



Supporting Team Submissions and Peer Grading within Submittly

Matthew Peveler, Samuel Breese, Evan Maicus,
Andrew Aikens, Timothy Cyrus, Elizabeth Dinella,
James Anderson, Josh Barthelmess, Marisa Lee,
Leon Montealegre, Jessica Wang,
Buster Holzbauer, Barbara Cutler, and Ana Milanova

Rensselaer Polytechnic Institute



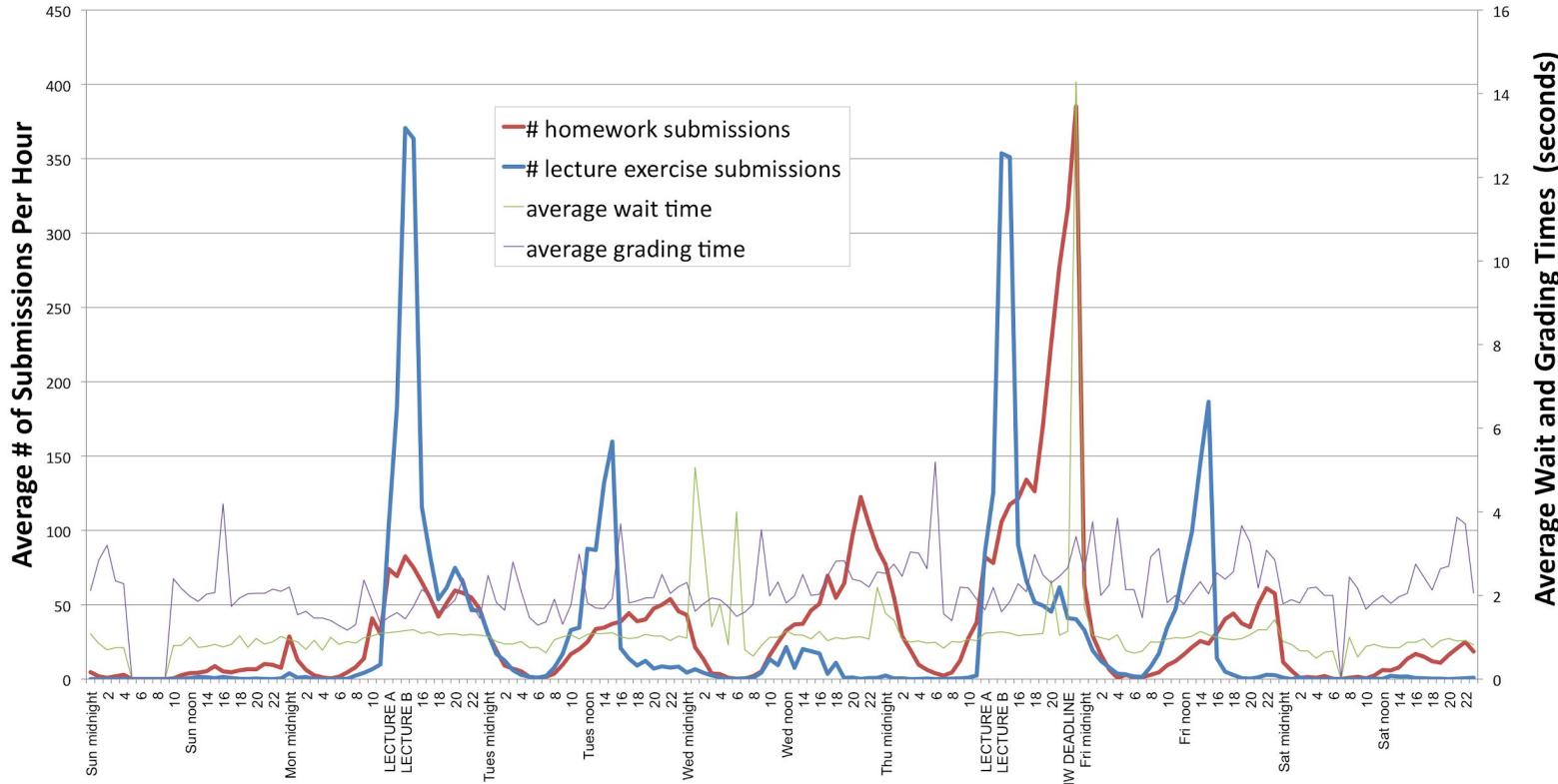
What is Submitty?

- Students upload code (and resubmit) for auto-grading
- TAs review and add additional grading/feedback
- Configurable number of late days per gradeable
- Open-source, free to use
- Installed on your own hardware or VPS
 - Instructors have ssh access to files & logs for debugging
- Support for any language / tool installed on your server
 - We use Python, C, C++, Java, Scheme, Prolog, and SPIM
 - JUnit, Emma code coverage, Dr Memory, static analysis, ...
- Supports dozen of courses with thousands of users at RPI
 - 500-700+ students in Computer Science I and Data Structures

Server Performance



Weekly Average Homework and Lecture Exercise Submissions





More Information

- All material from this demo available at
<https://submitty.org/tutorial>
<https://github.com/Submitty>
- New users are welcome! Ask us questions:

submitty-admin@googlegroups.com

- New developers are welcome:
Rensselaer Center for Open Source Software (RCOS)
Sponsored by RedHat Software
Google Summer of Code 2018
To access our Slack server:

