

# Redesign of the Online Calculus Placement Test at WPI

An Interactive Qualifying Project Report

submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

---

Ryan Bedell

---

Lawson Glidden

---

Matthew Kelly

---

Alex Ramsey

Date: June 2, 2009

Approved:

---

Arthur Heinricher, Dean of Undergraduate Studies - Advisor

1. Calculus
2. Education
3. On-line testing

## **Abstract**

In 2007, WPI implemented online Calculus Placement Tests. During the 2007-2008 academic year, a statistical analysis was performed for this new system. The same system was used again in 2008 and this project repeats the previous analysis and also introduces two significant changes to improve the original system. One change reduces the number of tests and simplifies the recommendation algorithm. The second change is a system to tailor the recommendation to each student's intended major and background.

## Contents

1. Introduction.....	1
2. Background.....	3
2.1 Overview of the 2008 Report.....	3
2.2 Research Methods.....	6
2.3 Comparison of the WPI Placement Test with Other Schools' Exams.....	14
2.4 Special Features of the WPI Placement Test.....	17
3. Project Goals .....	19
4. Analysis of Scores on the Calculus Placement Tests: .....	20
5. Merging the Three Tests .....	30
6. Digital Advising Algorithm .....	35
6.1 Recursive Algorithm for Determining Recommended Course Background.....	35
6.2 An Analysis of Recommended Course Groupings .....	37
6.3 Placement Test Algorithm Explained through Flowcharts .....	40
7. Improvements to Recommendation System.....	42
7.1 Introduction to WeBWork.....	42
7.2 Problems with WeBWork.....	42
7.3 Things no one told you about WeBWork.....	42
7.4 The Requirements .....	43
7.5 The Options .....	44
7.6 The Solution.....	44
8. Conclusion and Recommendations:.....	47
8.1 Review of Accomplishments .....	47
8.2 Recommendations for the Future .....	49
8.2.1 Performance Comparison.....	49
8.2.2 Compliance Comparison.....	49
8.2.3 Jumpstart Redesign .....	49
8.2.4 Using Major Data.....	50
9. Appendix.....	51
9.1 Digital Advising Algorithm Flowcharts.....	51
9.1.1 Flowchart 9.1: Inputs.....	51
9.1.2 Flowchart 9.2: Background Method for Course Elimination.....	52

9.1.3	Flowchart 9.3: Pass or Fail the Tests .....	53
9.1.4	Flowchart 9.4: Recommendation .....	54
9.2	Perl CGI Script for Major Recommendations and Web Page Generation.....	55
9.3	Perl Macro Developed for WeBWork.....	65
9.4	Excell Macro .....	72
9.4.1	Visual Basic Code after Update .....	72
9.4.1	Excel Data Sheet .....	85
9.4.2	Excel List Sheet .....	86
	Works Cited.....	87

## 1. Introduction

Calculus Readiness Tests have been used at WPI for more than 20 years. Originally, the tests were given in August during New Student Orientation with the entire freshman class collected in Alden Hall on the WPI campus, given pencils and scratch paper and 1 hour to complete a 25-question multiple-choice test. The test included only pre-calculus material and the scores were used only to identify students who were not quite ready for the regular 7-week Calculus 1 class.

About 6 years ago, the same multiple-choice Calculus Readiness Test was made available on-line so that students could complete the test early in the summer before New Student Orientation. The answer key was also available on-line and students were asked to compute their grade on the test and email it to the Mathematical Sciences Department at WPI. (Not surprisingly, student scores went up dramatically under this new on-line system!) The online test made it possible for students to complete their course schedule early in the summer before coming to campus in August.

In 2007, WPI introduced a new Calculus Placement Test for incoming freshmen. In place of a single test with only pre-calculus material, there were now three different tests. The first was the old Calculus Readiness Test. The second test included material on derivatives and was used to determine if a student was ready to start in Calculus 2. The third test included material on integration and was used to determine if students were ready to start in Calculus 3.

There was one more fundamental change made in 2007: the new tests were not multiple-choice. The new tests were delivered using WeBWork, a sophisticated homework and testing tool developed at the University of Rochester. WeBWork is able to evaluate short-answer problems and, most importantly, recognizes equivalent formulas for an answer. It gives the students immediate feedback (whether their answer is correct or not), has a huge library of problems which are easy to use and edit and adapt, and provides a good tool for communicating with the students as they work on the test.

