

Implementation of an Automated Programming Lab

Xue Bai, Ade Ola, Yingjin Cui
Virginia State University^{1,2}, The Governor's School @ Innovation Park³
USA

Abstract

The most effective methods of learning in programming courses depend on practical exercises. However, preparing and collecting programs manually takes time. Manual grading of programming assignments is also a tedious and error-prone task. This paper describes the implementation of an automated programming lab consisting of devices that would simplify the programming assignment management process, automate grading of programming assignments, and provide real-time feedback as students work to implement solutions to programming projects. The system has been used in actual courses at our university. With the system, instructors were able to assign more homework and provide quicker feedback to students.

1. Introduction

One of the primary functions of an academic institution that teaches computer programming courses is to imbue its students with the ability to program. It has been shown that the most effective methods of learning in programming courses depend on practical exercises [1]. Lab work, homework and project assignments provide ways for students to translate their theoretical knowledge into designing computer programs and to sharpen their programming skills. But, usually, computer programming instructors cannot handle large classes, and certainly not classes across campuses, because the process of collecting students' program files for evaluation is a tedious and error-prone task. Typically, programming assignments are distributed through printouts or via Internet (perhaps via a Learning Management System (LMS) or email). Upon completion, students submit their work via email or upload them to an LMS. The instructor then downloads submitted program files for compilation and testing. The manual process works with small number of students in a class or cohort; however, when the number of student increases, productivity decreases and human related errors tend to increase on the instructor side.

Grading of programming assignments is also challenging. Programming assignments are notoriously difficult to grade manually in a fair and timely manner. In most cases, one programming assignment would contain multiple files to be

downloaded into a particular directory in order to run and grade the program assignments. Matters become worse when some programs may not run properly after downloading. For some programming assignments, students are required to interact with databases, external files, services provided by a remote server, or external APIs. Some project assignments require student collaboration and integration of individual programming components into one integrated system. To support these activities, appropriate programming environment must be setup on lab computers. Institutions can configure programming labs to meet the requirements for the courses offered. However, students often prefer to setup their own computers with their preferred software so that they can work on assignments and projects at home and at any time that suits them, rather than on college lab computers which have restricted hours.

Based on our experience, students usually experience numerous problems when trying to configure their own programming environments and often create incompatibilities with university-configured software environments. For example, for the Java programming courses, some students would struggle a few weeks to set "class path" and other system environment variables in their own computer. These types of problems, even though they can enhance the learning experience, distract students from their class assignments and projects which are the main focus of a programming course [4]. Another challenge in computer programming instruction and learning is the issue of providing timely feedback to students. It is difficult to provide students with timely assistance and feedback beyond the class meeting times. Quite often, students are not able to determine whether they are on the right track when devising solutions to a given homework problem.

The motivation for designing the system hereby presented is to resolve these problems and enable students to focus on programming assignments and have a more effective learning experience. The system is a web based assignment and grading software, which provides a solution that simplifies the programming assignment management process, automates grading of programming assignments, and provides real-time feedback as students work to implement solutions to programming projects. The automated programming lab eliminates the need to download and upload files. The system also eliminates

the need for setting up programming environments, and students' programming assignments can be accessed simultaneously by instructors through the web application. The system provides a centralized location for all programming assignments. Students can work directly on the assignment folder when it becomes available, and all files in their respective assignment folders are editable until the assignment is due. After the due date, students are still able to view the files they created in the assignment folder, but they can no longer modify the files. On the other hand, instructors can access all files in the students' assignment folder. Instructors can test a student's program and make comments directly on the program files to provide instant feedback. This feature has proven to be an efficient feedback system when human intervention is required beyond the automatic feedback mechanism.

There are many online submission systems that allow documents to be distributed and collected online, which have alleviated the tasks of instructors in several ways. However, most of the systems could not meet all required criteria for an ideal submission system [7]. In particular, other systems require that relevant files be downloaded to a local computer in order to run and test programs.

2. Challenges of teaching programming

This section provides an expanded discussion on the challenges involved in implementing an automated programming lab and gives a comparison of our approach with current research and systems.

2.1. Software installation and compatibility issues

When students have to install software packages that are required for specific programming courses on their own computers, they can sometimes experience various installation problems, lack sufficient hardware resources, or run into compatibility issues [4]. Software packages sometimes require the installation of additional software components at a specific version level which can lead to greater installation complexity or a failed installation. Furthermore, periodic software auto-updates that may occur during the duration of the course can sometimes interfere with the installed software, which in turn may require reinstallations on systems in the college labs and on the student's personal computer. When students setup their own development environments, mismatching of the software versions or differences in system configuration may lead to incompatibility issues when the instructor attempts to run the students' code in different settings. Programs that appear to work on the student's computer may fail on others. These types of problems, though they may provide some learning experience, distract students from their class assignments and projects, which are the main focus of a programming course.