<http://submitty.org/developer/>





Today's Demo

- Support for team assignments *new for Fall 17!*
- Student, grader, and instructor views
- Manual grading interface with common marks *new for Fall 17!*
- Support for Peer Grading *planned for Summer 18!*
- Submitty Discussion Forum *new for Spring 18!*
- Autograding with External Tools
- Static Analysis for Autograding *poster at 2pm!*
- Flexible late day policy, plagiarism, student stress *poster at 2pm!*
- Sandbox/Docker *poster at 2pm!*
- Future Goals



Team Assignments

Instructors specify maximum size and “team lock date”

What is the type of the gradeable?: **(Required)**

- Electronic File Checkpoints Numeric/Text

Is this a team assignment? *Team assignments are new as of Fall 2017. Email questions/bugs/feedback to: submitty@cs.rpi.edu.*

- Yes No

What is the maximum team size?

What is the **Team Lock Date**? (Instructors can still manually manage teams):

What is the **Submission Open Date**? (submission available to students): must be >= TA Beta Testing Date

What is the **Due Date**? must be >= Submission Open Date

How many late days may students use on this assignment?

Are students uploading files or submitting to a Version Control System (VCS) repository?

- Upload File(s) Version Control System (VCS) Repository

Full path to the directory containing the autograding config.json file:

See samples here: Submitty GitHub sample assignment configurations

/usr/local/submitty/more_autograding_examples/upload_only/config (an assignment without autograding)

/var/local/submitty/private_course_repositories/MY_COURSE_NAME/MY_HOMEWORK_NAME/ (for a custom autograded homework)

/var/local/submitty/courses/s18/sample/config_upload/# (for an web uploaded configuration)

Should students be able to view submissions?

Create a new team and invite teammates



Hello Dustin ([Logout](#))
[Submitty](#) > [sample](#)

Submitty

Rensselaer Center for Open Source

Manage Team For: Open Team Homework

You are not on a team.

You have not received any invitations.

[Create New Team](#)

Hello Dustin ([Logout](#)) Created a new team
[Submitty](#) > [sample](#)

Submitty

Rensselaer Center for Open Source

Manage Team For: Open Team Homework

Your Team:

Dustin Borer (borerd) - borerd@example.com

Invite new teammates by their user ID:

User ID [Invite](#)

[Leave Team](#)

Or accept an invitation from a classmate

Hello Clementina ([Logout](#))
[Submitty](#) > [sample](#)

Submitty

Rensselaer Center for Open Source

Manage Team For: Open Team Homework

You are not on a team.

Invitations:

bored: [Accept](#)

[Create New Team](#)

Hello Clementina ([Logout](#)) Accepted invitation from
borerd
[Submitty](#) > [sample](#)

Submitty

Rensselaer Center for Open Source

Manage Team For: Open Team Homework

Your Team:

Dustin Borer (borerd) - borerd@example.com
Clementina Durgan (durgac) - durgac@example.com

Invite new teammates by their user ID:

User ID [Invite](#)

[Leave Team](#)



Instructor View of Teams

Instructors monitor team formation, and can create/edit teams

Hello Quinn (Logout)

Submitty > sample > Open Team Homework Grading > Student Index

Create New Team: bodew

Team Member IDs:

Registration Section:

Grade Details for Open Team Homework

Section	Edit Teams	Team Id	Students	TA Grading	Total	Active Version	Viewed Grade
1	1	00001_borerd	Dustin Borer, Clementina Durgan	5 / 5	<input type="button" value="Grade"/>	2	
2	1	00000_borerd	Joe Student	0 / 5	1 2	<input type="button" value="Grading Incomplete"/>	6
3	1		heanec		No Team		
4	1		hoegea		No Team		
5	1		jacobs		No Team		
6	1		kerluh		No Team		
7	1		mckenh		No Team		
8	1		smithj		No Team		
9	1		spinko		No Team		
10	1		warde		No Team		
11	1		willia		No Team		

Students Enrolled in Registration Section 2
Graders: manne

Section	Edit Teams	Team Id	Students	TA Grading	Total	Active Version	Viewed Grade
1	2	bechta		No Team			

Submitty

Rensselaer Center for Open Source

Team Submissions

- Anyone on a team can submit or resubmit
 - All team members share one sequence of versions
 - TAs add manual grades for teams just like single students

Team: borerd, durgac

Select Submission Version: Version #1 Score: 4 / 10 GRADE THIS VERSION ▾ Do Not Grade This Assignment

Note: This version of your assignment will be graded by the instructor/TAs and the score recorded in the gradebook.