The 2008 two WPI students completed an analysis of the first year of the new online calculus placement tests.<sup>1</sup> That report concluded that the new tests worked well and produced good recommendations. This year's study will begin by verifying a portion of the 2008 IQP report results through identical statistical methods. With that said, there were still some problems with the original online Calculus Placement Tests. First, the original system consisted of three separate tests and although the students were asked to complete all three tests, many students completed only one of the tests. In addition, only the recommendation for the Calculus 1 Placement test was automated. There was no good way to know when a student had finished either the Calc 2 or the Calc 3 test. As a result, students could, and often did, receive more than one recommendation regarding their first course at WPI.

A second problem is the fact that the choice of a first calculus course often depends on the student's proposed major. A student majoring in IMGD may need only one mathematics class and that one class does not have to be calculus. A student majoring in Management Engineering will take a different sequence of math courses depending on their concentration inside the major.

The result of this IQP is a new version of the Calculus Placement Test in which the original three separate tests are combined into a single test. There is now one "trigger" used to send the recommendation. In addition, the software used to generate the Calculus 1 recommendation has been updated to include both Calculus 2 and Calculus 3 information. Students now receive a single letter covering all three topics. The projects also developed software and a new question that uses the student's proposed major to tailor his or her recommendation. This step requires a detailed study of the mathematics requirements in every major at WPI.

---

<sup>1</sup> Leary, S., D. Smaltz, (2008). *Analysis of the WPI On-Line Calculus Placement Tests*. Worcester, MA: WPI.

## 2. Background

### 2.1 Overview of the 2008 Report

In 2008, Leary and Smaltz<sup>2</sup> performed an analysis of the effectiveness of the first year of the online Calculus Placement Tests. That report includes two main sections. The first part of the report explains the WeBWork system. The second part of the report is a statistical analysis of the results, including a basic summary of statistics on test completion, recommendations given and recommendations followed, an analysis of the thresholds used in making the recommendations, and finally correlations between performance on the placement test and performance in the first calculus class. The report provided quantitative evidence that the new tests gave reasonable recommendations to incoming freshmen students at Worcester Polytechnic Institute.

The 2008 report provided an explanation of the WeBWork system and a comparison to the old pencil-and-paper Calculus Readiness Test. WeBWork was designed as a homework delivery system for mathematics and physics classes. With support from the National Science Foundation, the University of Rochester and several partner universities around the country have created an extensive library of problems for mathematics (including pre-calculus material like algebra and trigonometry as well as calculus, differential equations, and linear algebra), statistics and probability, and physics. The key advantage of WeBWork over other on-line testing and homework systems is that it can evaluate “short answers” in addition to simpler multiple choice questions. For example, a typical calculus problem could ask the student to compute the derivative of  $f(x) = 3x^2 + 7x - 100$ . A correct answer to the question would be  $f'(x) = 6x + 7$  or  $f'(x) = 3*2x + 7 - 0$  or  $f'(x) = 7 + 6x + \sin(2\pi)$ . WeBWork would recognize each of these as a correct answer and give the student credit for being right.

---

<sup>2</sup> Leary, S. and D. Smaltz

An analysis of student performance in A and B term of 2007 reveals the correlation between student performance during and matching Calculus Placement Test scores. This analysis breaks down the Calculus I placement test into four sub-scores: Algebra, Functions, Geometry, and Trigonometry. When evaluating the entire freshman class, it was found that in A term, there was an 11% NR rate. During the same term, 33% of students received an "A", while 32% of students received a "B" and 24 % of students received a "C". The results of the B term analysis were very similar. There is a weak but positive correlation between the placement test scores and the performance in A and B term.

The Internal Consistency section analyzes specific individual questions and determines if any of them are too difficult or too easy. This is important to consider because many questions have randomly-generated coefficients. It is important for the test to be consistently challenging (with a few exceptions), such that the differences in scores accurately represent differences in the students' abilities, not the difference in randomly assigned coefficients. Internal consistency is an objective way to evaluate the test's reliability utilizing available data. The internal consistency yields Cronbach's alpha, a number between zero and one, which can be used to assess the amount of random error in the test.