2.2. Grading assignments

The nature of programming assignments makes them perhaps the most difficult type of assignment to grade. The labor-intensiveness of grading programming assignments is the main reason few such assignments are given to students each semester. As the number of students increases, the number of homework given tends to go down. Of course, reducing the number of programming assignments can diminish the effectiveness of the course.

An automated programming environment must include an auto-grader and a system that provides instantaneous feedback. There are three main components involved in the grading of programming assignments [1]: Correctness, Efficiency, and Maintainability. Given any input, a computer program is considered correct if it produces the desired output. This may include the ability of the program to handle illegal input (by exception handling and/or displaying the appropriate error message, which is part of the desired output). A program is efficient if it performs its tasks without consuming too much processing time and memory space. It is maintainable if the code is easily understandable and employs devices such as descriptive variable names, comments, indentation and modular programming. However, in practice, efficiency and maintainability are seldom used in evaluating students' work. Obviously, the most important component when grading a programming assignment must be its correctness, since a program must first be able to perform its basic functions before it can claim to have satisfied the requirements of the assignment. Most automated grading systems focus mainly on program correctness.

However, there is no effective automated grading system available when a program submitted by a student is not working as expected. In such a situation, one challenge is that there are many different ways to solve a given problem. It can be quite difficult to figure out what type of solution the student is trying to devise and what's wrong with it. A research paper [5] discussed a new system that will identify the minimum number of corrections necessary to get a program working, no matter how unorthodox the programmer's approach. Another published work [6] described error models in terms of correction rules that reduce the problem of finding minimal corrections in an incorrect program to the problem of synthesizing a correct program from a sketch. The authors reported that their results show that relatively simple error models can correct, on the average, 64% of all incorrect submissions in a benchmark set. However, as stated in the paper, this approach did not work well when there are conceptual errors or structural requirements.

2.3. Assessment and feedback

The two methods most commonly used in systems that provide feedback on programming assignments

are: (i) test-case based feedback and (ii) peer-feedback [3]. In test-case based feedback systems, the student's program is run on a set of test cases and the failing test cases are reported back to the student. The feedback of failing test cases is however not ideal, especially for beginner programmers who find it difficult to map the failing test cases to errors in their code. The peer-feedback is helpful, but it is not without problems. Peer-feedback has some inherent limitations, such as the time it takes to receive quality feedback and the potential for inaccuracies in feedback, especially when a majority of the student are themselves struggling to learn the material [3]. Students who have difficulties finding the preliminaries of an assignment, especially freshmen, often struggle to create a practical solution for an assignment. While working on the assignment, students are not sure if they are on the right track or not. When students need instant help in order to continue, it is hard, if not impossible, for them to get immediate and effective assistance from instructors because both the student and the instructor usually do not have a good way of distributing and accessing the relevant files. These problems do not contribute positively to the learning process; they only reduce the productivity of a course and ultimately the learning process [2].

2.4. Current Systems

There are a few automated grading systems in the literature or on the market. Moodle [8], one of the most prominent LMS, provides a plug-in called Virtual Programming Lab (VPL), which requires a dedicated execution server. The server runs supplied test scripts on programs submitted by the students. However, Moodle lacks the tools for compiling and executing programming assignments [9]. Besides, the system requires a significant amount of work before an assignment can be released. The instructor must spend time preparing an assignment, writing a solution program, generating scripts that will evaluate the program, testing the complete setup, and incorporating all these components as a new VPL Activity in Moodle [10]. Codebat [11], Betterprogrammer [12], and Practice-it [13] are commercial tools which provide users with instantaneous feedback. Both CodingBat and Practice-it permit the user to re-attempt problems until the correct solutions are achieved. Another good feature of the CodingBat and practice-it is the authoring facility where users can contribute their own programming problems to the existing set. However, these systems can handle only small pieces of code; they are not suitable for managing large-scale programming assignments. There are other categories of tools for conducting programming examinations online. One such tool [14] allows Java programming exams to be conducted securely online. However, none of the systems reviewed effectively addresses all the issues relating to management and grading of

programming assignments on a large-scale and most effectively.