Submitted Files

frame_buggy.cpp (0.25kb)	submission timestamp: 02/22/2018 01:28:08 AM days late: 0 (before extensions) grading time: 9 seconds queue wait time: 0 seconds
--------------------------	---

Results

	Total	
4 / 10	Total	
2 / 2	Test 1 Compilation	Detail
2 / 2	Test 2 Frame Size 1	Detail
0 / 2	Test 3 Frame Size 5	Detail
0 / 2	Test 4 Frame Size 10	Detail
Hidden	Test 5 Frame Size 13	
0 / 2	Test 6 Error Checking: Frame Size 0	Detail
Hidden	Test 7 Error Checking: Frame Size Negative	
Test 8 Error Checking: No Arguments	<i>Extra Credit</i>	Detail

3/3**Test 5 Imaginary Roots** 1 3 10 1 -3 10 1 -3 -10[Details](#)**2/2****Test 6 Double Root** 1 6 9[Details](#)**Student STDOUT.txt**

```
1 Enter 3 integer coefficients to a quadratic function: a*x*x + b*x + c = 0
2 The roots are: -3 and -3
3
```

Expected STDOUT.txt

```
1 Enter 3 integer coefficients to a quadratic function: a*x*x + b*x + c = 0
2 The roots are: -3 and -3
3
```

2/2**Test 7 Zero Root** 1 4 0[Details](#)**1/3****Test 8 a != 1** 2 7 3[Details](#)**Student STDOUT.txt**

```
1 Enter 3 integer coefficients to a quadratic function: a*x*x + b*x + c = 0
2 The roots are: -2 and -12
3
```

Expected STDOUT.txt

```
1 Enter 3 integer coefficients to a quadratic function: a*x*x + b*x + c = 0
2 The roots are: -0.5 and -3
3
```

Standard Error (STDERR)

WARNING: This file should be empty

```
1 ERROR: -2 is not a root of this formula.
2 ERROR: Unable to verify one or both roots.
3
```



Git Integration

- Can be configured with an internal git server
 - Installed to /var/local/submitty/vcs
 - Students get a web path to access
ex: <https://submitty-vcs.cs.rpi.edu/git/>

Version Control System (VCS) Base URL

Base URL if students are submitting via VCS repository.

external ex. https://github.com/test-course

internal ex. ssh+svn://192.168.56.101/test-course

/var/local/submitty/vcs/s18/sample/

Version Control System (VCS) Type

Choose the type of VCS if students are submitting via VCS repository.

Git

- Can use external sources as well, such as Github



Instructor Configuration

What is the **Submission Open Date**? (submission available to students): must be >= TA Beta Testing Date

What is the **Due Date**? must be >= Submission Open Date

How many late days may students use on this assignment?

Are students uploading files or submitting to a Version Control System (VCS) repository?

- Upload File(s) Version Control System (VCS) Repository

Path for the Version Control System (VCS) repository:

VCS base URL: <http://192.168.56.102/git/s18/sample>

The VCS base URL is configured in Course Settings. If there is a base URL, you can define the rest of the path below. If there is no base URL because the entire path changes for each assignment, you can input the full path below. If the entire URL is decided by the student, you can leave this input blank.

You are allowed to use the following string replacement variables in format \${...}

- gradeable_id
- user_id OR team_id OR repo_id (only use one)

ex. /\${gradeable_id}/\${user_id} OR [https://github.com/test-course/\\${gradeable_id}/\\${repo_id}](https://github.com/test-course/${gradeable_id}/${repo_id})

VCS URL: [http://192.168.56.102/git/s18/sample/\\${gradeable_id}/\\${user_id}](http://192.168.56.102/git/s18/sample/${gradeable_id}/${user_id})

Full path to the directory containing the autograding config.json file:

See samples here: Submitty GitHub sample assignment configurations

`/usr/local/submitty/more_autograding_examples/upload_only/config` (an assignment without autograding)

`/var/local/submitty/private_course_repositories/MY_COURSE_NAME/MY_HOMEWORK_NAME/` (for a custom autograded homework)

`/var/local/submitty/courses/s18/sample/config_upload/#` (for an web uploaded configuration)

Should students be able to view submissions?

- Yes No (Select 'No' during grading of a bulk upload pdf quiz/exam.)



Git Integration - Submitting

Hello Joe ([Logout](#))

[Submitty](#) > [sample](#) > [Final Project](#)



New submission for: Final Project

Due: 03/02/2018 @ 23:59

No automatic grading for this assignment

To access your Repository:

Note: There may be a delay before your repository is prepared, please refer to assignment instructions.

```
git clone http://192.168.56.102/git/s18/sample/final_project/student SPECIFY_TARGET_DIRECTORY
```

[Grade My Repository](#)

No submissions for this assignment.



- Components represent top level rubric items
- “Marks” add and/or subtract points (“count up” or “count down”)
- Instructor can “publish” certain marks (student sees the grading criteria, even if the mark was not selected for them)

Manual/TA/Peer Grading Rubric		Points
Read Me		Points: 2
Message to TA/Grader (seen only by TAs/Graders)		Extra Credit: <input type="radio"/> Yes <input checked="" type="radio"/> No
Message to Student (seen by both students and graders)		Penalty: <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Grade by count up <input checked="" type="radio"/> Grade by count down
• 0	Full Credit	
• -1	Minor errors in Read Me	<input type="checkbox"/> Publish
• -2	Major errors in Read Me or Read Me missing	<input type="checkbox"/> Publish
Add Common Deduction/Addition		
Coding Style		Points: 5
Message to TA/Grader (seen only by TAs/Graders)		Extra Credit: <input type="radio"/> Yes <input checked="" type="radio"/> No
Message to Student (seen by both students and graders)		Penalty: <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Grade by count up <input checked="" type="radio"/> Grade by count down
• 0	Full Credit	
• -5	Code is unreadable	<input type="checkbox"/> Publish
• -3	Code is very difficult to understand	<input type="checkbox"/> Publish