The project team analyzed the total time students spent on the test by extracted time stamps from the WeBWorK database. The difference in these two time stamps was considered the total time that was taken to complete the placement test. The study showed no significant correlation between performance on the test and the time that it took to complete. It did show a classic pattern observed in many online homework and testing systems. Students cluster into two main groups: one that completes the test in one sitting and another that returns to the test several times over several days. Test completion times ranged from 5.6 minutes to 35 days. It is unlikely that anyone would work for 35 days straight on the Calculus Placement Test. It is also unreasonable to believe that anyone could complete the tests in less than six minutes.



The next topic discussed the correlation between placement test scores and performance on the common final exam for Calculus 1. The correlation between the Calculus I Placement Test and General Final Exam was positive but not significant. The slope of the best-fit line was 0.8703 and the coefficient of determination of  $R^2 = 0.0139$ . Even when the lowest scores were removed, there was still a low correlation, with slope=0.3411 and  $R^2 = 0.0043$ . The 2008 project also studied the correlation between the placement score and the Basic Skills Exam. (This is a component of the common final exam that includes fundamental derivatives and is graded without partial credit.) This slope was 0.4256 and an  $R^2$  of 0.0252. Therefore, there was a positive but small correlation between the placement test score and performance on the final exam.

There was, however, a higher success rate in Calculus I by students who performed well on the Calculus Placement Tests. It is possible that the Calculus Placement Tests are an accurate estimator of a students' readiness for different calculus courses. It is also important to remember that a perfect correlation would not necessarily show that students had been placed in the correct course, just that the placement test and the final exam were measuring the same performance.

The report also compares Calculus Placement Test recommendations to the courses the student took. About 66% of the students followed the recommendation, while 18% took a more challenging class than that recommended and 12% took a class that would be less challenging. The remaining 4% includes students who either did not take a math course or took a non-calculus mathematics course.

Finally, last year's study evaluated the cutoff scores used to define the different recommendations. The approach was to change the cutoff scores and estimate the performance and placement of any students whose recommendation would have changed. The study concluded that there is no justification to change the thresholds.

## 2.2 Research Methods

The WPI library has a vast amount of free information on thousands of topics; most people, unfortunately, do not know that the library is such a powerful resource or they do not know how to use it. The library system is a bit convoluted, but it gives the user the opportunity to narrow down their search tremendously before they are forced to press a search button. The library website does this by allowing the user to select groups of databases based on subject, and then provides the user with details about what each database covers.

Once the user selects a database, they are able to search in different ways depending on the database selected. The power of these search tools varies, but a few features are true for all of them. Each of the search bars that were used in the WPI databases utilized Boolean searching. Boolean searching allows the user to use three different commands: “AND”, “OR”, and “NOT”. These commands can be entered between keywords or descriptors. The “AND” command will make the search find only items that have both the keywords that the “AND” is linking together. The “OR” command will require the search to have at least one of the terms that is linked together by the “OR”. The “NOT” command will eliminate search results with that keyword. For example, ((“Calculus” OR “College Mathematics”) AND (“Placement” OR “Student Placement” OR “Student Aptitude”) NOT (“SAT test”)) will perform a search which will include at least one term from the first group and one term from the second group, but will not include any results with the term “SAT test” in it.

One of the most powerful tools used was the descriptor-based search. This search type allows the researcher to search through descriptors instead of keywords. Descriptors are much more intuitive than keywords and yield better results. This is because descriptors search through the descriptive words that relate to each article and finds the one that matches the best instead of simply looking for words in the text.

In this case, the WPI library resources were used to find mathematics placement exams at other undergraduate schools. As a clear means to exemplify these resources, the search procedure that produced many useful placement exams is listed below.

1. Start out by going to the WPI library website: <http://www.wpi.edu/Academics/Library/> . At this web page, the researcher will see the Quick Start search area. From here, the different types of sources can be selected for more specific searches. The most useful area discovered here were the Articles; Select “All Databases & E-Resources”. (WPI George C. Gordon Library, 2009)



**Figure 2.1: Quick Start**

2. In this area, the researcher can narrow the search to a given subject, resource type, or reference. Each of these tools can be useful. Searching databases by subject, however, seems to be the most intuitive. It is important to note that what the item or items being searched for may be found under any number of subjects. For example, while searching for articles related to Calculus Placement Tests, the best documents were not located under the “Mathematics” subject, but were instead under “Education and Teaching”.

