Curator System

Student Guide

Table of Contents

1	Introduction	2
2	Submitting an Assignment	3
3	Submitting Responses to a Quiz	12
4	Announcements	14
5	How the Curator Scores Auto-Graded Submissions	15
6	Using Submissions Effectively	21
7	The Curator and the Virginia Tech Honor Code	22

Appendices:

I	Some Testing Issues	23
II	Deadly Sins	24
III	The Curator Password	25
IV	Known Bugs and Alarming Behaviors	27
V	Getting Help	27

Disclaimer: Every effort has been made to ensure that the contents of this document are accurate and complete. However, the Curator Project is slowly ongoing. It is always possible improvements and bug fixes have been implemented since you obtained this document. The current version of this manual will be available from the Curator Project Homepage:

http://www.cs.vt.edu/curator/

1 Introduction

The Curator System (also known as Curator) allows a course instructor to:

- collect and automatically evaluate the behavior of student solutions to programming assignments
- collect and archive student solutions to any kind of assignment for later grading
- collect and archive student responses to multiple-choice assignments for later grading
- post announcements and custom individual reports for student access

The Curator provides a web-based interface, which you will use to submit assignments. This document describes how to use the Curator interface to submit your work, confirm submissions, and check results for automatically graded assignments.

To submit an assignment, you must use your web browser (e.g., Microsoft Internet Explorer, Netscape, Mozilla) to view the appropriate submission page for your course. Your course instructor may announce a link in class, but the current links should always be available at the Curator Project Homepage. This will display a submission client, which we will refer to as the Curator Client.

You will be required to log into the system to confirm your identity, and that you are a recognized member of the appropriate course. You will then find yourself with a variety of options, depending upon how your course instructor is using the Curator System. The following sections of this manual describe those options and how to use them.

In general, the Curator Client will transfer your submission to the computer on which the Curator Server is running, and provide confirmation that the submission was successful. For auto-graded submissions, a general description of how the Curator scores submissions is given in Section 5.

The Curator is a client-server application written primarily in the Java programming language, developed by the Virginia Tech Department of Computer Science, with support from Virginia Tech and Microsoft Research Corporation. The Curator is the latest in a sequence of automated grading systems that have been used by the Virginia Tech Computer Science department over a period of more than 30 years.

The Curator System was first deployed in 1998 and various versions of the system have collected over one hundred thousand submissions from over ten thousand of students.

2 Submitting an Assignment

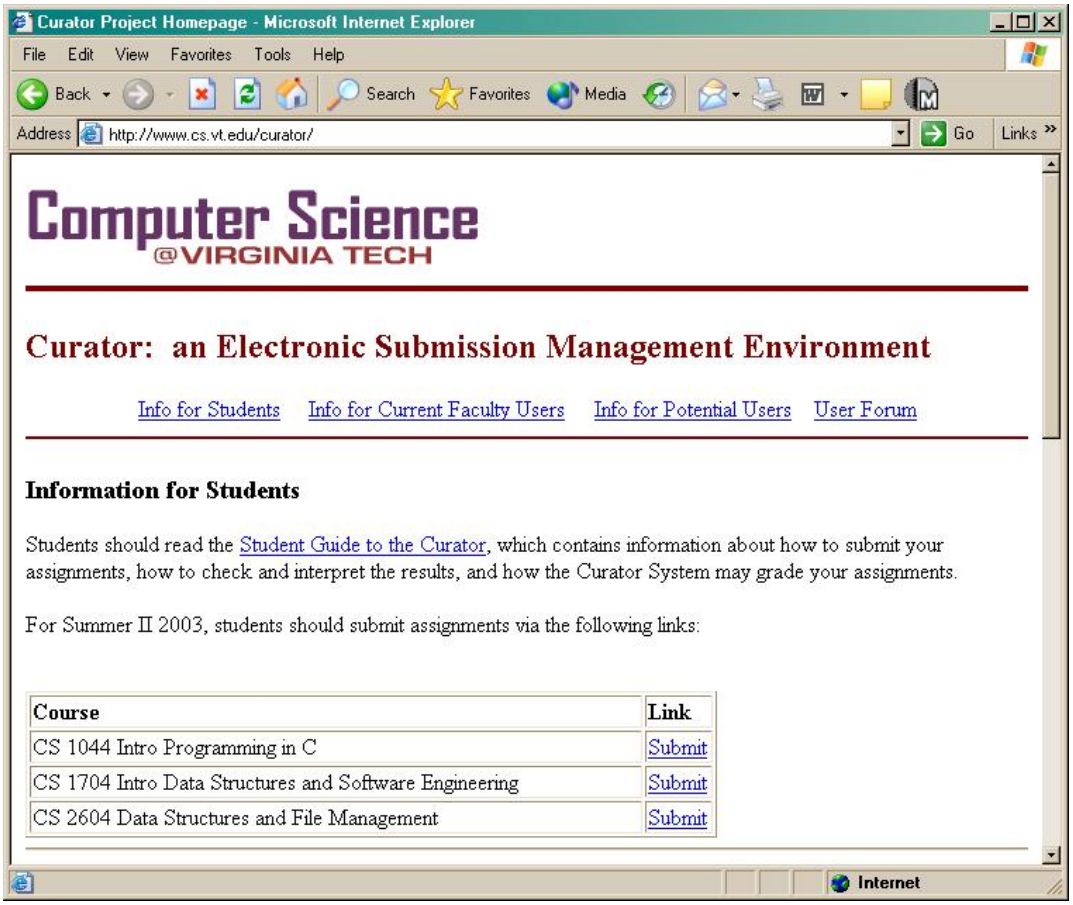
This section describes how to use the Curator Client to submit a file to the Curator Server. No special setup is required on your computer before you can submit to the Curator Server.

The Curator Client is a Java servlet, which means it is accessed via your Web browser.

Connect to the Curator Client on the Web:

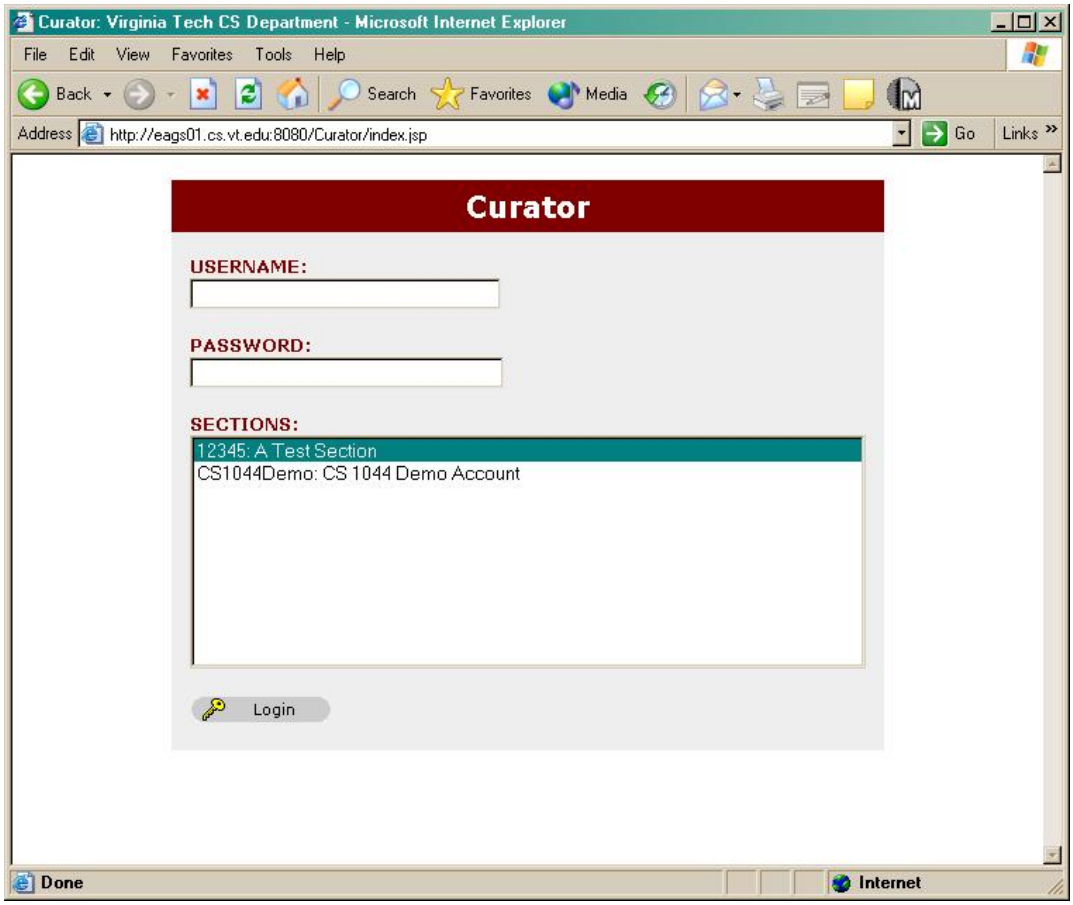
To run the Curator Client, start your WWW browser and enter the appropriate URL or follow the link for your course which will be posted on the Curator Project Website at http://www.cs.vt.edu/curator/.

There will be a table of courses and links near the top of the main page:



Just click on the link that is listed for your course to connect to the correct Curator Server.

A login page will load, similar to:



Fill Out the Login Form:

The Curator Client login form must be filled out in order for you to log onto the Curator System and make a submission, or check the status of your old submissions. You must fill out all of the information correctly in order to log onto the system. Each of the fields in the form is discussed in detail below:

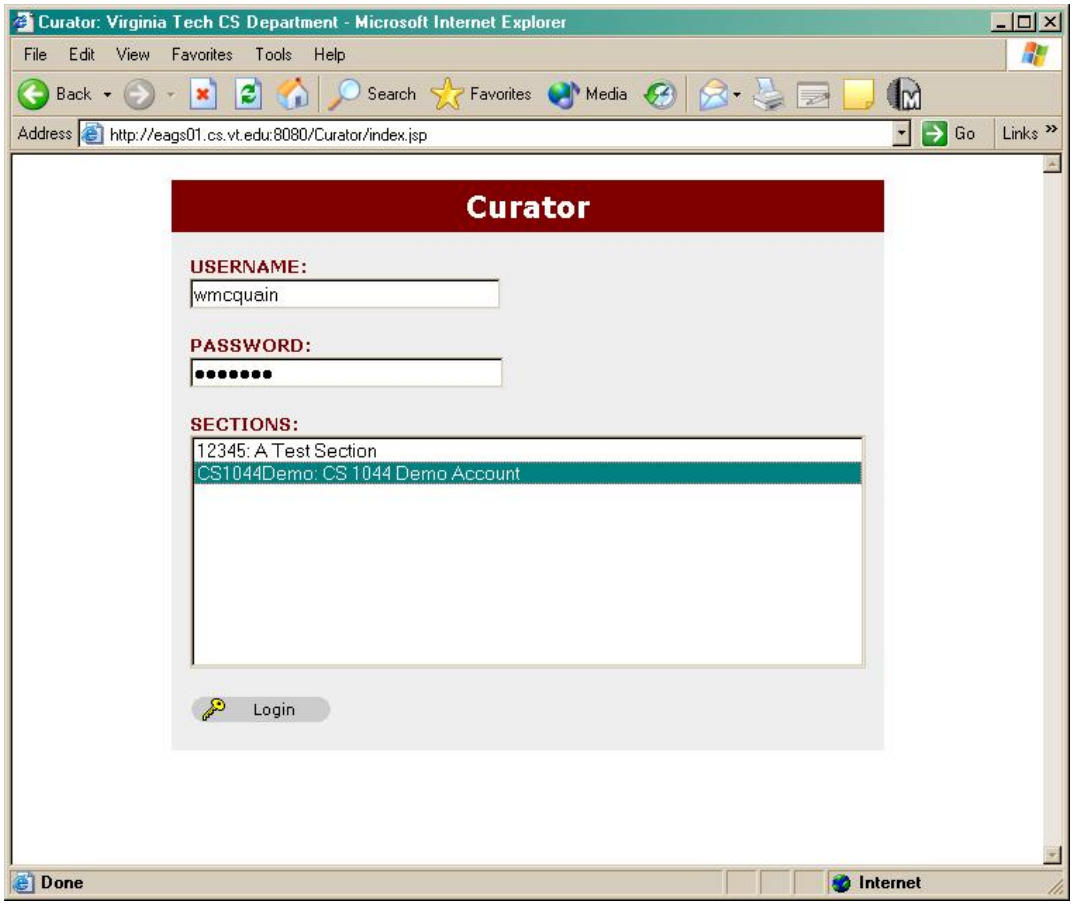
- **Username**: Enter your University PID (this is the same name you use when you connect to the university network or check your e-mail). This must be your original PID, not an e-mail alias. (Note: don't include `@vt.edu`)