- Components represent top level rubric items
- “Marks” add and/or subtract points (“count up” or “count down”)
- Instructor can “publish” certain marks (student sees the grading criteria, even if the mark was not selected for them)

Manual/TA/Peer Grading Rubric		Points
Read Me		Points: 2 Extra Credit: <input type="radio"/> Yes <input checked="" type="radio"/> No Penalty: <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Grade by count up <input checked="" type="radio"/> Grade by count down
Message to TA/Grader (seen only by TAs/Graders)		
Message to Student (seen by both students and graders)		
• 0	Full Credit	
• -1	Minor errors in Read Me	<input type="checkbox"/> Publish
• -2	Major errors in Read Me or Read Me missing	<input type="checkbox"/> Publish
Add Common Deduction/Addition		
Coding Style		Points: 5 Extra Credit: <input type="radio"/> Yes <input checked="" type="radio"/> No Penalty: <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Grade by count up <input checked="" type="radio"/> Grade by count down
Message to TA/Grader (seen only by TAs/Graders)		
Message to Student (seen by both students and graders)		
• 0	Full Credit	
• -5	Code is unreadable	<input type="checkbox"/> Publish
• -3	Code is very difficult to understand	<input type="checkbox"/> Publish



- Components represent top level rubric items
- “Marks” add and/or subtract points (“count up” or “count down”)
- Instructor can “publish” certain marks (student sees the grading criteria, even if the mark was not selected for them)

Manual/TA/Peer Grading Rubric		Points
Read Me		Points: 2
Message to TA/Grader (seen only by TAs/Graders)		Extra Credit: <input type="radio"/> Yes <input checked="" type="radio"/> No
Message to Student (seen by both students and graders)		Penalty: <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Grade by count up <input checked="" type="radio"/> Grade by count down
• 0	Full Credit	
• -1	Minor errors in Read Me	<input type="checkbox"/> Publish ×
• -2	Major errors in Read Me or Read Me missing	<input type="checkbox"/> Publish ×
Add Common Deduction/Addition		
Coding Style		Points: 5
Message to TA/Grader (seen only by TAs/Graders)		Extra Credit: <input type="radio"/> Yes <input checked="" type="radio"/> No
Message to Student (seen by both students and graders)		Penalty: <input type="radio"/> Yes <input checked="" type="radio"/> No <input type="radio"/> Grade by count up <input checked="" type="radio"/> Grade by count down
• 0	Full Credit	
• -5	Code is unreadable	<input type="checkbox"/> Publish ×
• -3	Code is very difficult to understand	<input type="checkbox"/> Publish ×

Auto-Grading Testcases

Expand All | Close All

Score	Total	
12 / 10	Total	
2 / 2	Test 1 Compilation	
2 / 2	Test 2 Frame Size 1	Details
2 / 2	Test 3 Frame Size 5	Details
2 / 2	Test 4 Frame Size 10	Details
0 / 2	HIDDEN: Test 5 Frame Size 13	Details
2 / 2	Test 6 Error Checking: Frame Size 0	Details

Grading Rubric

Auto scroll / Auto open Overwrite Grader

Read Me		Graded by instructor
2 / 2	(* lorem ipsum	
Coding Style		Graded by instructor
4 / 5	(* -1.0 Code is difficult to understand (*) lorem ipsum	
Documentation		Graded by instructor
4.5 / 5	(* Full Credit (*) -3.0 Very little documentation or documentation makes no sense (*) 2.5 lorem ipsum	
Extra Credit		Graded by instructor

Submissions and Results Browser

Open/Close Submissions

Open/Close Results | Download Zip File

- submissions
 - .submit.timestamp
 - frame_hardcoded.cpp

```
#include <iostream>
#include <cstdlib>
#include <string>
int main(int argc, char* argv[]) {
  if (argc == 1) {
    std::cerr << "ERROR! Wrong number of arguments" <<
    std::endl;
    exit(1);
  }
  int value = atoi(argv[1]);
  if (value == 0) {
    std::cerr << "ERROR! Argument should be >=1" << std::endl;
    exit(1);
  }
}
```

Student Information

Toy Mraz (mratz)
 Submission Number: 1 / 1
 Submitted: 12/31/1971 23:59:59

Browse Student Submissions: Version #1 Score: 12 / 10 GRADE THIS VERSION

Overall Late Day Usage

[Cancel Student Submission](#)

	Allowed per term	Allowed per assignment	Submitted days after deadline	Extensions	Status	Late Days Charged	Total Late Days Used	Remaining Days
Closed Homework	3	2	0	0	Good	0	0	3
Grading Homework	3	2	0	0	Good	0	0	3
TA Grade Only Homework	3	2	0	0	Good	0	0	3

Grading With Common Marks



Submitty
Rensselaer Center for Open Source

Customize
panel layout

Add/edit marks
during grading

Improves
consistency
between
graders

The screenshot shows the Submitty web application interface. At the top, there is a header bar with navigation icons (back, forward, search, etc.) and a progress bar showing 42.8%. Below the header, the URL path is displayed: Submitty > sample > Grading Homework Grading > Student Index.