## Databases & Electronic Resources

Off-campus? WPI Community members may connect through our proxy server to gain access.

    
advanced | off campus | ?

### Databases by Title

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [All](#) | [By Course](#)

### Databases by Subject

Choose a subject area to find databases that contain references and abstracts to scholarly articles, conference papers, handbooks,

Education & Teaching ▼

- Select a Subject-- ▲
- Biology & Biotechnology
- Biomedical Engineering
- Business & Management
- Chemistry & Chemical Engineering
- Civil & Environmental Engineering
- Company
- Computer Science
- Economics
- Education & Teaching**
- Electrical Engineering
- Energy
- Environment
- Fire Protection Engineering
- General & Multidisciplinary
- Government & Law
- History
- Humanities & Arts
- Industry
- Interactive Media & Game Development ▼

Journal, newspaper, or magazine.  
Articles available online or leads you to the print version at the  
Library resources from off campus.

**Figure 2.2: Databases by Subject**

3. After selecting a subject, the researcher will be presented with databases relating to that subject.

These databases have a variety of search tools. As a result, different styles of search are required to

clean relevant sources from them. The database selected was Education Resources Information Center

or "ERIC." (Education Resources Information Center)

## Databases by Subject - Education & Teaching

See also [General & Multidisciplinary](#).


  
[advanced](#) | [off campus](#) | [?](#)
  
 For databases searched, see [below](#)

Database Title	In Quick Find	
 Academic OneFile	<input checked="" type="checkbox"/>	Search full-text articles on physical science, technology, medic
 EBSCOhost Business Source Premier	<input checked="" type="checkbox"/>	Includes full text for 2,300+ business journals, indexes 3,300+ 1 <a href="#">Journals Included</a>   <a href="#">Other Sources</a>   <a href="#">Tutorial</a>
 Educator's Reference Complete	<input checked="" type="checkbox"/>	Covers more than 450+ full-text academic journals, hundreds of child development and psychology, and best practices in educa

