codio

Codio and Wright State University

Case Study on using Codio for CS and Non-CS Majors

About WSU

The WSU faculty, consisting of 27 professors and lectureres, delivers Bachelors (600 students), Masters (300) and PhD (70) Computer Science programs. We also deliver CS courses to approximately 600 non-CS majors.

The two faculty members who have been leading the adoption of Codio at WSU are:

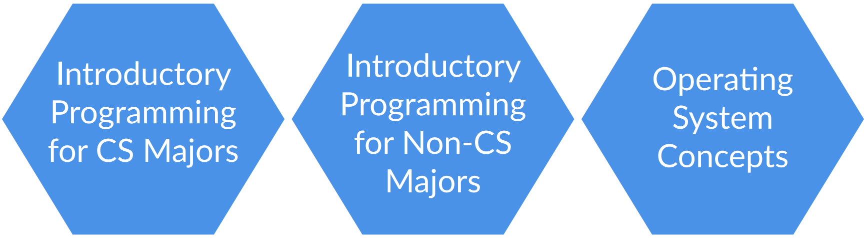
Professor Mateen Rizki, Chair of the Department of Computer Science and Engineering. Prof. Rizki is actively engaged in research in the fields of pattern recognition and bio-informatics. He teaches several programming courses at various levels.

Erik Buck, Instructor, has had a long career in the industry, much of that time active in the aerospace sector. Erik has also authored several popular books, including Cocoa Design Patterns and Learning OpenGL ES for iOS: A Hands-on Guide to Modern 3D Graphics Programming.

> **“The killer combination of a VM with an integrated IDE and class monitoring tools means we no longer require students to keep a work journal to document their project assignment. With Codio, we have instant access to student VMs and code. The time saving for students and faculty is huge.”**

What courses are taught using Codio

Currently we use Codio to deliver the following materials:



Introductory Programming for CS Majors

Introductory Programming for Non-CS Majors

Operating System Concepts

We will add more courses over the coming year.

How Codio is used at Wright State University

Codio is used in a variety of praktikums and labs. Thus far, it is used primarily at the undergraduate level, but there are plans to expand across the department.

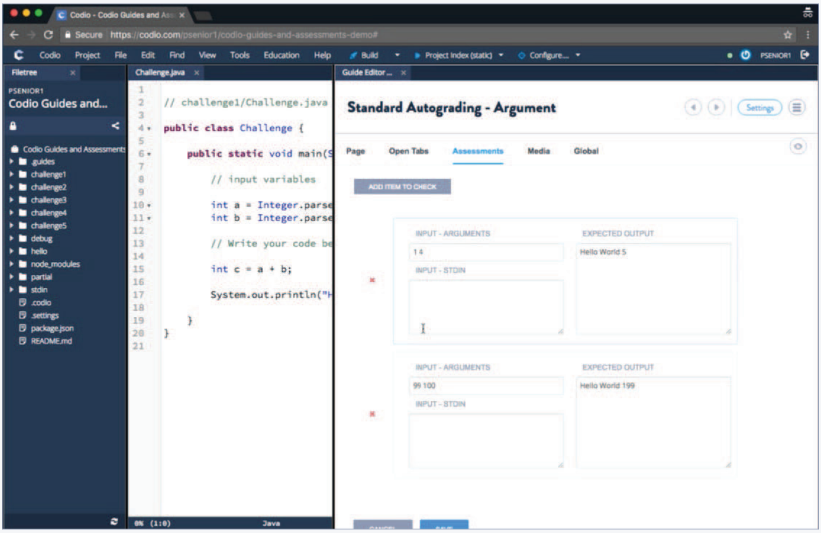
Because everything in Codio is a VM with an integrated IDE, a killer combination, we can configure assignments with any programming languages and components we choose. We can set it up to our own exact specification. Students can also create their coding projects needing nothing more than a browser.

We have made extensive use of the curriculum authoring tool, called Codio Guides, as a way of delivering instructional content. In addition, we can assess and monitor student progress.

Both auto and manual grading are extensively used to allow faculty to always have a clear overview of how students are progressing at all times. Codio's instant access to student code is a key feature that faculty rely on to significantly streamline key student interactions.

There are many "student engagement and management" tools but none of these are subject specific. What sets Codio apart is its focus on CS. The student management and coding specific features give a unique experience when compared with anything else we have considered in the past, both for students and faculty.