The main area is divided into two main sections:

- Submissions and Results Browser:** This section shows the student's submission for a homework assignment. It includes a file tree with 'submissions' and 'results' folders, and a code editor window displaying Python code for calculating percentage change. The code is as follows:

```
1 def percent_change(old,new):
2     return int(100*(float(new)-old)/old)
3
4 def print_change(old1, new1, old2, new2):
5     p1 = percent_change(old1,new1)
6     p2 = percent_change(old2,new2)
7     print p1, "vs", p2
8
9 print "#icebucketchallenge vs #alsicebucketchallenge, percentage
change"
10 print_change(200,500,100,300)
11 print_change(500,2000,300,1500)
12 print_change(2000,12000,1500,13000)
13 print_change(12000,24000,13000,25000)
14 print_change(24000,65000,25000,105000)
15 print_change(65000,70000,105000,85000)
16
17 i = 0;
18 while i != 0:
19     #print "hello", i
20     i+=1
21
```

- Grading Rubric:** This section allows the grader to assign points and provide feedback for different aspects of the assignment. It includes sections for 'Read Me', 'Coding Style', and 'Documentation'. For 'Documentation', there are five options ranging from 'Full Credit' (5/5) to 'Very little documentation or documentation makes no sense' (-3). A 'Custom' option is also available. There is also an 'Extra Credit' section with a 'Click me to grade!' button.

*We support bulk pdf upload (test/quiz),
name matching and manual grading*



The screenshot shows the Submitty web interface for managing uploaded files. On the left, there is a list of uploaded items:

- Item 1: 02-22-2018 01:22:08-0500. Content: Dustin Borer <borerd> HW 2, Writup. Description: For this project, I implemented a calculator in python. To do this, I had to use functions and variables. Installing python on my computer was easy because it was already installed. The
- Item 2: 02-22-2018 01:22:08-0500. Content: Otho Spinka <spinko> HOMEWORK 2. Description: Creating this calculator was an interesting and enriching experience, which I really feel helped

On the right, there are two separate boxes, each containing a file name and a list of names:

- Box 1: File name: du. Names: Dustin Borer <borerd>, Clementina Durgan <durgac>. Buttons: Submit (green), Delete (red).
- Box 2: File name: batch_upload_1.pdf. Names: (empty). Buttons: Submit (green), Delete (red).



Peer Review/Grading

- Instructor configures some components for peer review
- Instructor specifies # of peer review assignments
 - Assigned randomly
 - Or assigned by .csv upload
- Peers use same interface as TAs to review/grade
 - Access restricted to relevant files and dates
 - Student identities are randomized
- Average scores given by multiple peer graders
 - Compute consistency of grades — “grade the grader”



Submitty Forum

- Purpose: For instructor/TA announcements, student question/answers, offline discussion of assigned readings, etc.
- Stay within Submitty environment (same login, moderated by TAs/instructor)
- Threads/posts/replies/hierarchy
- Multiple Image attachments/links/prose vs. code segments
- Current/Future work: search, notifications (incl. email), direct message/regrade requests



Create a New Thread



Create Thread

Title: Homework 1 Now Posted

Link ↗

Code </>

Homework 1 has been posted on the course website. You can view it [url=http://example.com]here.[/url]

Hint: In this homework you will have to use a switch statement. I have included an example of one below.

[code]

```
switch ($_REQUEST['page']) {  
    case 'create_thread':  
        $this->showCreateThread();  
        break;  
    case 'make_announcement':  
        $this->alterAnnouncement(1);  
        break;  
    case 'view_thread':  
    default:  
        $this->showThreads();  
        break;
```

Announcements, Code Segments, and Links



Hello Alyssa P ([Logout](#))

[Submitty](#) > [sample](#) > [Discussion Forum](#)

Quiz Reminder
In addition to the homework that was just released. Please don't forget that we have a...
02/22/2018 11:44 PM

Hw 1 Released
Homework 1 has been released on the course website. You can view it...
02/22/2018 11:27 PM

getRGB() function...
I think that my switch statement is correct and it seems as if it is getting into the right switch...
02/22/2018 11:42 PM

Hw1 switch statement not working...
I created what I believe is the correct switch statement but it seems to not be working. Can...
02/22/2018 11:33 PM

★ Hw 1 Released

Homework 1 has been released on the course website. You can view it [here](#).

Reminder: make sure to use a switch statement in your code. I have provided an example below:

```
1  public function run() {
2      switch ($REQUEST['page']) {
3          case 'create_thread':
4              $this->showCreateThread();
5              break;
6          case 'publish_thread':
7              $this->publishThread();
8              break;
9          case 'make_announcement':
10             $this->alterAnnouncement(1);
11             break;
12         case 'publish_post':
13             $this->publishPost();
14             break;
15         case 'delete_post':
16             $this->deletePost();
17             break;
18         case 'remove_announcement':
19             $this->alterAnnouncement(0);
20             break;
21         case 'view_thread':
22             $this->showThreads();
23             break;
24     }
25 }
```

Quinn I 02/22/2018 11:27 PM

[Link](#)

[Code](#)

Enter your reply here...