- **Password:** Type in your password. This may either be the same password you use to connect to the University network or check your e-mail, OR a dedicated password used specifically for your Curator System account. In the latter case, you will receive your initial Curator password in an email message prior to the deadline for your first assignment. See the Appendix on Curator passwords for details.

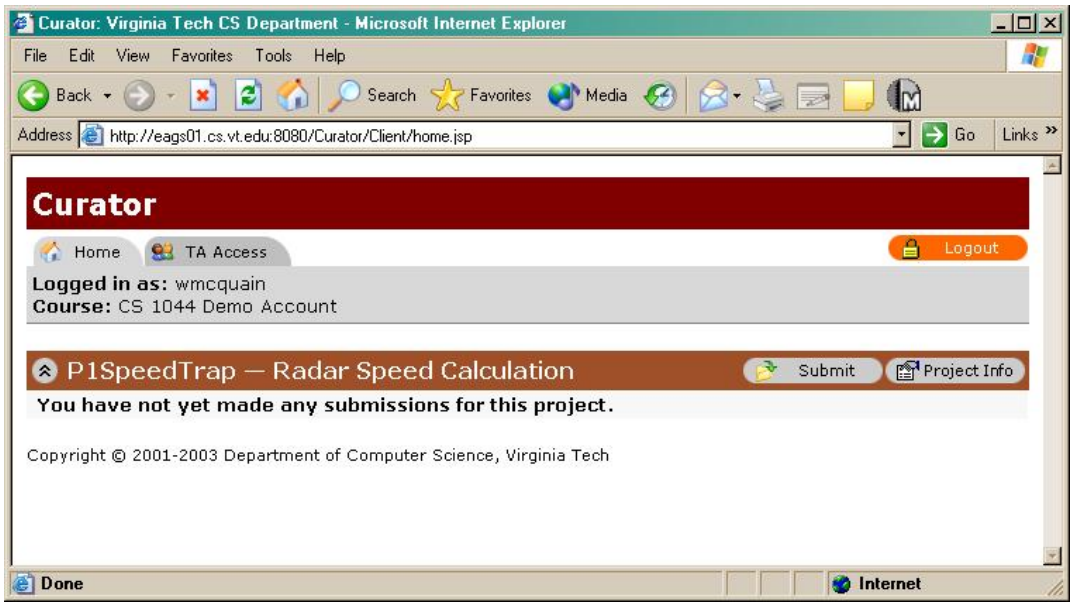
  As always, be careful that you remember your password, that you keep it secret, and that you type it correctly.

- **Section:** Click on the correct class section listed in the sections box. If you pick the wrong section, you will probably not be able to log in. If your section is not listed then either there is no account for it or you have connected to the wrong Curator Server.

  At this point the Curator Client window should look something like:



Click the Login button to log on to the Curator System. If you've entered all the information correctly you will see your Curator Home page:



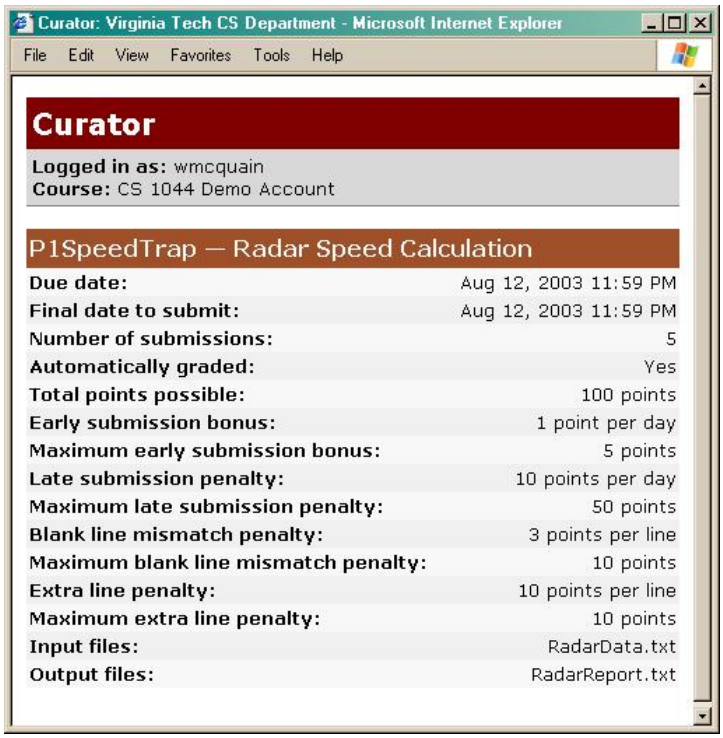
This page lists all available assignments (if any), your submissions (if any), with timestamp, file size and score (if applicable). The TA Access tab will be inoperable for you. To disconnect from your Curator account, click the Logout button. Always log out when you are finished with your submission, especially if you are using a computer that is accessible to anyone else.

Checking Assignment Properties:

Choose the relevant assignment from your Curator Home page. If the assignment you are looking for is not listed, then it is simply not available. If you have questions about that, they should be addressed to your course instructor.

Once you have selected an assignment, you can click on the Project Info button to display information about the assignment, such as due dates. Here's one for an assignment that is collected but not auto-graded:

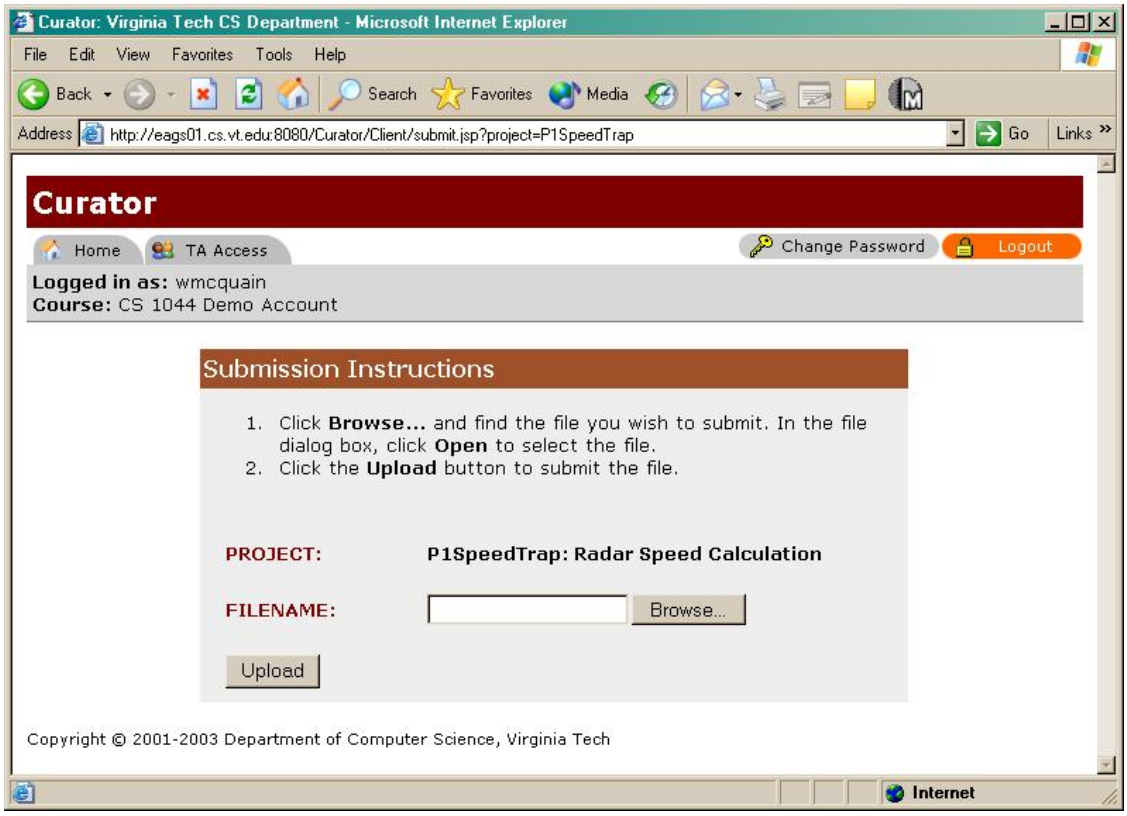
And here's one for an auto-graded assignment:



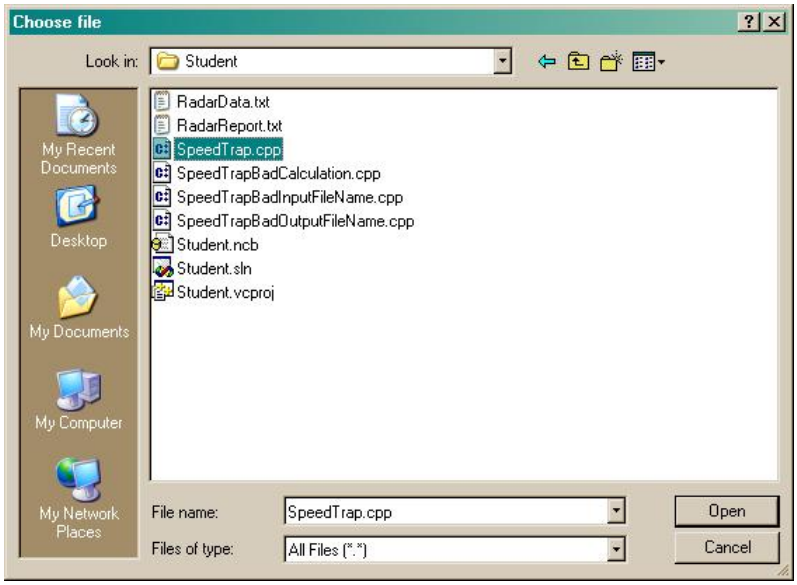
We won't explain all of the entries at this point. Some of them are obvious; the rest will be explained in the section that describes how the Curator's auto-grading works.