3. Proposed Devices and System for Auto-grading and Instantaneous Feedback

The purpose of providing feedback is to enhance the learning process [15, 16]. Feedback is one of the most powerful influences on learning and achievement [17]. Thematic analysis [17] revealed five major themes: student involvement or individualization, gentle guidance, being positively constructive, timeliness, and future orientation. Effective feedback is proposed to improve the learning process by reducing discrepancies between current understandings/performance and a desired goal [15, 16 and 17]. To achieve this goal, the feedback provided must actually be reviewed, studied, and used by the student to revise the submitted assignment. However, it is commonly reported that students do not read the instructor's feedback comments [15], especially when the feedback does not come in a timely manner. To ensure that feedback is being used to improve students' programming skills, it must be delivered in a timely and actionable manner. But the reality is that programming assignments are still managed manually in most classrooms; hence it may take a few days, or even a few weeks for graders to grade and provide feedback to students. By the time a student receives feedback, he or she could have forgotten the details of the program in question. The student may simply ignore the feedback. Furthermore, for feedback to be effective, it must be concrete, specific, and useful. It should provide actionable information which students can understand and follow to revise their work and reduce the discrepancies between expected outcome and the output of their program.

3.1 Proposed Methods for Instant Feedback

The proposed system provides among other advantages, instant feedback. Generally, there are three types of feedbacks: compiling errors, runtime errors, and testing cases. Compiling errors are generated by the compiler. However, the error messages are often difficult to understand and not user-friendly for most students in the introductory programming courses. The following devices have been implemented to provide feedback in real time.

3.1.1 Post-processing of compiler errors. In the proposed system, compiling errors are processed further so that the error information is displayed within the code where the errors occur. Students can easily spot the errors in the code and make corrections accordingly.

3.1.2 Correcting one error at a time and recompile. In some situations, compilers are not very good at detecting syntax errors. Frequently, a single intangible mistake will cause the compiler to issue a cascade of

messages. The system provides additional error messages so students can work on fixing errors step by step. The in-built logic is to fix the first detected error and then recompile.

3.1.3 Templates for coding one test case at a time.

In software engineering, a test case is a set of conditions under which one determines whether an application or software system (or one of its features) is working in accordance with its original design. Our system uses the JUnit packages for assessing each method in a program. JUnit, a unit testing framework for the Java programming language, is one of a family of unit testing frameworks in test-driven development. The system provides templates that use the paradigm of the testing framework: “first test then code”. In practice, a set of test cases will be provided for each assignment. Students start with the simplest case when they start coding, and gradually move towards the more complicated cases until meeting the full requirements of the assignment.

3.1.4 Method evaluation and syntax checking.

When grading assignments, if a java program compiles successfully, most instructors use output similarity as the basis for awarding grades. We have built a system that goes beyond just checking for output similarity: If a program failed to compile or have runtime errors, the system applies method evaluation and syntax checking and generates a report that can be used to award partial credits. In that case the human grader has to be involved.

3.2. Treatment of compiler errors

Compiling errors are generated by the compiler. However, the error messages are hard to understand for most students in the introductory programming courses. For example, the following is the standard output generated by a compiler for program (A.java) in shown in figure 1.

```
A.java:16: error: missing return statement
)
^
A.java: error: variable highestGrade might not have been initialized
    System.out.println(highestGrade);
                        ^
2 errors
```

Figure 1 shows how our system processes the compiling errors and displays the error information within the code where the errors are from. Students can easily spot the errors in the code and make correction accordingly.

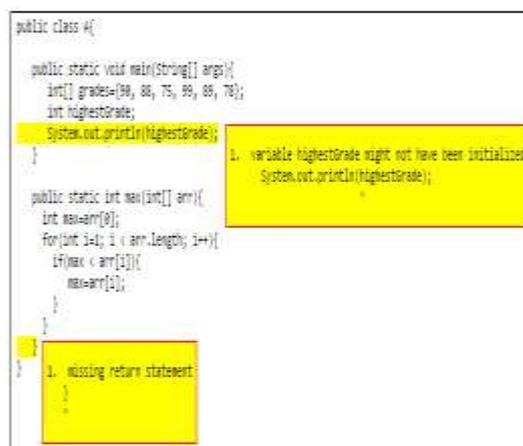


Figure 1. Processed error messages

In some situation, compilers are not very good at detecting syntax errors. Frequently, a single small mistake will cause the compiler to issue a cascade of messages. For instance, there are two errors in the example displayed in the next paragraph: the variable *k* in program *B.java* is not declared and the closing brace of the method *foo()* is missing.

```
B.java:5: error: illegal start of expression
Void bar() {
^
B.java:5: error: ';' expected
Void bar() (
^
B.java:8: error: reached end of file while parsing
}
^
3 errors
```

As shown, the compiler overacts on the missing brace; on the other hand, it did not catch the undeclared variable. The error messages are not too helpful in finding the real cause of the errors. When that happens, students are often confused where to get started in order to fix the errors, because the error messages did not tell any information about the missing brace where all of the errors are originated. Our system provides better user-friendly messages as shown in Figure 2. The system issues error messages so students can work on fixing errors step by step. Multiple errors may be connected with a single mistake, as show in Figure 2; therefore, fixing one error may eliminate all related errors. Our logic is to fix the first detected error and then recompile. In this example, after adding a closing brace to the *foo()* method then recompile, a new message (displayed in figure 2) provides a guide on how to fix the problem of the undeclared variable.

```

public class B{
    void foo() {
        k=0;
    }
    void bar() {
        int m = 20
    }
}
    
```