★ Quiz Reminder

In addition to the homework that was just released. Please don't forget that we have a quiz coming up next week. There will be a review session in the Library on Monday at 5:30pm with the quiz being on Thursday.

Jill T 02/22/2018 11:44 PM

Thank you for the reminder!!!

Anonymous 02/22/2018 11:47 PM

What sections are on the quiz again?

Alyssa P H 02/22/2018 11:56 PM

[Link ↗](#) [Code </>](#)

Enter your reply here...

[Upload Attachment](#)

Anonymous (to class)? [Reply](#)

★ Quiz Reminder

In addition to the homework that was just released. Please don't forget that we have a quiz coming up next week. There will be a review session in the Library on Monday at 5:30pm with the quiz being on Thursday.

 Jill T 02/22/2018 11:44 PM

Thank you for the reminder!!!

 Ben Bitdiddle (bitdiddle) 02/22/2018 11:47 PM

What sections are on the quiz again?

 Alyssa P H 02/22/2018 11:56 PM

[Link ↗](#) [Code </>](#)

Enter your reply here...

Anonymous (to class)? [Reply](#)



Student View

TA/Instructor View

Attaching images to posts



Hello Quinn ([Logout](#))
[Submitty](#) > [sample](#) > [Discussion Forum](#)

+ Quiz Reminder
In addition to the homework that was just released. Please don't forget that we have a...
02/22/2018 11:44 PM

Hw 1 Released
Homework 1 has been released on the course website. You can view it...
02/22/2018 11:27 PM

getRGB() function...
I think that my switch statement is correct and it seems as if it is getting into the right switch...
02/22/2018 11:42 PM

Hw1 switch statement not working...
I created what I believe is the correct switch statement but it seems to not be working. Can...
02/22/2018 11:33 PM

★ Hw1 switch statement not working
I created what I believe is the correct switch statement but it seems to not be working. Can someone take a look? I believe that it is in my switch condition.

```
1 switch(color) {...}
```

Anonymous 02/22/2018 11:33 PM

It seems like you are getting an error similar to what I have attached. Make sure to use, assuming that it has been previously instantiated:

```
1 $color
```

Quinn 02/23/2018 12:27 AM

[Link ↗](#) [Code </>](#)

Enter your reply here...

[Upload Attachment](#) Anonymous (to class)? [Reply](#)

Edit Posts



Hello Quinn ([Logout](#))

[Submitty](#) > sample > Discussion Forum

Homework 1 Now Posted
Homework 1 has been posted on the course website. You can access it...
02/23/2018 1:29 AM

Quiz Reminder
In addition to the homework, please don't forget that we have a quiz coming up next.
02/23/2018 11:44 PM

getRGB() function...
I think that my switch statement is correct and it seems as if it is getting into the right switch...
02/22/2018 11:42 PM

Hw1 switch statement not working...
I created what I believe is the correct switch statement but it seems to not be working. Can...
02/22/2018 11:33 PM

Editing a post by: ta on 02/22/2018 11:44 PM

In addition to the homework, please don't forget that we have a quiz coming up next week. There will be a review session in the Library on Monday at 5:30pm with the quiz being on Thursday.

Q In a bein
That

[Cancel](#) [Submit](#)

What sections are on the quiz again?

[Link](#) [Code](#)

Enter your reply here...

[Upload Attachment](#)

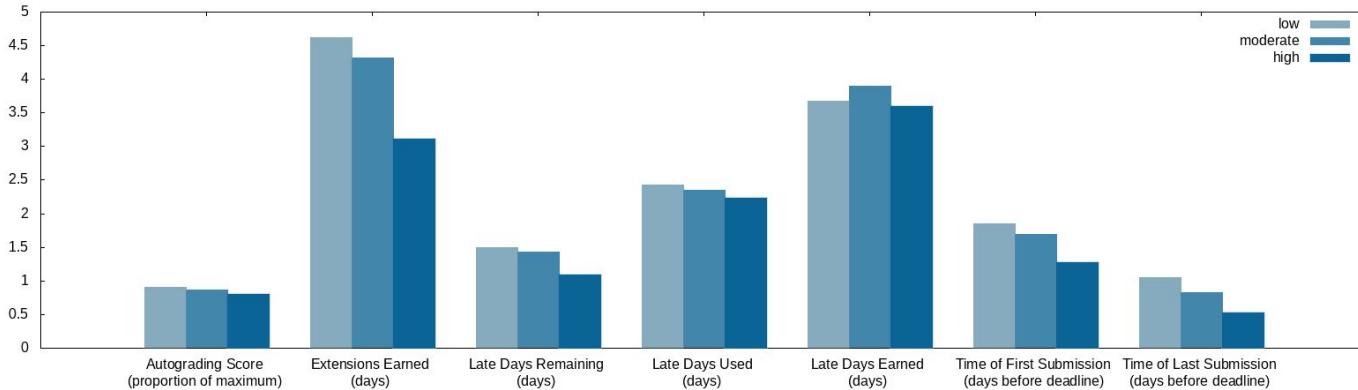
Anonymous (to class)? [Reply](#)