Make Your Submission:

To submit an assignment, click on the Submit button for the assignment, which will take you to the Curator submission page:

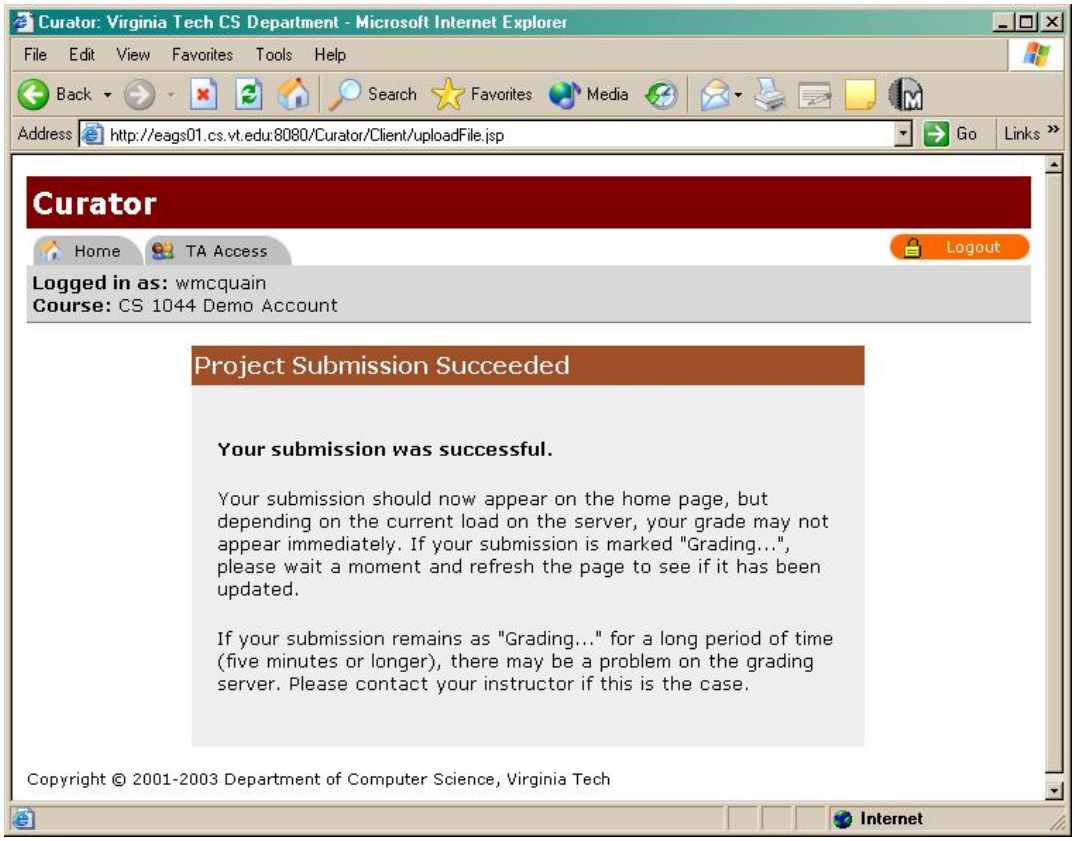


Click Browse to open a file selection dialog box. Select the file you wish to submit and click on Open:



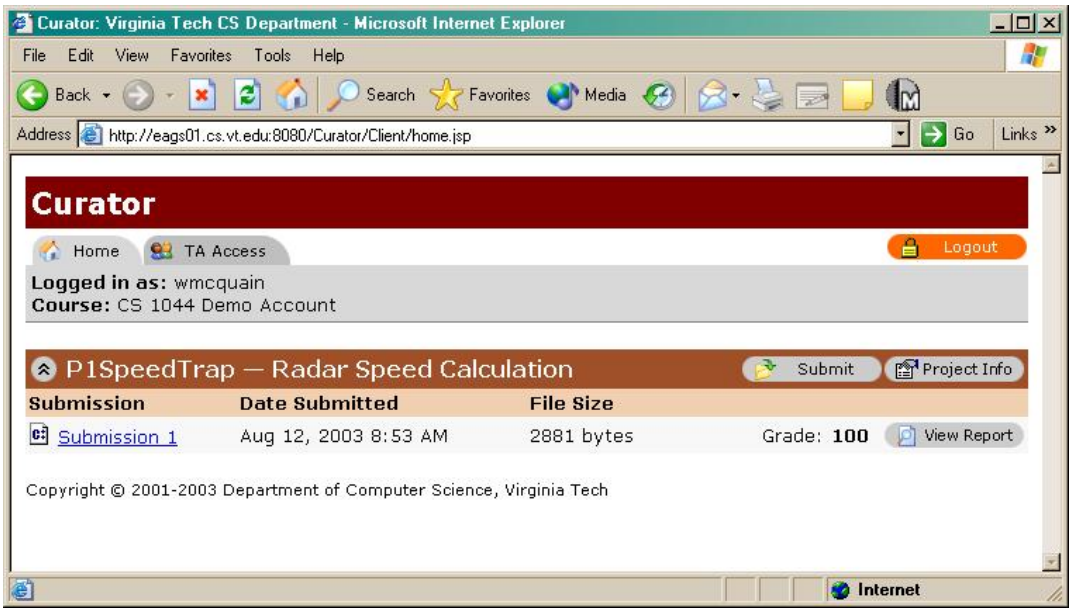
Be sure to submit the correct file. There is, in general, absolutely no way for the Curator to check whether you have selected the right file. If you submit the wrong file you will usually find you have wasted a submission. In order to actually submit your file, click the Upload button.

You should then see the following page, confirming the upload has been completed:

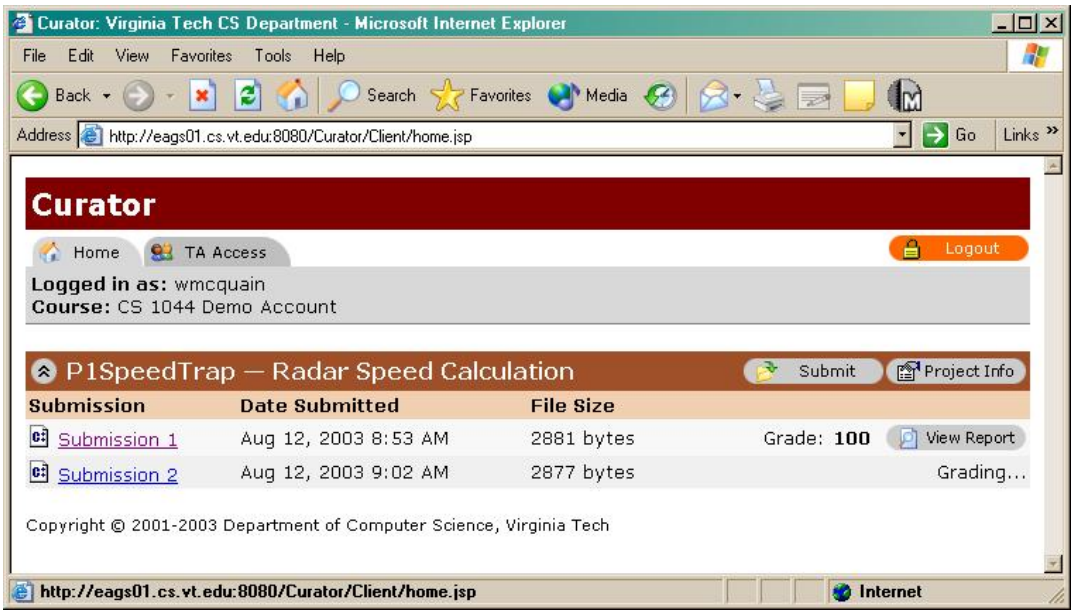


If you try to submit a file after using all your submission, you will see an error message. You may also see an error message in other situations, such as trying to submit after the assignment deadline.

Return to your Curator Home page to confirm your submission has been archived and logged. The submission you just made should be added to the table on your Curator home page:



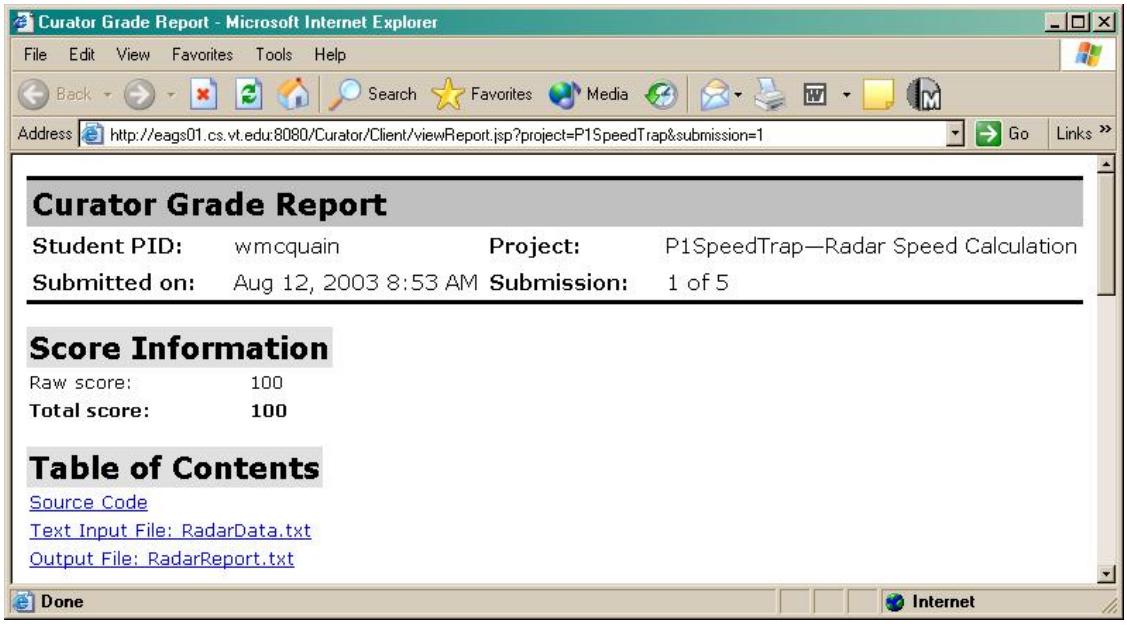
If your assignment is auto-graded, sometimes it takes a few seconds, or minutes, for your score to be available. In that case, you may see the message “Grading” instead of your score:



Just toggle the "refresh" button in your web browser a few times. In rare cases, grading of an assignment may be suspended while maintenance is being performed. Do not re-submit if the “Grading” message has not been replaced by a score; you’ll probably just waste submissions if you do.