1. cannot find symbol
symbol : variable k
location: class B
k=0;
^
1 error

```

1 public class B{
2     void foo() {
3         k=0;
4     }
5     void bar() {
6         int m = 20
7     }
8 }
    
```

1. A closing brace in the method before here may be missing.

Figure 2. Processed compiling errors

3.3. Use of Test Cases and Method Evaluation

In software engineering, a test case is a set of conditions under which one determines whether an application or software system (or one of its features) is working in accordance with its original design. The grading system uses the JUnit packages for assessing each method in a program. JUnit, a unit testing framework for the Java programming language, is one of a family of unit testing frameworks in test-driven development.

Our system provides templates that use the paradigm of the testing framework: “first testing then coding”. In practice, a set of testing cases or data file are provided for each assignment. Students start with the simplest case when they start coding, and gradually move towards more complicated cases until meeting the full requirements for a particular assignment. Let us consider an assignment which requires that students write a code to count how many times a word appears in a data file. The first testing data file may contain only one word. The code simply reads in the single word file and uses “if” statement to test whether the word is the same as the one in question. If the output differs from the expected result, the feedback “You may want to consider using if statement to test whether the word is the same as the one in the data file” is generated and sent back to the student. The second testing data file contains multiple words and students are expected to use a loop structure to test each word in the data file; and the third data file may contain some words with mixed cases, or a word interspersed with punctuations. This process helps the student to make progress until all testing cases are tested.

When grading assignments, if a java program compiles successfully, most instructors use output similarity to evaluate the whole program. The instructor may choose not use test cases and/or syntax checking for more detailed evaluation. However, in our case, if a program failed to compile or have runtime errors occur, we have chosen to use method evaluation and syntax checking so as to be able to award partial credits. This process requires grader involvement. However, since most students’ work can compile and run properly with testing cases or data file, only a few submissions need be graded manually. Figure 3 illustrates the auto-grading policy for our courses. Solid lines represent the recommended auto-grading methods, and the dashed lines represent optional methods

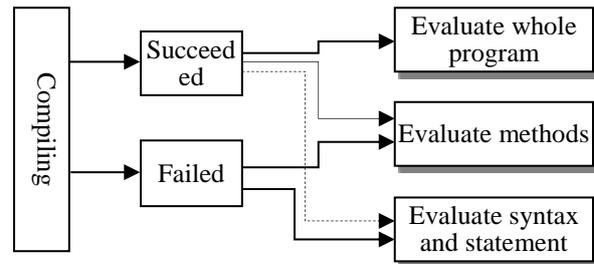


Figure 3. Auto-grading Policies

In the following discussions, we present how an assignment was created and assessed with testing cases. In this assignment, students were required to write a method named findMaxIndex, which takes an integer array as its argument and returns the index of the largest value in the value. If the largest value has multiple occurrences in the array, the maximum index of the largest value is returned. If a null value is passed as its argument, it returns -1 indicating no values in the argument. Figure 4 shows the sample solution to this problem. The code between //CODE START and //CODE END will be replaced with student’s code. Figure 9 shows the testing cases that are used to test student’s method.

```

public class Lab5{
//CODE START
    public int findMaxIndex(int[] arr){
        if(arr==null){
            return -1;
        }
        int max=arr[0];
        int index=0;
        for(int i=1; i<arr.length; i++){
            if(max<= arr[i]){
                max=arr[i];
                index=i;
            }
        }
        return index;
    }
//CODE END
}
    
```

This part will be replaced with students' code

Figure 4. Coding template with sample solution

As shown in Figure 4, the coding template provides a sample solution for testing purposes when an instructor is designing the test cases for this assessment. The code between the coding marks //CODE START and //CODE END will be replaced with each student's code. Student's code will be injected into the program, and the method will be tested against testing cases as specified in Figure 5. Figure 6 shows the editing tool for students to work on this problem. Students will get instant response which tells them whether their codes pass each testing case or not. Figure 6 shows how the methods were tested. For example, Table 1 shows the method was tested using three integer arrays and a null value. The three integer arrays are:

[1, 2, 3, 4, 5], [1, 5, 3, 4, 5], and [1, 5, 3, 4, 4], respectively.

Instructors may set a limit on the number of times a student can submit their code. Students are allowed to submit 5 times in the example in figure 6. In this case, the student tried twice.