Jill T 02/22/2018 11:44 PM [x](#)

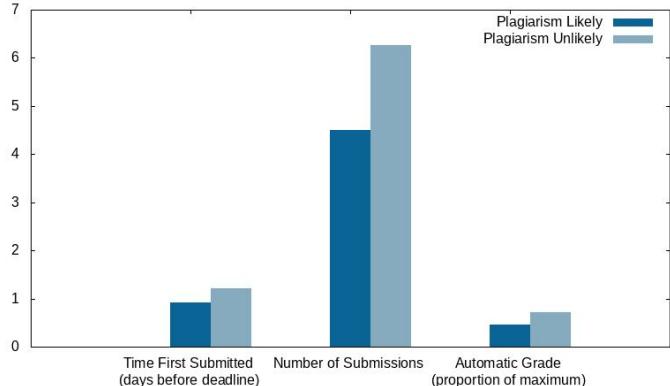
Anonymous 02/22/2018 11:47 PM [x](#)

Alyssa P H 02/22/2018 11:56 PM [x](#)

Flexible Late Day Policy -- Reduces Student Stress



Integrated Plagiarism Detection



Poster
“Correlation of a Flexible Late Day Policy with Student Stress and Programming Assignment Plagiarism”
Today 3-5pm

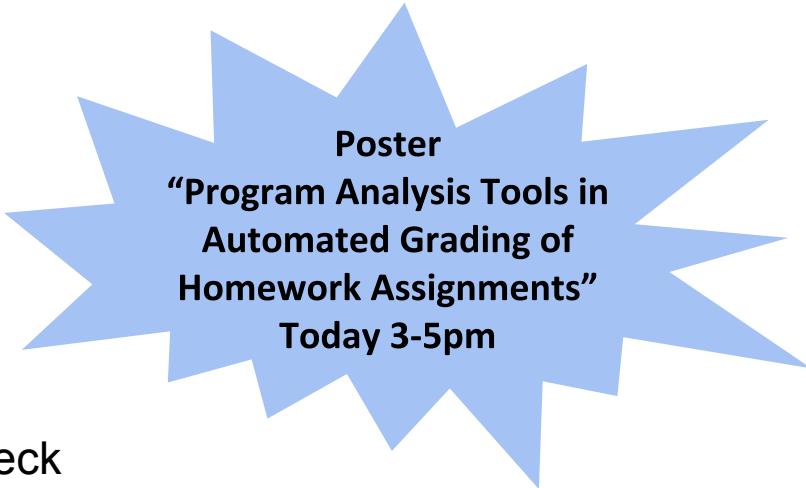


Autograding w/ External Tools

- JUnit, Python unittest
- Memory Debugging (Dr. Memory / Valgrind)
- Code Coverage (EclEmma, JaCoCo)
- Verification-Aware Programming Language (Dafny)
- MPI / OpenMP
- OpenGL/GLFW
- TkInter
- Matlab
- Anything you can install & run on GNU/Linux!

Static Analysis

- In use:
 - Count print / assignments / multiplication
 - Forbid use of goto / auto
 - Verify use of for vs. while
 - Verify use of dictionary
- Current work
 - Loop depth -- naive complexity analysis
 - Function calls itself -- naive recursion check
 - Forbid STL Vector erase
 - Confirm all exceptions are caught
 - Check that all member variables are private
- Future
 - Reverse engineer UML diagram -- design pattern check