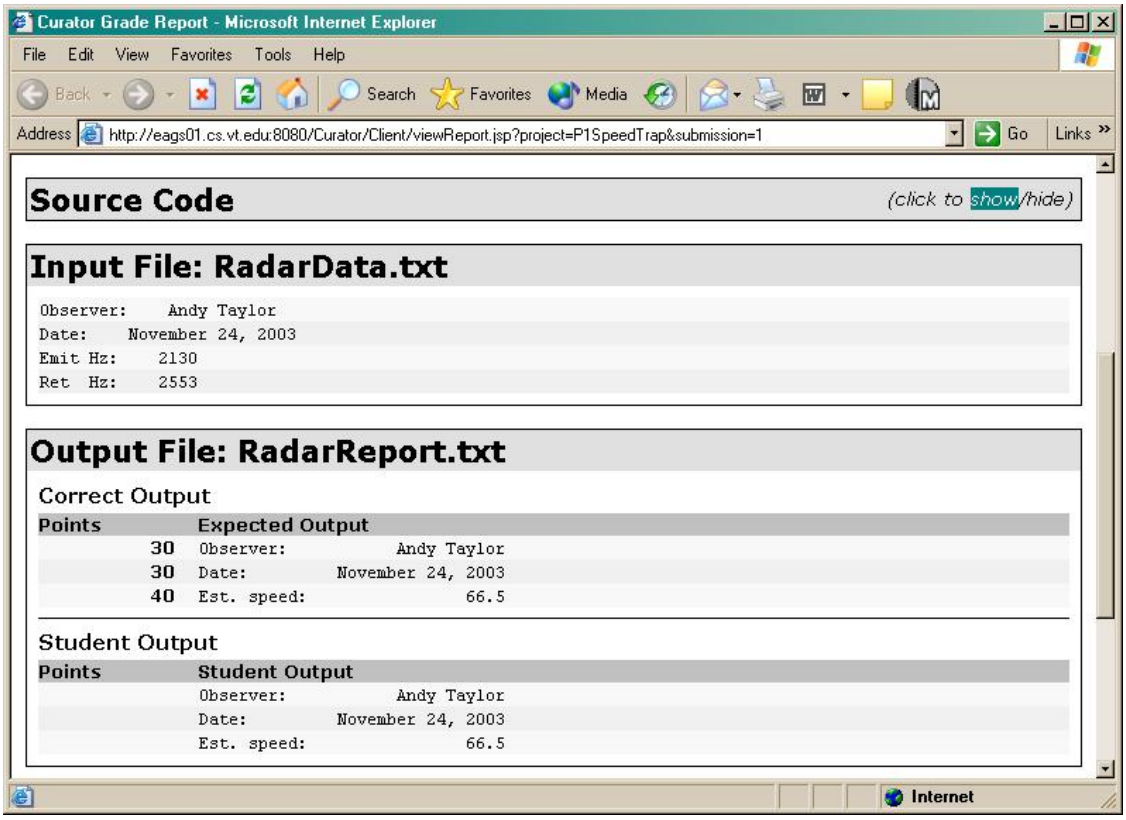
Getting Results:

The type of response you receive from the Curator depends on whether your assignment is to be auto-graded, or archived for later grading. For an auto-graded assignment, once the score is displayed, click the View Report button to view detailed scoring information. The Report window begins with a header:



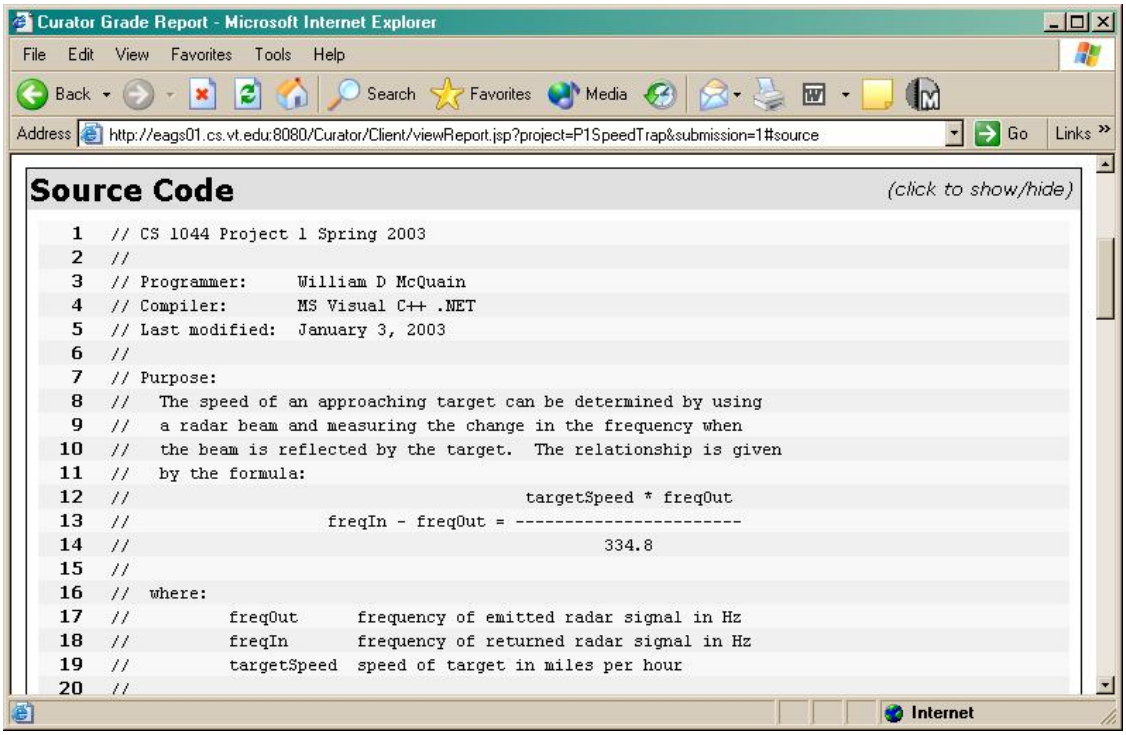
The table of contents provides links to the source code you submitted, and to any relevant input and output files that were used in testing your submission. The basic structure of the report is simple; there is a sequence of sections, containing your submitted source code, followed by any input files that were used in testing, and then by the correct output that should have been produced, and then by the output that your program actually produced.

Here is a simple example for a very simple progam:

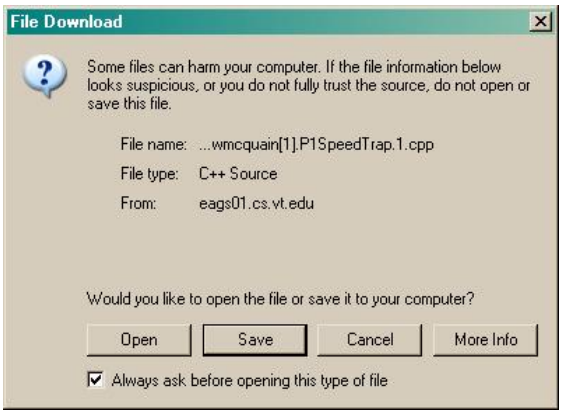


A following section explains how to interpret the scoring information provided in the Report.

Clicking on the show/hide link allows you to view the contents of your submitted file:



You can also download the file you submitted by clicking on the link provided on your Curator Home page; for example, right-clicking on the Submission 1 link shown above, and choosing Save Target, raises the following dialog using Internet Explorer:

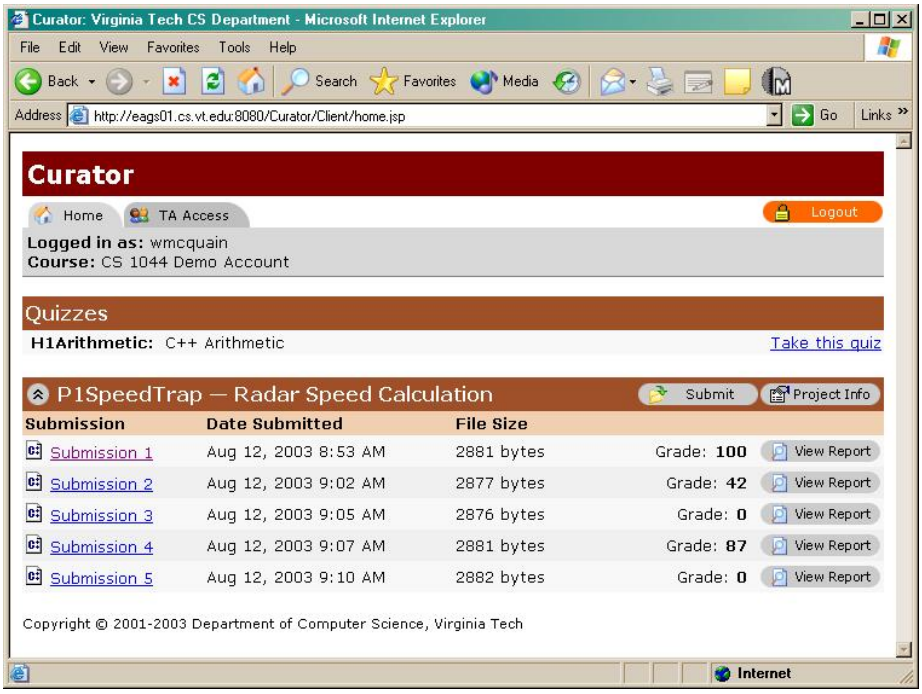


Note that the Curator saves your submitted file with a name that it generates, not the name you originally used. Note also that your Web browser may modify that name when you download your submission (as shown above). Be sure you note where you save the file and what it is called.

For assignments that are not auto-graded, there is no Report link; the only confirmation is that the file is listed on your Curator Home page.

3 Submitting Responses to a Quiz

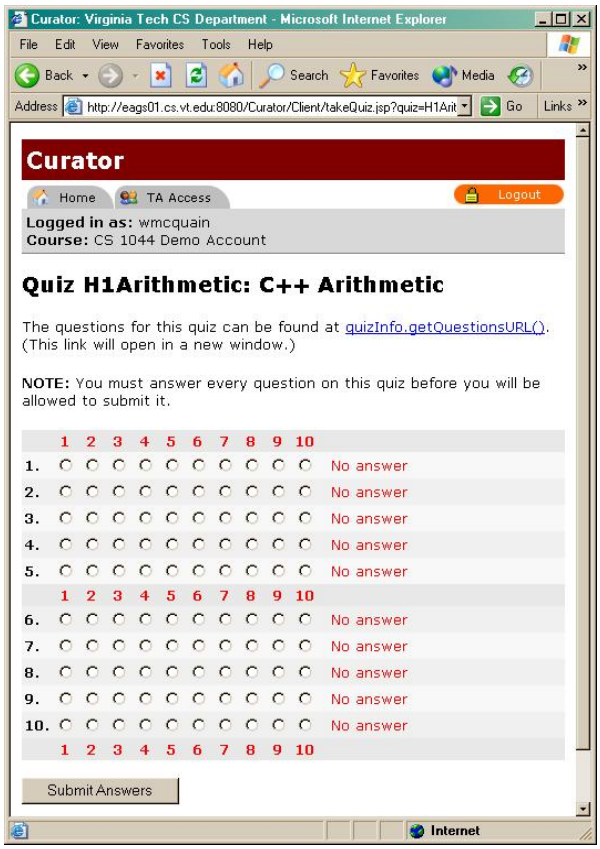
Your instructor may also use the Curator to collect your responses to a multiple-choice assignment. These will be listed as Quizzes on your Curator Home page:



Only one submission is allowed for Quizzes. Clicking the link brings up a "virtual opscan" page:

Your instructor may provide a link here to the questions that you are to answer, or may provide those questions in some other way.