Table 1. Testing result for three arrays and null vale

Testing Results				
No.	Name	Expected	Output	Message
1	[1, 2, 3, 4, 5]	4	4	Pass
2	[1, 5, 3, 4, 5]	4	1	Failed
3	[1, 5, 3, 4, 4]	1	1	Pass
4	test4(Lab5Tester): null			Failed

If a student has no clue on how to fix a failed testing case, the information displayed in the table is not very helpful. However, as part of the test case, instructors may choose to provide a hint for each failed case. For example, if a student's code failed the second case, it is more likely that their methods did not consider multiple occurrences of the largest value in the array. In this case, a meaningful hint may be:

“There are multiple occurrences of the largest value in this case. This method is supposed to return the maximum index of the largest value.”

or

“The largest value in this array is 5. But the value 5 appears in the array twice in the index of 1 and 4, respectively. This method is required to return the largest index of the largest value if it occurs multiple times. So this method should return 4, instead of 1”. Most students were able to follow the hint and revise their code to handle multiple occurrences of the largest value.

Computing and information technology educators have begun to call for the introduction of test-driven development into the curriculum [18]. Current systems generally depend on compiling and code execution. One of the systems reviewed allows a student's program to be checked against given test

cases. Our system provides two modules for automated grading; the first is based on JUnit package (as demonstrated in figures 4, 5 and 6) and the other module is based on expected output, in which the expected output and the result from the student's program are compared for similarity. The second module is basically designed for fill-in blank or short answer problem. Both automated systems automatically generate feedback and record grades to gradebook.

```

import org.junit.Test;
import org.junit.*;
import java.io.*;
import java.util.*;
import static junit.wholelearning.MyJUnit.*;

public class Lab5Tester{

    @Test(timeout=1000)
    public void test1(){
        Lab5 lab5=new Lab5();
        int[] arr={1,2,3,4,5};
        assertEquals(4, lab5.findMaxIndex(arr), Arrays.toString(arr));
    }

    @Test(timeout=1000)
    public void test2(){
        Lab5 lab5=new Lab5();
        int[] arr={1,5,3,4,5};
        assertEquals(4, lab5.findMaxIndex(arr), Arrays.toString(arr));
    }

    @Test(timeout=1000)
    public void test3(){
        Lab5 lab5=new Lab5();
        int[] arr={1,5,3,4,4};
        assertEquals(1, lab5.findMaxIndex(arr), Arrays.toString(arr));
    }

    @Test(timeout=1000)
    public void test4(){
        Lab5 lab5=new Lab5();
        assertEquals(-1, lab5.findMaxIndex(null), "null");
    }
}
    
```

Figure 5. Testing cases

4. Web-based assignment management system

The assignment management system provides a stable software development environment for students' programming assignments and projects. It uses a remote application server to compile and run students' solutions to programming assignments. The system is a pure web-based application that is accessible from anywhere and from any type of browser. The programming environment that is configured and installed on the remote server includes java runtime environment, compiler, databases, externals files, services, and external APIs. The configuration sets up an environment where all students have access to identical software development environment from any location where

there is Internet connection. The common programming environment spares the students of the tedious and, sometimes frustrating, task of configuring individual computers. Students require only Internet access to begin work on assigned assignments and projects. The system consists of four basic modules: Editing tool, file management, assignment management and file sharing.