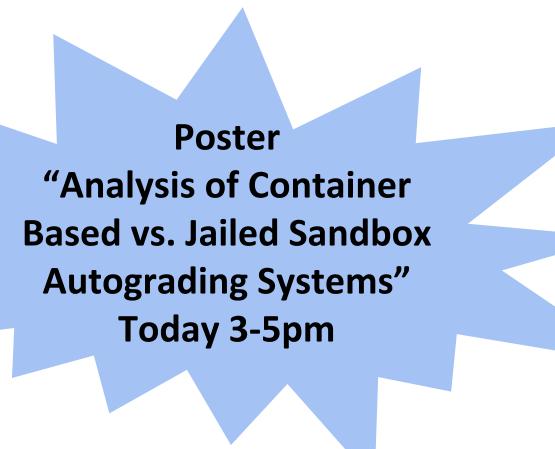
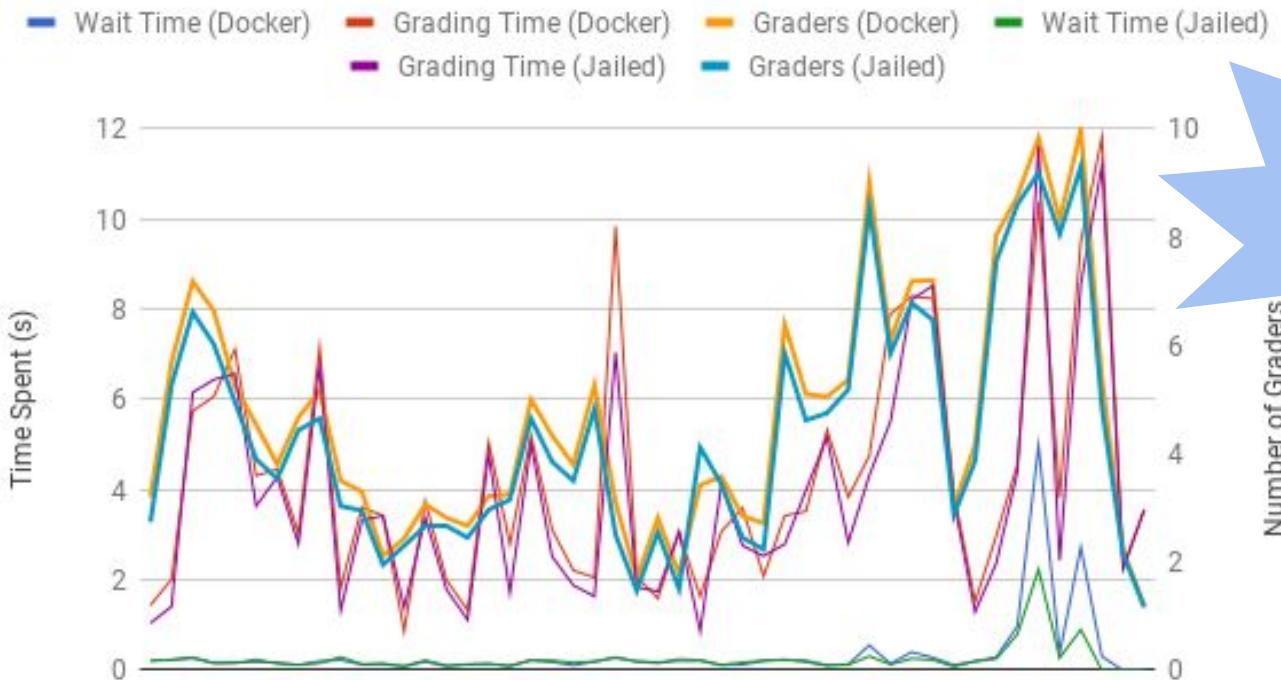
Security via Jailed Sandbox

- Database access done through the PHP Data Objects (PDO) library which protects against malicious and malformed inputs
- Instructor configures appropriate resource limits (GNU Linux `rlimit`) to sandbox testing of electronically-submitted student code and prevent issues like infinite loops, runaway output, and excessive use of other system resources
- Before running the student code, we switch from a privileged system user to an untrusted user using GNU Linux `setresuid`
- Careful design of file and directory permissions and database access maintains confidentiality of student work and grades
- Uses secure computing mode (GNU Linux `seccomp`) to prevent use of sockets, fork, and other unnecessary system calls by student code

*Thanks to RPSEC (our undergraduate Computer Security Club)
for helping find & patch vulnerabilities*

Docker for Autograde Isolation

Customize docker images per course, per assignment





Future Work

- Make it more mobile friendly
- Expand usage of Submitt beyond RPI
- Support SQL autograding
- PDF/Code annotation for TA grading
- Improved Docker integration/support
- Remote graders for advanced topic classes
 - Graphics
 - Parallel Computing
 - Distributed Systems and Algorithms



Thanks!

- All material from this demo available at
<https://submitty.org/tutorial>
<https://github.com/Submitty>
- New users are welcome! Ask us questions:

submitty-admin@googlegroups.com

- New developers are welcome:

Rensselaer Center for Open Source Software (RCOS)

Sponsored by RedHat Software

Google Summer of Code 2018

To access our Slack server:

<http://submitty.org/developer/>





examples/01_simple_python

```
{  
  "testcases" : [  
    {  
      // Student-visible testcase name.  
      "title" : "Python - Simple Grading",  
  
      // Commands to run (in order). These are not shell commands, although  
      // they support some common shell wildcards. This can either be a  
      // list or a single string.  
      "command" : [ "python *.py" ],  
  
      // Point value of this testcase.  
      "points" : 10,  
  
      "validation" : [  
        {  
          // Grade by "diffing" the student output with an  
          // instructor-provided file.  
          "method" : "diff",  
          // The student's output.  
          "actual_file" : "STDOUT.txt",  
          // The title seen by students.  
          "description" : "Program Output",  
          // The instructor-provided file (the correct answer).  
          "expected_file" : "output.txt"  
        }  
      ]  
    }  
  ]  
}
```



examples/02_simple_cpp

```
{  
    // For compiled languages, typically two testcases are used to allow points  
    // to be assigned independently for compilation and execution.  
    "testcases" : [  
        {  
            // Indicate that this is a compilation step.  
            "type" : "Compilation",  
            "title" : "C++ - Compilation",  
            "command" : "clang++ -Wall -o a.out -- *.cpp",  
            // Name of the result of compilation.  
            "executable_name" : "a.out",  
            // Point value of compilation.  
            "points" : 5  
        },  
        {  
            "title" : "C++ - Execution",  
            "command" : "./a.out",  
            // Point value of correct output.  
            "points" : 15,  
            "validation" : [  
                {  
                    "method" : "diff",  
                    "actual_file" : "STDOUT.txt",  
                    "description" : "Program Output",  
                    "expected_file" : "test1_output.txt"  
                }  
            ]  
        }  
    ]  
}
```