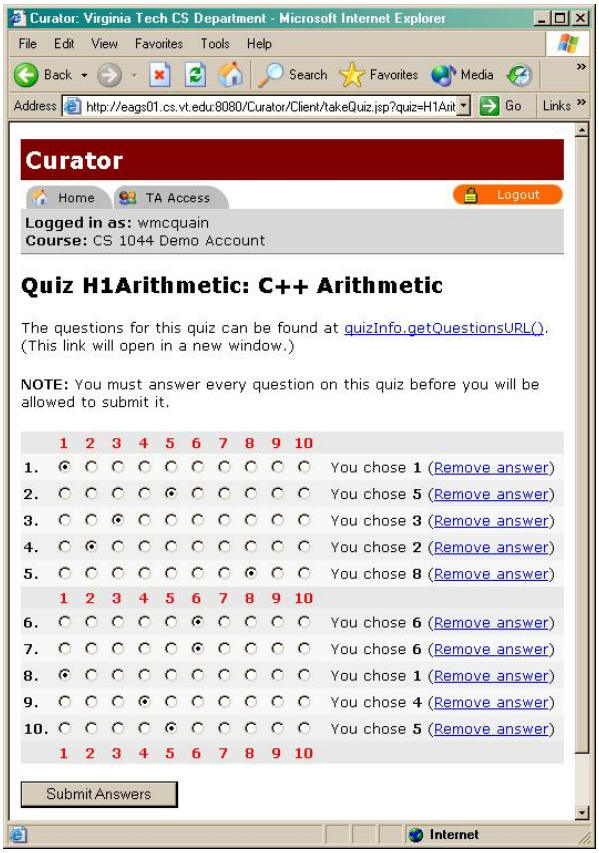
Your instructor may require that you select an answer to every question before you are allowed to submit your answers, or not.



Just click to select your responses.

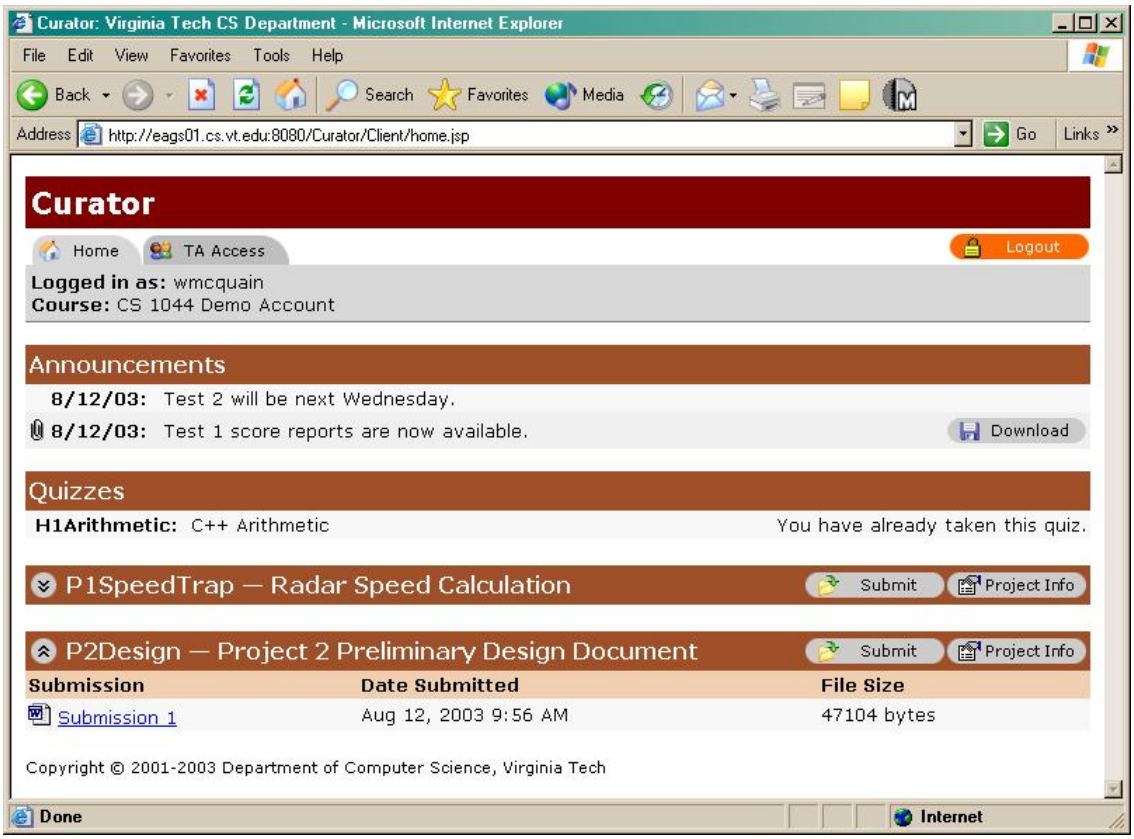
When you are done, just click the Submit Answers button to upload your answers to the Curator. Be careful! Once you've submitted your answers, there is absolutely no way for you to change any of them!

Quizzes are not auto-graded. How your instructor grades your response, and provides the results to you, will vary.



4 Announcements

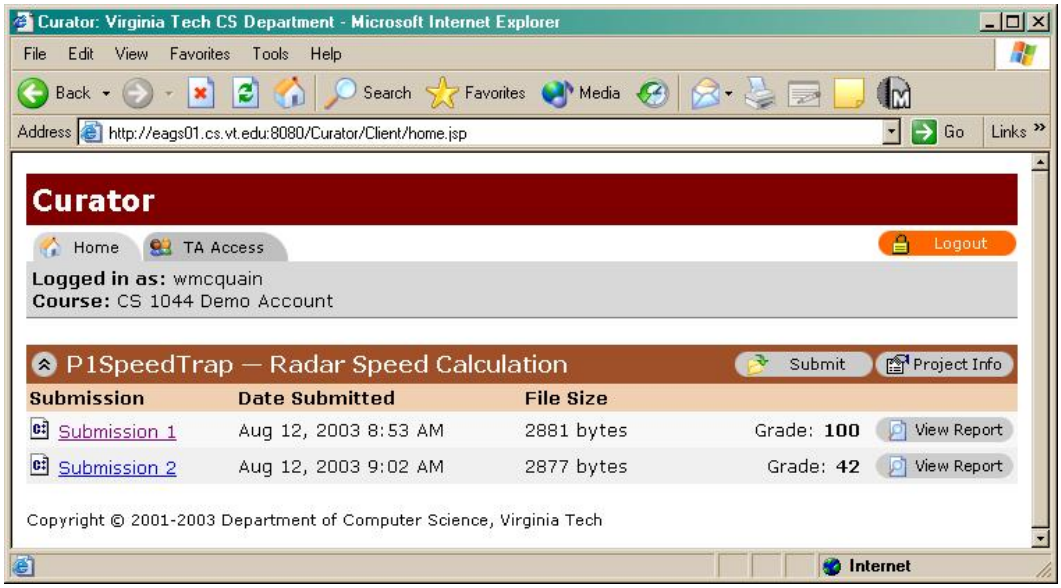
The Curator may also be used to post announcements to students, optionally including personalized attachments such as grade reports. For example:



Here, you have two announcements. The second includes an attachment that is presumably a file containing a score report for Test 1. Clicking on the Download button will raise a file download dialog box to retrieve the file to your computer.

5 How the Curator Scores Auto-Graded Submissions

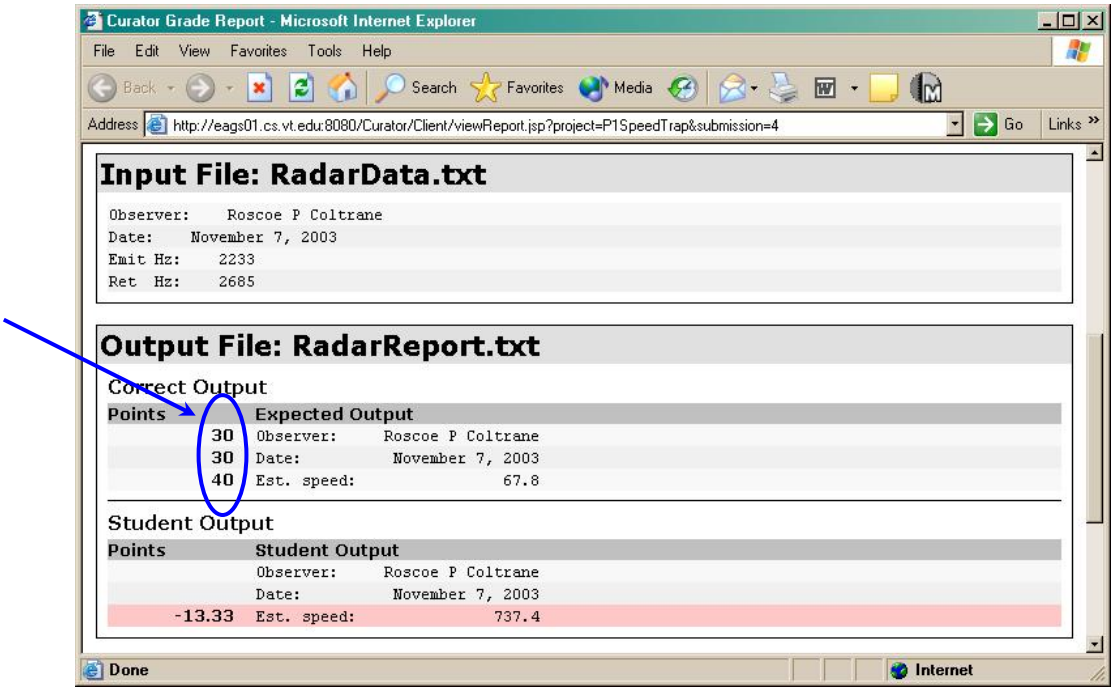
If your assignment is automatically graded you may not always receive a perfect score:



In that case usually want to examine the Report carefully in order to determine what was wrong and how to correct it. First, you must understand just how the Curator determines the score for a submission.