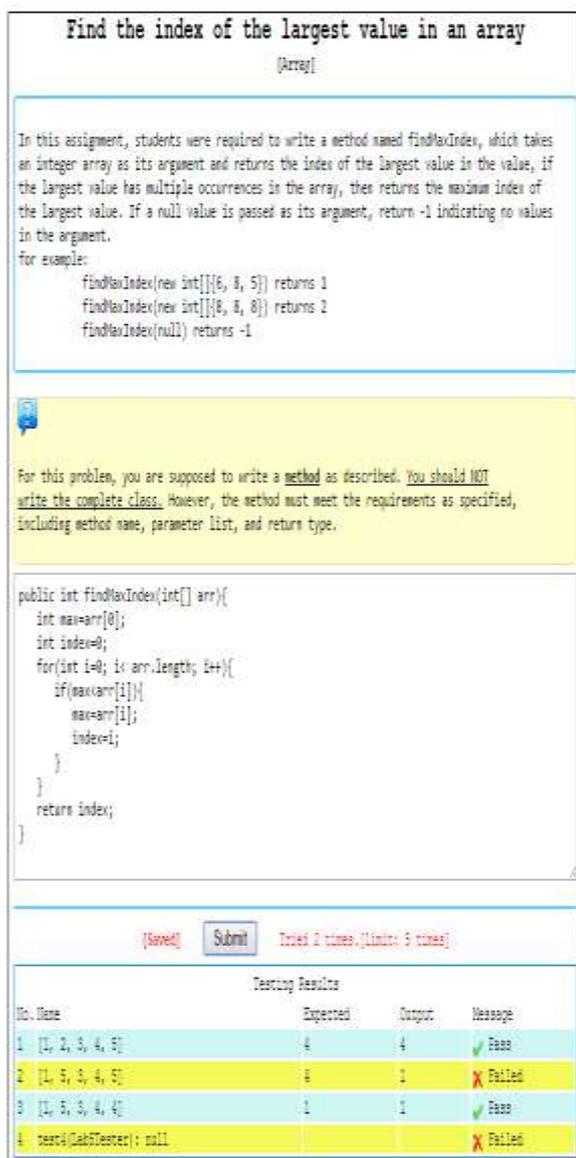


Figure 6. Editing tool for programming assignment

4.1. Editor tool

Integrated Java Development environments often require a high amount of computing resources, which may not be feasible on students' personal computers. Furthermore, there are hardly any free systems available that can be used to manage java programming courses in a distributed fashion. Thus, we embarked on developing the Java-Editor to allow students to work at home or from anywhere with

Internet access. The editor is a cloud-based Software as a Service (SaaS) online IDE which is available through a simple Web browser Internet connection. All that a student needs is an Internet-accessible device for virtually ubiquitous document management. As such, students can work on the programming via any portable devices, including smart phone, iPad, surface, and any internet ready devices. The editor provides an interface where users edit, compile, and run their java programs.



Figure 7. Editor Tool

Figure 7 shows the main windows of the Java-Editor for a student account; the toolbars and other input fields in the middle of the window allow the student to write, edit, compile and debug java programs. The editor tool simulates the command-based environment, where one may specify the arguments for the main method, read data files, or get input from keyboard. In addition to the program editing and running environment, the system provides many code templates to speed up the program development process. Let us use encapsulation to illustrate this feature of the editor. Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Those methods are called accessor (getter) or mutator (setter) methods. It is a tedious coding job if a class contains many such instance variables because for each of them an accessor and a mutator method should be provided. With the code template, all those accessor and mutator methods are automatically generated when a new Java class file is created. The code templates not only expedite the development process but also provide template codes with uniform coding convention style. For example, by convention, all accessor methods start with "get" followed by a variable name with the

first letter is capitalized and without any parameter. On the other hand, all mutator methods starts with “set” followed by a variable name with the first letter is capitalized and with the same data type of the variable as its parameter. The use of templates helps the beginner programmer adhere to these error-prone coding practices. The templates currently provided include file manipulation (reading/writing text and object files), inherited methods, generating JUnit testing cases, GUI components, and event handing.

4.2. File Management

This online IDE provides cloud-based SaaS document management which is available through a simple Web browser Internet connection. The Cloud is available anytime, anywhere. All that the students need is an Internet-accessible device for virtually ubiquitous document management. Document sharing and collaboration are much easier and accurate with cloud-based DMS. The cloud-based file system interface (Figure 8) provides each user with a directory where he or she can manage files. Users can create, rename, and delete directories and files. User directories are private; no one else has access to the files and directories. Through the upload utility, users can upload files to the system from their local directories. The system also provides a backup tool, which allows a user to download files or an entire directory to local storage. Documents stored in the system are easily shared and accessed via SaaS cloud-based DMS. As such, users can log in to the system from any computer with Internet access to gain access to their files. As a result, students can work on programming work without dealing with installation issues and all the other issues discussed in section 2.

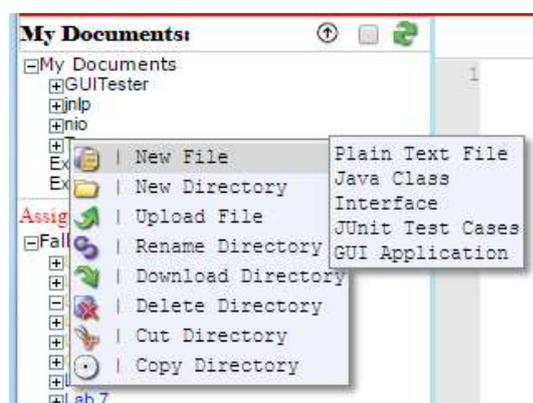


Figure 8. File management tool

4.3. Assignment Management

There are many online submission systems that allow documents to be distributed and collected online, which have alleviated the tasks of instructors in several ways. However, most of the systems could not meet all required criteria for an ideal submission system [6]. In particular, for programming assignments, all files that are relevant to the project

must be downloaded to the local computer in order to run and test the program. In addition, instructors usually distribute some precompiled java classes that are needed to complete programming assignments. Those class files are posted along with the assignments, and students have to download them in order to start working on their homework assignments. Once completed, students upload all files to the online system before the due date. Providing solutions to these problems was the motivation for implementing an online submission system with features that are generally desired.