Using Codio, we have reached a higher level of "feel" for student progress and gained far more insight than previously. In the past, we would require students keep a work journal and document what they did during project assignments. With the combination of progress monitoring tools and instant access to student VMs and code, we no longer have to do this, saving time and administrative work for both students and faculty. We can see everything they have done and are doing whenever we choose.



```
// challenge1/Challenge.java

public class Challenge {

    public static void main(S

        // input variables

        int a = Integer.parse
        int b = Integer.parse

        // Write your code be

        int c = a + b;

        System.out.println("H

    }
}
```

Standard Autograding - Argument

INPUT - ARGUMENTS: 1 4
INPUT - STDIN:
EXPECTED OUTPUT: Hello World 5

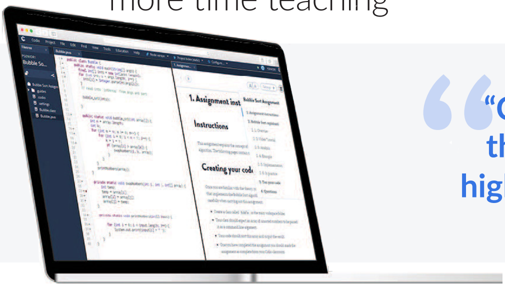
INPUT - ARGUMENTS: 99 100
INPUT - STDIN:
EXPECTED OUTPUT: Hello World 199

Why we selected Codio

There have been many benefits to using Codio at WSU. Some of the benefits were clear to us before we adopted the platform. However, some of the most exciting benefits were rather unexpected and have led to new technology approaches.

The main reasons we originally chose Codio were:

- The ability to configure environments of any complexity that can be accessed as easily at home on students' own PCs as on a lab PC.
- Auto-graded questions and auto-graded code assessments
- Manual grading of project assignments
- Ability to instantly access student code whenever required
- Each assignment is a full-powered VM that can be configured with any programming language, database, component, etc., even providing sudo level privileges
- Knowing that all students would start each assignment with the exact same configuration, no matter how complex
- Being web-based, we didn't have to act as support staff for students' own laptops and this allows us to spend more time teaching



"Codio has delivered on all of these expectations and been highly satisfactory in operation."

How Codio has improved how we teach programming

Having become more experienced using Codio, we have developed methodologies that changed the way we teach:

> **"We now have a collaborative approach whereby we show student assignments alongside the lecturer's own solution on the classroom projector. This lets us demonstrate that there are many valid solutions to the same problem, occasionally superior to the teacher's! We also discuss and diagnose problems collaboratively during class and being able to boot up projects in seconds means we can do it seamlessly."**

- In many classes, the lecturer now uses the classroom projector to show different student assignment projects as well as the lecturer's own solution. This has led to a new and highly collaborative approach to classes where it is shown there are many valid and different solutions, and sometimes superior to the lecturer's own solutions! We also discuss and diagnose problems with student solutions.
- The benefits of a VM only became fully clear a little while later. Now, lecturers and TAs can design projects that allow them to monitor the progress in greater detail, track the command history and coding history to gain a far deeper understanding of the challenges faced by individuals and the class as a whole.

- The strategic use of simple MCQ assessments, and their results clearly displayed in the Teacher Dashboard allows faculty to clearly see whether students are taking due care when reading the teaching materials properly and how fast they are moving through them. This has led to unprecedented insights into class behavior.

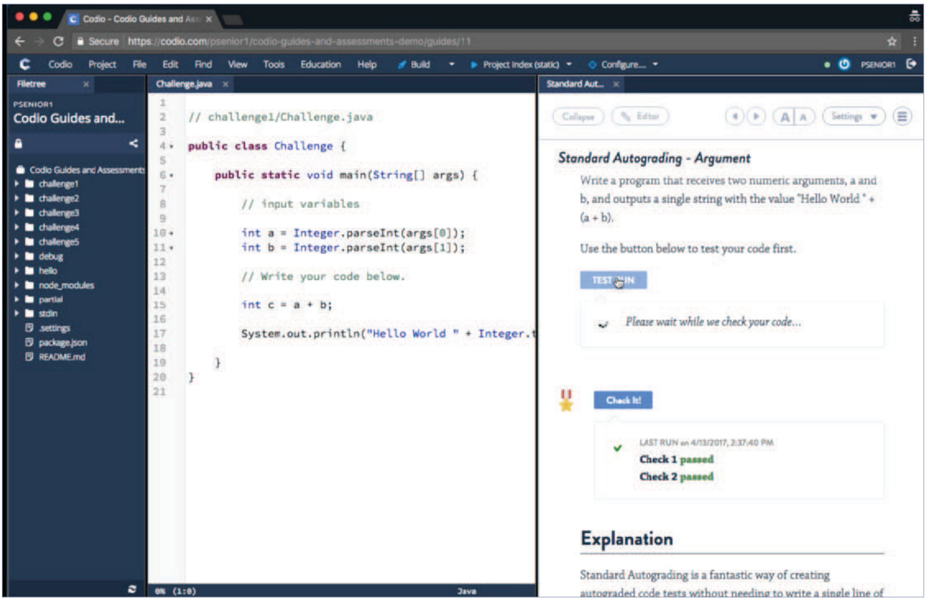
- In the past, students did not get meaningful feedback until their assignments were turned in. Now, faculty can instantly review, compile and run student code when students ask for assistance or they want to review progress manually.

- We now often show different example projects in a class to illustrate or reinforce concepts. Being able to pre-prepare these examples, no matter how different or complex each stack configuration, makes for a more engaging experience. Each example running in its own isolated environment and booting up in a couple of seconds enables us to do this seamlessly.

The importance of Codio's auto and manual grading

We spend a six figure sum each year on grading assignments. We will now convert many of our manually graded assignments into auto-graded coding assessments.