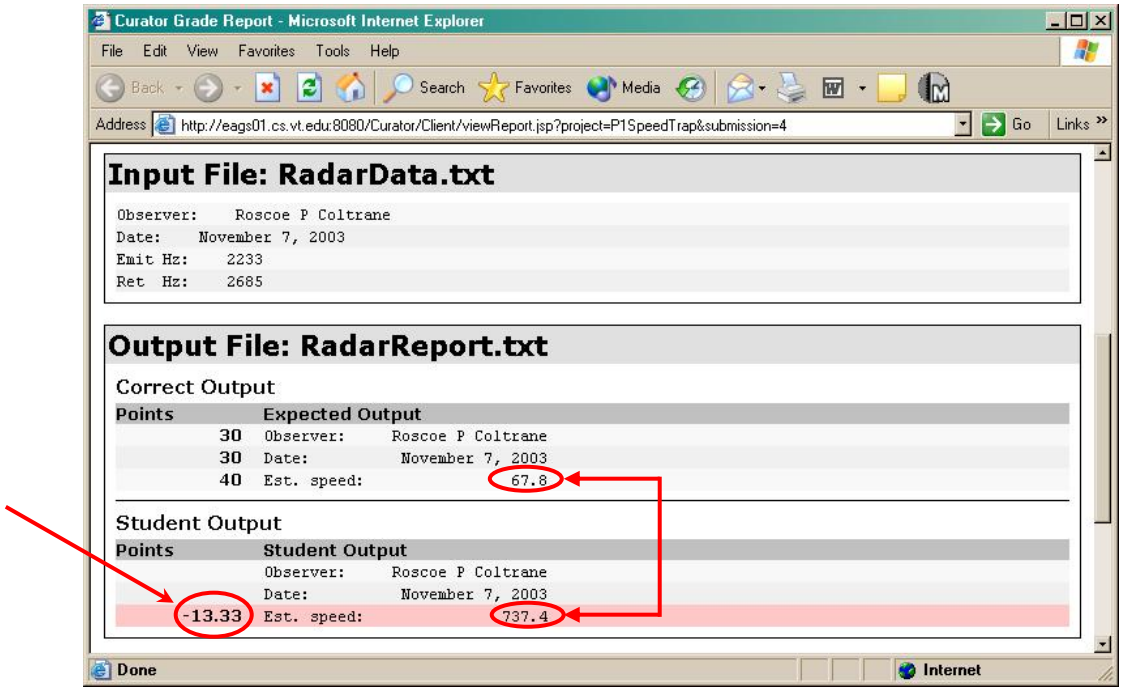
The score depends on whether the output produced by your submission matches the officially correct output. Normally, your course instructor will provide a special solution to the assignment. Each time you make a submission, the instructor solution will create the input data that will be used in testing your submission, generating a different set of input values for each submission. The instructor solution will also produce the correct output, based upon the input that it generated. In most cases, there is a file of single input data and a single file of output; that is the only case we will consider here.

When the instructor solution generates the correct output, it also assigns a point value to each line of output data. These point values are displayed in the Report:



If no point value is displayed for a line, then no points were assigned to that line. Warning: even though a line may not be assigned points, it may still affect the scoring of your submission. Read the description given below very carefully!

Deductions may also be shown in the Report, preceding lines in the Student Output section:



In this case, there is only one line containing an error, and that led to a total deduction of 13.33 points (roughly). When saving score, the Curator will truncate the decimal values to integer values at some point, and so the recorded score for the submission shown above would be 87.

So why were 13.33 points deducted on that line? First of all, you must compare the corresponding lines in the Expected and Student Output sections. It's quickly apparent that there's a difference in the value reported: 67.8 was expected but 737.4 was produced. So, that's wrong. (And that's because of an error in the program that was submitted, which is the fault of the programmer, and which the programmer must fix.)

Now, what's the actual scoring logic? The Curator compares the expected output to the student output, line by line (almost). When comparing two lines, the Curator breaks each line into a sequence of strings or tokens, deciding where to divide things up by looking for whitespace characters (like spaces and tabs). Looking at the expected output, the Curator finds three tokens: `"Est."`, `"speed:"` and `"67.8"`.

The Curator sees that line is worth a total of 40 points, and divides those evenly among the tokens it found, so each is worth about 13.33 points.

The Curator applies the same logic to tokenize the line of student output, finding the tokens: `"Est."`, `"speed:"` and `"737.4"`.

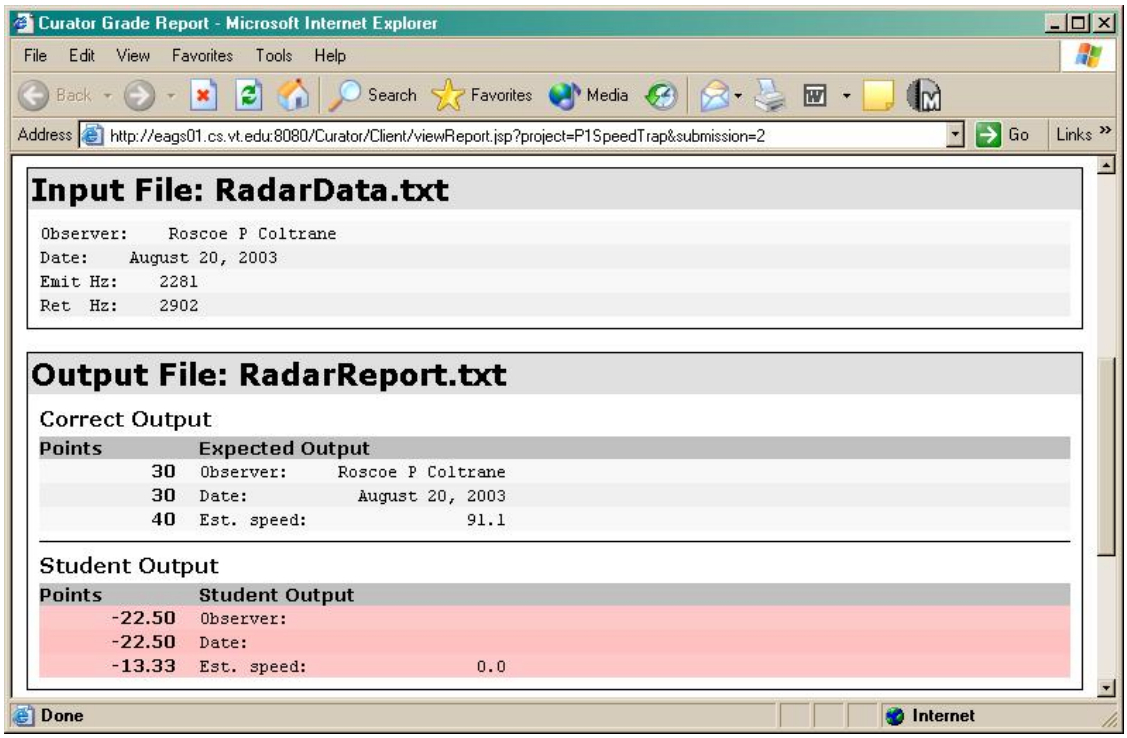
It then compares the tokens, in the order they were found. If two compared tokens are <u>exactly</u> the same, that's fine. If there is <u>any</u> difference whatsoever, then the Curator deducts all the points that the token is worth. So, here the Curator would deduct the value of one token.

There are also special rules for odd cases, such as when the number of tokens is different between the expected and student output, but the details are essentially boring. The important point is that the Curator compares output, line by line, and does comparisons that require an exact match. So, remember that spelling and punctuation DO matter. If the student output had included `"Speed:"` or `"sped:"` or `"speed"` instead of `"speed:"`, that would have resulted in a deduction.

Now, what about spacing? Don't insert extra spaces or tabs where they aren't supposed to be. Don't omit spaces or tabs when they are needed. Either of those will mess up the tokenization and result in deductions. For example, if the student output above had been `"speed :"` instead of `"speed:"`, the extra space before the colon would have caused the Curator to find four tokens in the student output line instead of three. That would have caused a mismatch and resulted in a deduction of 26.66 points (there'd be an extra token besides the mismatch).

But, spacing doesn't matter except in those two cases! For example, it doesn't matter whether `"Observer:"` and `"Roscoe"` in the first line are separated by four spaces, or five spaces, or two spaces and a tab, as long as they are separated.

Here's a somewhat more complex example, resulting in an even worse score:



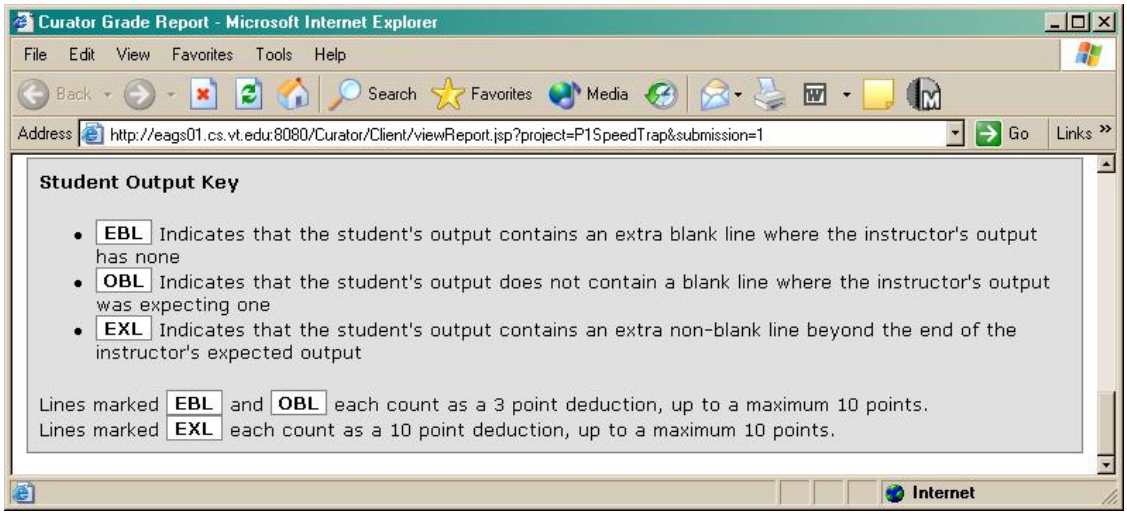
You should be able to determine why the particular deductions shown here were assessed.

Your score is determined entirely by a how nearly your output matches the expected output, and how early or late your submission is. One picky point: the input data and expected output are NOT created by the Curator itself, but by a program provided by your instructor. If you believe there's a problem with either of those, take that up with your instructor.

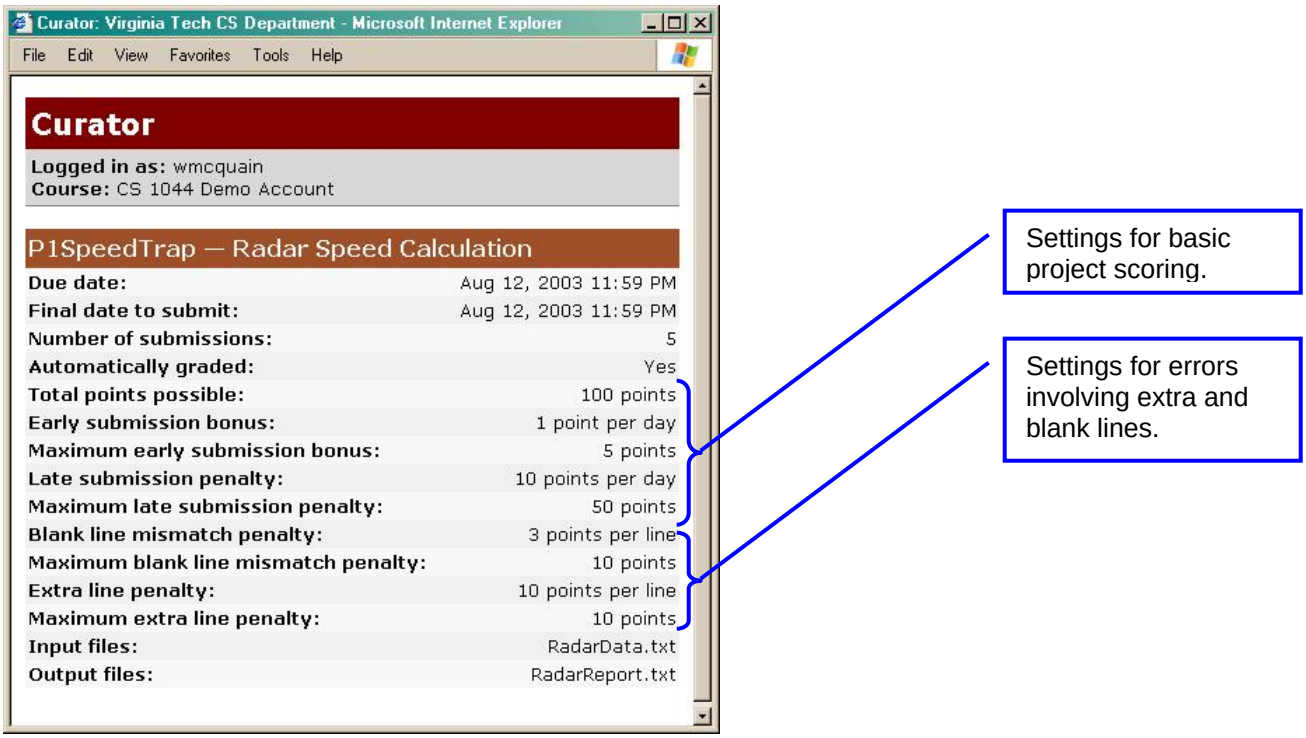
It may also be important that your output file not contain any extra lines, or omit any lines. If you do have extra lines that are not blank, or missing lines, then the Curator may compare the wrong lines, in which case you will probably receive a very low score.