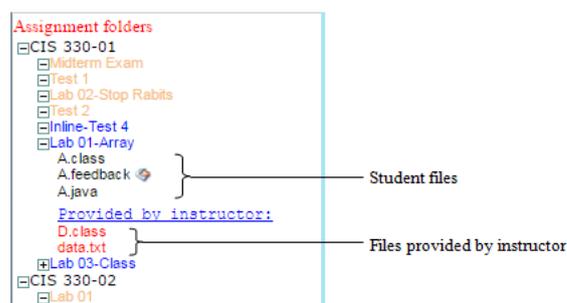


Figure 9. Assignment Management

Our web based online assignment system eliminates the requirements to download and upload files. The system consists of modules where students can directly access assignments and all shared files after the instructor has posted an assignment (see Figure 9). Students can work directly in the assignment folder when it becomes available, and all files in the assignment folder are editable until the assignment is due. After the due date, students are still able to view the files they created in the assignment folder, but they can no longer modify the files. On the other hand, instructors can access all files in the students' assignment folder. If students have questions about their work, instructors can test their programs and make comments directly on their files to provide them with instant feedback. This feature has proven to be the most efficient feedback system for programming assignments.

4.4. File Sharing

File sharing is the practice of distributing or providing access to digital media to be used in a programming assignment. Shared files may include text, audio, image, video, and any other types of files to be processed in a program. File sharing may be achieved in a number of ways. Common methods of storage, transmission and dispersion include manual sharing utilizing removable media, centralized servers on computer networks, World Wide Web-based hyperlinked documents, and the use of distributed peer-to-peer networking. In an academic environment where LMS is available, files are most often distributed through a LMS system where instructors upload files for distribution and students download those files. However, these methods are not adequate if instantaneous file sharing and access is desirable. For example, for all programming courses, instructors

need to demonstrate codes in class to explain theoretical concepts or distribute files to the whole class. Students like to keep the sample code, but taking notes may distract them from focusing on the lecture.

In our system, each course section has a designated folder which is accessible by all students enrolled in that class; instructors can use this directory to demonstrate programming codes in class or online and students have instant access to these files. File sharing makes the task of accessing the desired files easy, convenient, and instantaneous. In the system presented, file sharing is implemented through both assignment directories and designated directories. Both directories are accessible by all students enrolled in that class; instructors can use this directory to demonstrate programming codes in class or online, or distribute auxiliary class or data files in order for students work on coding assignment. This feature makes file distribution and sharing extremely easy. Files placed in these folders are accessible to all students so they no longer need to download or store files on their own computers.

5. Preliminary Experimental Data

In order to evaluate the effectiveness of the testing cases assessment method for programming assignments, we used three programming assignments with different level of complexity from an introductory java programming course. Students completed the three assignments using the system and their codes were assessed with the auto-grading tool. There were 26 students in the evaluated section. For each assignment, we randomly selected three students and downloaded their work to a flash drive and have them evaluated manually by two other professors. Then the grades assigned by professors and auto-grading tool were compared and analyzed. The first assignment included two files, a data file, and a partially completed program file; students were asked to write a method to read data file and count the number of words in the data file. The second assignment included a class file, a driver class, and a data file. Given the driver class and data file, students were asked to create an Employee class.

Table 2. Comparison of the grades assigned by professors and auto-grading tool

No.	Assignment 1			Assignment 2			Assignment 3		
	P1	P2	Auto	P1	P2	Auto	P1	P2	Auto
1	73	72	75	77	75	80	70	74	73
2	80	78	83	73	69	75	85	80	85
3	70	73	75	65	65	70	50	63	70

The third assignment involved five files, one date file, and four class files: Employee, Hourly Worker, Manager, and Commission Worker. Given the data

file and the Employee class, students were required to design three sub-classes from super class Employee. The grades for the assignments are displayed in Table 2.

Overall, the grades awarded by the two professors and auto-grading tool were consistent. However, the grades assigned by the auto-grading were slightly higher than the ones assigned by professors. Especially for assignments 2 and 3, the grades assigned by the auto-grading tool were much higher than those assigned by professors. We met and discussed the assignments and agreed the grades assigned by the auto tool were more reasonable. In this scenario, the professor did not look into the details when a program fail to compile or an exception is raised. With the use of 15 test cases for assignment 2 and 20 cases for assignment 3, the auto-grading tool is more likely to evaluate more traits of the program, which would result in a higher score.

