assignments, submits the assignments. The system should automatically grade the submitted assignments by taking some parameters required for it e.g. compilation command, test-cases etc. from teacher. In IITB, currently, Moodle is used for creating courses, creating assignments, submitting assignments; it lacks the feature of automated grading. Also, event handling requires the whole page to refresh, which makes the process less user friendly. Our proposed system needs to take care of all these issues.
5. Work done

The work carried out till now has been theoretical with very less focus given to implementation and its details. A programming grader is a tool that can assess student’s knowledge by grading students programming source code after successful compilation using a good grading strategy. Compilers that are platform independent and free downloadable so as it should be easy for a beginner student to easily download a good compiler for C/C++. Compiler will compile student’s program source code, if code will be compiled successfully then after code will be graded or evaluated by automatic grader, using assessment strategy of dynamic testing and static analysis method. Grading process should be consistent and close to manual grading process. I did literature survey and studied about so many existing autograding tools like how they work. They includes so many features but lacking some features that can be appended to them in future. Techniques that have been implemented in many autograding tools are Static analysis and dynamic testing. Aim is to extend traditional dynamic analysis methods and integrate them with other methods of autograding viz. static analysis methods and software quality metric methods.

6. Conclusion

MOOCs offer university-level courses without the need to complete an entire programme of studies. They are ideal for independent study and users can select courses from any institution offering them. Video-based, they offer interaction either through peer review and group collaboration or automated feedback through objective, online assessments (including quizzes and exams). Most courses are free, some are fee-paying Drop-out rates are high – up to 90%. Rates are marginally lower for paid-for courses. A reasonable degree of computer literacy is needed. Many MOOC users are graduates seeking to top up their skills and competences. They are low cost routes to accessing quality courses. They can be combined with other study or work. They are accessed from any computer at any location including your home. Automated assessment is very positive, indicating a significant potential for increasing the quality of student code. We plan to apply this technique in our introductory programming sequence, where students will program in C and C++ introductory languages.
7. Implementation and Future Work

A compiler or programming tool is used for learning programming skills by a beginner. Programming efficiency of any programmer anyhow depends on the programming environment in which he/she is working. So a student requires a programming tool that can compile his/her program as well as assess it instantly. Self-assessment will help students to enhance their programming knowledge and diversifies their programming skills.

In future we will plan, how to modify and extend traditional Dynamic analysis methods and integrate them with other methods of automatic grading viz. static analysis and Software quality metric method. In the next stage of the project the implementation of the proposed approach shall be carried out as an extension to traditional Dynamic Analysis approach. Cyclomatic complexity and dynamic testing with some static analysis grading strategies will be incorporated in this compiler cum assessment tool. This tool will work for C/C++ languages. But in future we will enhance this tool for other languages also.

It will work as self-assessment tool for students who will start as a beginner as well as those who wants to diversify their programming skills and basic programming skills.