If your output file has extra nonblank lines at the end, when compared to the expected output file then you may be penalized for each of those; the default deduction for each extra nonblank line is 10 points, but the actual value is set by your instructor. On the other hand, if your output has fewer lines than the correct output, then you're penalized the number of points that are assigned to those lines.

How extra or missing blank lines are handled depends on where they occur. Extra blank lines at the end of your output file should be ignored by the Curator. If you insert an extra blank line in the middle of your output file, the Curator may assess a penalty for that, determined by your instructor, and then attempt to resynchronize its comparison by reading the next line of your output. If you omit a specified blank line, that is handled in a similar manner. If any of these errors occur in your output, you will find special markers inserted in the student output section of the report:



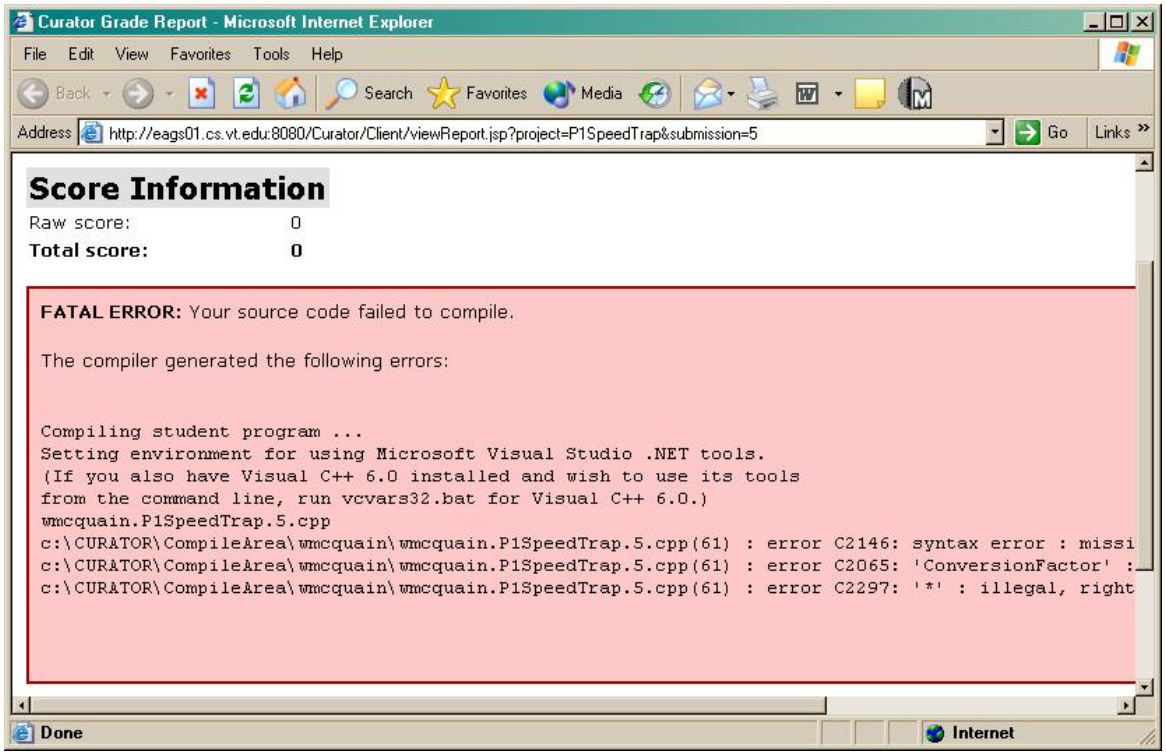
The Project Info button will show you the specific values your instructor has set for a number of possible deductions:



Other messages, other problems: The example Curator Reports discussed above are the most common. However, there are several other scenarios that you should be aware of.

What if your program does not compile? Of course, you should usually not submit a program that does not compile since that guarantees a score of zero. Two possible causes of this are using a different compiler (your instructor should specify the mandatory compiler and version for your course) or submitting the wrong file. There have even been instances where students have submitted term papers to the Curator – you shouldn't be surprised that a term paper usually doesn't compile properly as a program in C++ or any other programming language.

If your submission does not compile, you will receive a Report that contains the error messages from the compiler:



The Curator gives your program a fixed amount of time to finish (the default is 10 seconds). If your program has not finished within the time limit, the Curator kills your program. Depending upon the exact circumstances of the failure, you may receive a score based upon partial output from your program, or receive a score of zero and a Report that indicates your program took too long to execute and was killed.

When this happens, you need to do further testing and fix the problem before resubmitting. There is absolutely no point in resubmitting the same source code. This sort of problem may occur for a variety of reasons, including, but not limited to:

- Using the wrong name for the input file.
- Writing a loop that does not terminate properly.
- Submitting a program that is interactive and waits for user input.
- Using a variable that has not been properly initialized before use.

The Report will also include the input file and expected output, so you may use that input data to try to determine why your program misbehaved.

It is possible the Curator may terminate your program because it attempted an illegal action (divide by zero, writing too much output to disk, etc.) In that case, you will receive a score of zero for the submission, and the Report may contain some details of why your submission was terminated.

It is also possible that your program may be terminated abnormally (killed) by the operating system on the Curator machine before the time limit has expired. In that case, the Curator may or may not actually "know" that happened.

If the Curator is unaware, it will look for an output file to score as usual and you will receive a report similar to the one for a time-out or crash. However, if this happens you may notice that your output is incomplete since your program did not run to a normal termination. If that happens, you need to debug your program to eliminate the error before using another submission.

You may also find that when you run your program on your computer with the input data the Curator used, your program produces a complete output file and appears to operate correctly. This is annoying. It also still means that there is something wrong with your program.

If the Curator indicates your program is not producing correct results, but it does something different on your computer, the most likely explanation is that you are testing your program under a weak version of Windows (95/98/Me/XP Home). The Curator operates on a machine using Windows NT or better, which is much less forgiving of misbehaving programs than weaker versions of Windows. You may see different behavior because of a variety of errors, including those described above.

You may not be able to discover the source of the problem and fix it under a weaker version of Windows. In that case, you should test your program in the Computer Science Undergraduate Lab on one of the computers equipped with Windows 2000. Understand: the fact that your program appears to run correctly on your machine but not on the Curator machine does not mean that your program is correct.

The moral of all this is simple:

Follow the project specifications precisely and use a real OS!

6 Using Submissions Effectively

Some instructors allow some assignments, especially auto-graded programming assignments, to be submitted more than once. This gives students a chance to fix problems that were detected by the Curator. The number of submissions that are allowed depends on your instructor. The Curator server will automatically reject any "extra" submissions. How grades are assigned to multiple submissions are also up to the instructor. Some will use the highest grade from all of your submissions. Others may use the grade from your last submission. Others may use some other scheme.

If multiple submissions are allowed, the Curator will typically use a different input file each time. These input files are generated by a program provided by your course instructor, and should conform to the program specification provided for your assignment.

You should send another submission only after you have received the results from the previous submission, determined what errors you had in that version of your program, and attempted to fix those errors.

Tips on Using Submissions Effectively

Here are a few tips that can help you use your submissions effectively. Following these hints should help you get the most out of your programming assignments.

Test your program thoroughly before you send it. Sending in a program that has not been adequately tested will very likely result in a poor grade. Your instructor may have supplied some example input and output data with the assignment specification. Testing your program with only one input sample is not enough to assure that your program is correct. In fact, it is generally impossible to ensure your program is entirely correct merely by testing it. However, the more testing you do before submitting, the higher the probability you will achieve a good score.

The Curator uses its own set of input and output data, which will conform to your program specification, but may provide a more rigorous test of your program than your assignment's sample data. Try a wide range of input values, and study the output carefully to be sure that it is correct.

Do not use the Curator to do your testing. Test on your own computer. Use data supplied by your course instructor, data you create yourself, or data that you obtain from earlier submissions to the Curator. But do not submit a program to the Curator until you have confirmed that it performs correctly on all of the test data you have.

Use the results from a submission to diagnose your problem. When you get the results back from the Curator on your last submission, it will include the input data the Curator used to test your program, the correct output, and the output from your program. If points were deducted, study the correct output to see if you can determine what you are doing wrong. When you try to fix the problem, test it with the Curator input file so you can compare it to the correct output. It is important to remember that the Curator uses different input data for every submission, so make sure that you design your program so that it works for all possible ranges of input values, not just the assignment's sample input or the input that came back from a previous submission.

The bottom line on point deductions is this: the Curator only deducts points if your program is late or if there is a difference between the output your program produced and the correct output produced by your Instructor's solution. In some cases the difference may be subtle and you may have difficulty finding it. That is all part of the programming experience.

Be careful of deadlines. The clock on the Curator machine is synchronized with a timeserver, so we have great faith in its accuracy. Remember that deadlines are ill-tempered beasts. Hugging one too tightly may result in a bite.

7 The Curator and the Virginia Tech Honor Code

Each program you submit to the Curator is subject to the Virginia Tech Honor Code, just as if you had given the program to a human for evaluation.

Your Instructor will specify precisely what sources of help are allowed and how much, if any, collaboration with other students is allowed. Be certain you understand and follow the rules set by your Instructor. Ignorance of those rules is not an effective defense before the Virginia Tech Honor Court.

Regardless of the rules set by your Instructor, each of the following is considered a flagrant violation of the Virginia Tech Honor Code and/or Acceptable Use Policies, and will result in a formal charge:

- submitting a program designed to alter or circumvent the operation of the Curator's scoring mechanism
- submitting a program designed to crash or otherwise damage the operation of the Curator software or the machine on which it is installed
- editing or otherwise faking a grade report created by the Curator, and presenting that altered version when raising a question relating to the scoring of your program
- attempting to access or alter files on the machine on which the Curator is installed, whether physically or via a network connection; the sole exceptions would be normal use of the download options provided by the Curator Client

This list is not intended to be comprehensive; resolve any questions you have about these policies with the Instructor of your course.