6. Conclusion

It has been shown that the most effective methods of learning in programming courses depend on practical exercises [1]. But preparing, collecting and grading homework manually takes a lot of time. When the number of students increases, the number of homework given reduces. As a result, the effectiveness of the whole course is diminished. One way to solve this problem is to distribute homework through an automatic grading system and provide fast feedback. Automatic grading of programming assignments is an important topic in academic research. It aims at improving the level of feedback given to students and optimizing the professor's time. Its importance is more pronounced as the amount and complexity of assignments increases. Several studies have reported the development of software tools to support this process. Such systems usually consider particular deployment scenarios and specific requirements of specific institutions. However, the quantity and diversity of these tools makes it difficult to get a quick and accurate assessment of their features. This paper proposes a web-based system that automatically grades programming assignments and provides interactive or live comments and feedback to the student. The system accepts a Java application program submitted by a student and returns test results immediately. The test consists of compiler check, result token evaluation, and JUnit test cases. The system has been used in two actual courses at our university. It has been observed that the accuracy of scoring is very high and the response time of the system is satisfactory. The experience in the classroom is very positive and the system did increase the productivity of the class. A significant increase in the quality of student codes and a decrease in compiling and runtime errors have also been observed.

7. References

- [1] Brenda Cheang, Andy Kurnia, Andrew Lim, Wee Chong Oon (2003). On automated grading of programming assignments in an academic institution. *Computers and Education* 41 (2003) 121–131.
- [2] Demir, Ö., Soysal, A., Arslan, A., Yürekli, B., and Yılmazel, Ö., (2010). Automatic Grading System for Programming Homework. *Computer Science Education: Innovation and Technology, CSEIT*.
- [3] P. Ertmer, J. Richardson, B. Belland, D. Camin, P. Connolly, G. Coulthard, K. Lei, and C. Mong, (2007). Using peer feedback to enhance the quality of student online postings: An exploratory study. *Journal of Computer-Mediated Communication*, 12(2):412–433.
- [4] George Stefanek and Nipul Patel, (2015). Moving Toward a Server-Based Virtual Machine Hosting Environment In Support Of University Information Technology Programming Courses. *Issues in Information Systems*. Volume 16, Issue 1, pp. 219-225.
- [5] Larry Hardesty, MIT News Office, (2013). “Automatically grading programming homework-Software that identifies errors in students’ programs - however idiosyncratic their approaches - could be a boon for online learning.” June 3, <http://news.mit.edu/2013/automatically-grading-programming-homework-0603> (Access Date: December 19, 2015).
- [6] Ramnarain-Seetohul, V., J. Abdool Karim, I., Amir, A., (2012). A Case Study of an Online Assignment Submission System at UOM. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering* Vol:6, No:8.
- [7] Ramnarain-Seetohul, V., J. Abdool Karim, I., Amir, A., (2012). A Case Study of an Online Assignment Submission System at UOM. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering* Vol:6, No:8.
- [8] Moodle.org, (2017). Moodle plugins directory: Virtual programming lab. [online] Available at: https://moodle.org/plugins/mod_vpl, (Accessed 29 Jun. 2017).
- [9] J.M. del Alamo, A. Alonso, M. Cortés, (2012). Automated Grading and Feedback Providing Assignments Management Module, *Proc. Int’l Conf. Educ., Research, & Innov.*, Madrid, Spain.
- [10] Thiébaud, D., (2015). Automatic Evaluation Of Computer Programs Using Moodle’s Virtual Programming Lab (VPL) Plug-In. *Consortium for Computing Sciences in Colleges (CCSCNE) 2015*, Boston.
- [11] Codingbat.com, (2017). CodingBat Java. [online] Available at: <http://codingbat.com/java>, (Access date: 29 Jun 2017).
- [12] Betterprogrammer.com, (2012). Java programming test, hands-on skills. [online] Available at: <http://www.betterprogrammer.com/>, (Accessed 29 Jun. 2017).
- [13] Practice-it, (2015). <http://practiceit.cs.washington.edu/> (Access date: 6 April 2015).
- [14] Roberts, G.H.B. and Verbyla, J.L.M., (2003). An Online Programming Assessment Tool. In *Proc. Fifth Australasian Computing Education Conference (ACE2003)*, Adelaide, Australia. *Conferences in Research and Practice in Information Technology*, 20. Greening, T. and Lister, R., Eds., ACS. 69-75.
- [15] Duncan, N., (2007). „Feed-forward“: improving students’ use of tutor comments, *Assessment & Evaluation in Higher Education*. 32 (3), 271 -283.
- [16] Taras, M., (2003). To feedback or not to feedback in student self-assessment. *Assessment and Evaluation in Higher Education*, 28 (5), 549- 565.
- [17] Getzlaf, B., Perry, B., Toffner, G., Lamarche, K., and Edwards, M., (2009). Effective Instructor Feedback: Perceptions of Online Graduate Students. *Journal of Educators Online*, 6(2), n2.
- [18] Jones, C.G., (2004). Test-driven Development Goes to School. *Journal of Computing Sciences in Colleges*, vol. 20, pp. 220-231.