We anticipate that at least 80% of our auto-grade assessments can take advantage of Codio's "Standard Autograding" feature, a feature that was originally suggested by WSU. The advantage of this feature is that it lets code tests be graded without needing to write a single line of testing code.



```java
// challenge1/Challenge.java

public class Challenge {

    public static void main(String[] args) {

        // input variables

        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);

        // Write your code below.

        int c = a + b;

        System.out.println("Hello World " + Integer.t

    }
}
```

Standard Autograding - Argument

Write a program that receives two numeric arguments, a and b, and outputs a single string with the value "Hello World " + (a + b).

Use the button below to test your code first.

Please wait while we check your code...

LAST RUN on 4/13/2017, 2:37:40 PM

Check 1 passed

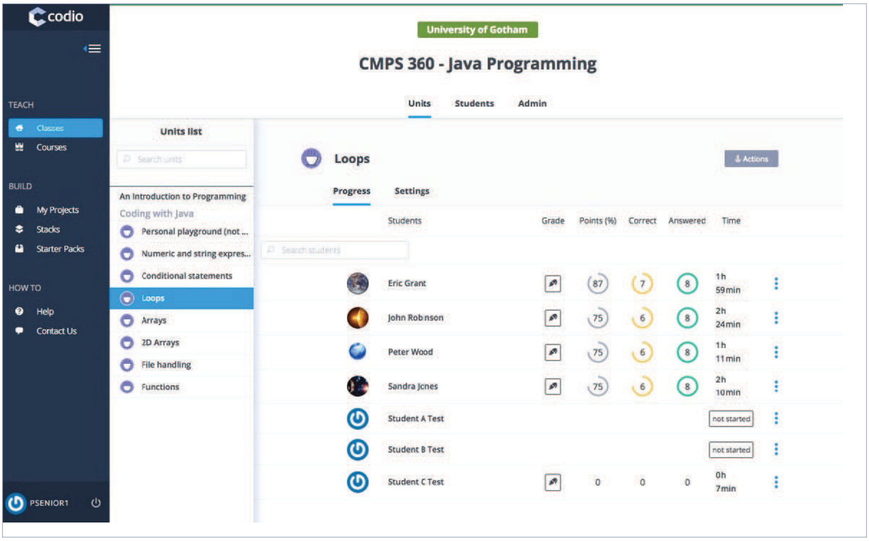
Check 2 passed

Explanation

Standard Autograding is a fantastic way of creating autograded code tests without needing to write a single line of

"We fully expect to be able to make cost savings in the tens of thousands of dollars by using Codio's auto-grading functionality."

We have also developed an approach to student progress monitoring based upon the use of multiple choice and "fill in the blank" questions. Within the Guides tutorial content we embed a series of simple questions that allows faculty to monitor how thoroughly students are reading the subject matter. This ability to monitor student engagement is especially useful in the context of the flipped classroom where students are expected to prepare for a class.



Students	Grade	Points (%)	Correct	Answered	Time
Eric Grant		87	7	8	1h 59min
John Rob nson		75	6	8	2h 24min
Peter Wood		75	6	8	1h 11min
Sandra Jcnes		75	6	8	2h 10min
Student A Test					not started
Student B Test					not started
Student C Test		0	0	0	0h 7min

How do students find Codio?

We have assessed student satisfaction with Codio over 2 semesters. There was some resistance to Codio in the first semester as students got used to it. However, we have since encountered little or no resistance.

> **"We do not mandate development tools on our students, but we are now finding that almost all students are choosing to do their project assignments in Codio due its reliability and ease of use."**

LMS Integration

We use the Desire2Learn LMS platform for managing students and classes. Codio integrates very nicely with D2L and the mapping between D2L assignments and Codio Units is quick and easy to configure. For access to more detailed assessment information, time spent statistics and direct access to student code, we use Codio's own LMS features, which nicely coexist with D2L.

CS Lab Investment and BYOD

One of the things we expect to achieve in the longer term with Codio is a gradual reduction in our CS lab hardware investment. At WSU, we tend to renew lab hardware every three to four years. There are also associated support and maintenance costs.

Although there is no official Bring Your Own Device policy and WSU operates an 'Open Access' policy, the student BYOD rate is at least 50% and increasing all the time. As this rate increases, WSU expects to make cost savings as we reduce hardware investment over time. Students generally prefer to use their own laptops and because Codio is browser-based, we are able to indulge this preference. The only expectation is for specific controlled assessments where we provide a locked down environment.

Codio Support

Codio offers exceptionally good customer support. They provided us with dedicated trouble-shooting when we were experiencing networking issues.

Codio has responded very well to various feature requests. For example, they developed the Standard Autograding assessment type for grading student code. The original code auto-grading feature (which Codio has renamed Advanced Autograding) required us to write our own grading scripts. This placed an additional burden on faculty to carefully debug their testing scripts whereas Standard Autograding is dialog driven.

Faculty and students have access to a customer support feature directly within the platform, so any issues we encounter can be instantly communicated and tracked. The response to any such issues has been exemplary.

codio