All submissions to the Curator are archived. The programs submitted to the Curator may be automatically analyzed for suspicious similarities. When such similarities are found, the programs involved are compared (by humans) and charges are filed with the Virginia Tech Honor Court if the similarities warrant action.

> The Honor Code will be strictly enforced by the Instructors and GTAs who use and administer the Curator. All assignments submitted shall be considered pledged graded work, unless otherwise noted. All aspects of your work will be covered by the Honor System. Honesty in your academic work will develop into professional integrity. The faculty and students of Virginia Tech will not tolerate any form of academic dishonesty.

Appendix I Some Testing Issues

The Curator Server is currently running on Windows XP Professional.

Ideally, students will develop and test their programming assignments under Windows XP Pro as well. If that is not the case, students may discover that their programs behave differently during Curator testing than on their systems. It is impossible to give a comprehensive list of causes, but here are some that we have seen at Virginia Tech, where many of our introductory programming students are developing programs under Windows XP Home, or worse.

File Names: One of the most common errors we have seen is the use of an incorrect name for the input and/or output files. These names must be specified prominently in the assignment statement. Even so, many students will fail to follow instructions. Several scenarios may play out if a student uses the wrong name for the input file. The most common is that his/her program will produce a trivial output file, terminating normally. It is also possible that a design flaw may lead to an infinite loop or runtime error in this case. If a student uses the wrong name for the output file, the Curator will not find any output, assign a score of zero, and notify the student that no output was produced. This typically perplexes students because they are using the same incorrect file names when testing their program, and hence do not reproduce the error.

Array Indices: Failure to restrict array indices to valid ranges can lead to a plethora of unfortunate behaviors. Some of these will be exhibited regardless of the operating system being used, others only with systems that provide adequate memory protection for user processes.

It is possible for an out-of-bounds array index to cause no logical errors. In that case, the program may appear to be correct when tested under a weak version of Windows. Executed under Windows NT or better, the same program may exhibit a runtime access violation, or simply produce different results.

One nasty truth is that the behavior of a program containing this type of error may depend upon the contents of system memory and hence be nondeterministic. Another nasty truth is that memory protection faults are not reliably detected by the weaker Windows kernels. It is important to emphasize to students that the fact their program produces a correct result under a weak version of Windows does not mean their program is correct, merely that it caused no error that particular version of Windows is capable of detecting. Requiring students to make proper use of asserts or exceptions can alleviate this problem.

Appendix II Deadly Sins

Here are some simple things you can do to increase the probability you will receive low scores on your programs:

1. Submit an untested program.

Testing is the programmer's responsibility. Submissions to the Curator are precious; using the Curator to test your programs is a good way to ensure you will never receive a good score.

2. Submit an inadequately tested program.

Test data files are provided for each assignment. Usually several sets of input and correct output are provided. Just because your program works correctly with one test case doesn't imply it will work with others. In fact, it's usually impossible to test any program thoroughly enough to prove it's correct. That said, the more testing you do, the more likely you are to discover your errors yourself.

3. Use incorrect names for the input and/or output files.

Do this and you'll never find the data you're supposed to be processing… or you'll write your output somewhere the Curator will never look. Either way, you'll receive a very low score.

4. Fail to initialize all of your variables.

Do this and your program may behave differently on different computers. In fact, your program may behave differently every time you run it. Or maybe not… This is especially important with counters, running totals, and any array variable.

5. Fail to properly control array index values.

This is discussed in the course notes for any introductory programming class where arrays are used. All sorts of puzzling and nasty behavior can result.

6. Fail to check results and to take those results seriously.

Count on this: if the Curator deducts points, there is a difference between your output and the correct output. The core of the current scoring system has been used on many thousands of submissions, and there has never been a documented case in which points were deducted unless there was a discrepancy in the student output.

7. Submit the wrong file or omit necessary files when making a multi-file submission.

Careless error, common error, and costly.

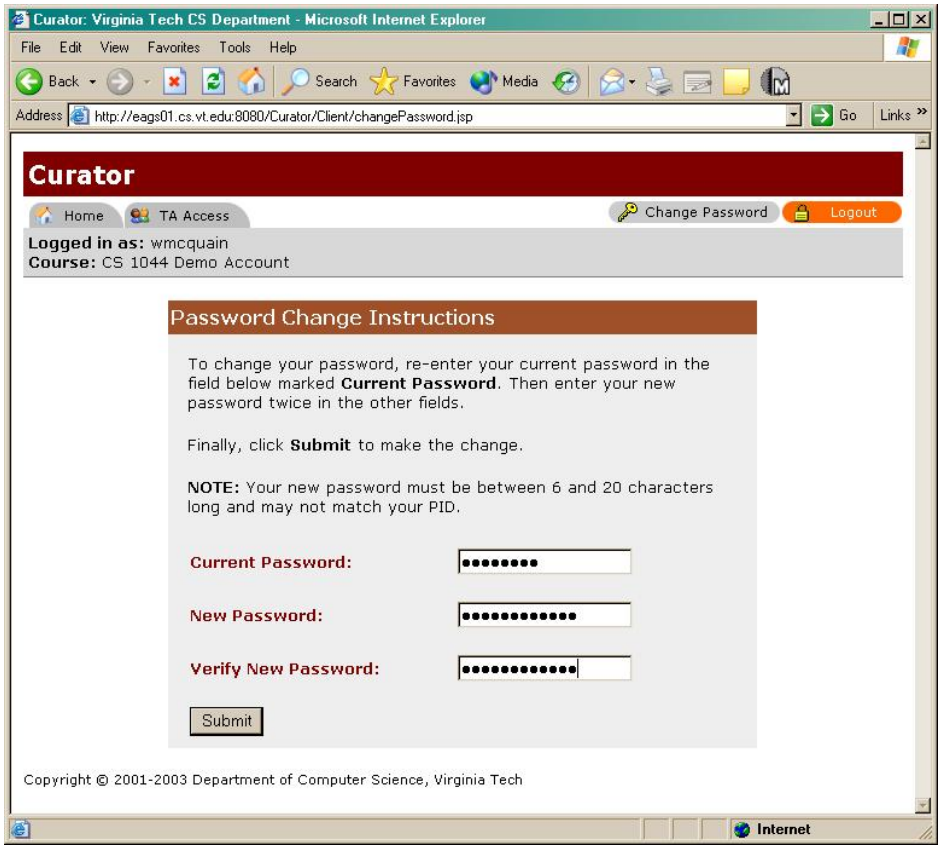
Appendix III The Curator Password

Note: this section applies only if the Curator's internal password authentication is used.

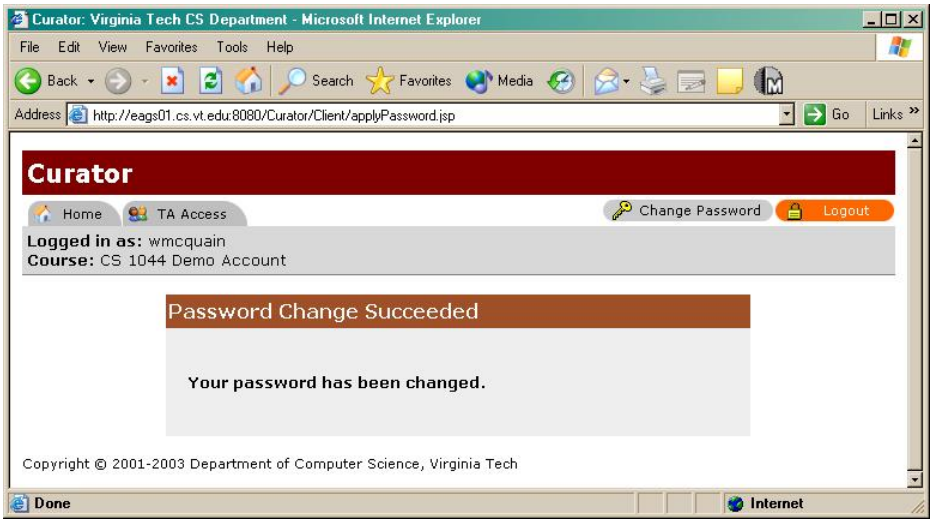
When your course instructor adds you to the internal class roll the Curator uses, the Curator will send you an email message notifying you that you now have a Curator account, and including your initial password for that account:

```
Date: Fri, 15 Aug 2003 14:29:13 -0400 (EDT)
From: grader@cs.vt.edu
Subject: Your Curator password has been assigned.
To: johokie@vt.edu

. . .

Server: 128.173.40.202
Login name: johokie
Password: XAbmW4QH


You should change this password once you log in to the Curator.
The new password must be between 6 and 20 characters in length
and cannot match your PID.
```

The first time you log into your Curator account, you must use the initial password generated by the Curator. You can, of course, opt to simply keep that password, but since it is generated randomly it is usually not easy to remember. In order to change your password, click the Change Password button at the top of your Curator homepage. This will take you to the form shown below. Enter your old password, enter your new password twice, and click the Submit button.



You will receive a confirmation message:



The next time you log into your Curator account, you must use your new password. No email notification is sent when you change your password.

What if I forget my password?

Don't. If you do, do not send email to the Curator system administrator requesting that your password be reset. All such requests will be ignored.

If you do, you must ask a TA or your course instructor to reset the password for you. That will generate a new, random, password that will be emailed to you.

Obviously, this can take some time, and that may prevent you from submitting an assignment on time, so it's best to just not forget your password.

Appendix IV Known Bugs and Alarming Behaviors

Curator Server Problems: In almost all cases, if a student program commits a runtime error (such as a divide by zero or an infinite loop), the Curator will simply kill the process, assign a score of zero, and proceed with the next submission. However, it is possible that some programs may misbehave in such a way that Windows XP prevents the Curator from killing the program. In such cases, the Curator may fail to score, or incorrectly score, any subsequent submissions from **that** student correctly until a human operator kills the offending program.

No submissions will be lost in this situation, and the Curator will rescore any pending project submissions, with the correct timestamp, once the misbehaving program is killed. You may, however receive multiple e-mail messages in the interim, each indicating a problem with your program and a score of zero. Don't be alarmed by this if you're sure your program does run properly, but notify your GTA or Instructor, or the Curator Administrator (`curator@cs.vt.edu`) to be sure the problem is fixed as quickly as possible.

Enrollment Issues: The Curator will indicate you are not enrolled in the class if your PID is not correctly recorded in the roll file the Curator uses. This may be because you selected the wrong section, or because there is an error in the roll file. If the Curator insists you are not enrolled in your section, contact your GTA or Instructor as soon as possible to resolve the problem. Note that the Curator uses an internal roll file; it does not access University records to confirm your enrollment.

Appendix V Getting Help and Reporting Problems

Sources of Help: The best first source of help is this manual. Read it carefully.

If you need additional help using the Curator Client, see your Instructor or your GTA, not necessarily in that order. If you have questions about the score your program received, always bring or e-mail a copy of the score report.

Bug Reports: Problems and possible bugs can be reported to the Curator Administrator by e-mail:

`curator@cs.vt.edu`.

Do not expect an immediate reply to email sent to this address. The Curator project staff is at low ebb. You may also report problems on the CS Forum board for Curator issues at `forum.cs.vt.edu`.