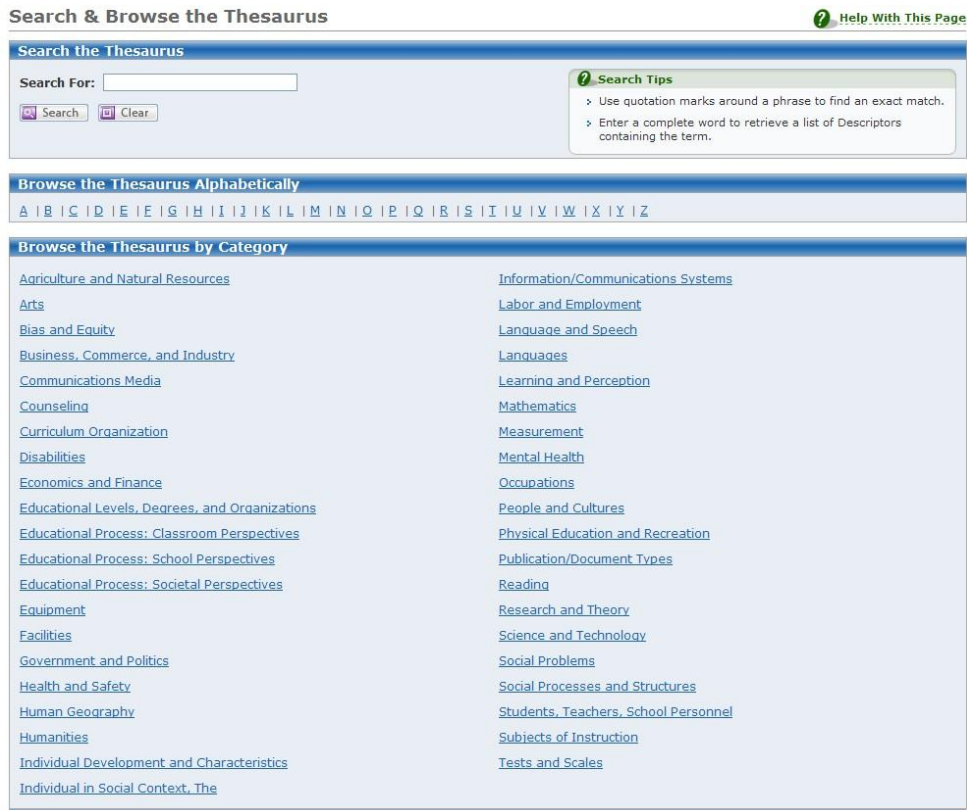
**Figure2. 3: Education and Teaching**

4. At the ERIC website, the researcher is able to search the entire database. ERIC had one of the most powerful searching tools of all the databases; this was the thesaurus, a tool that allows the user to search for key terms associated with the articles available. The researcher can reach this tool by selecting the “Thesaurus” button to the right of the Searching fields.



**Figure 2.4: ERIC Collection**

5. Once at the thesaurus start page, the researcher is able to either search for potential key terms by searching directly, browsing alphabetically, or searching by category. To begin the keyword search, the category “Mathematics” was selected to browse the different keywords associated with mathematics.

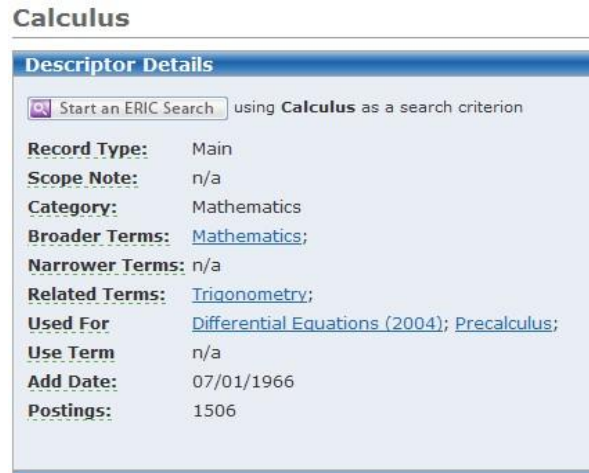


**Figure 2.5: Thesaurus**

6. After selecting a category, you will be presented with all of the descriptors that relate to it. From here the researcher can select which descriptor he or she would like to search by clicking it. For this search, “Calculus” was selected.

7. After selecting the descriptor, the researcher will be presented with the descriptor details. From here, the researcher can see the other descriptors that relate to the originally selected descriptor. Some descriptors will produce only a few others, but many descriptors will produce a great many more descriptors that relate. Often, the researcher will even find more appropriate descriptors for searching

on this page. After checking for more useful descriptors on this page, “Start an ERIC Search” was selected using “Calculus”.



**Calculus**

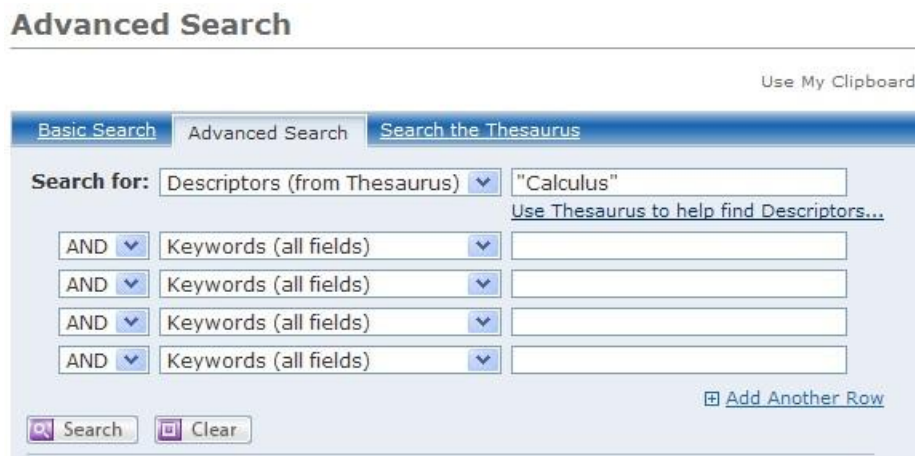
**Descriptor Details**

using **Calculus** as a search criterion

**Record Type:** Main  
**Scope Note:** n/a  
**Category:** Mathematics  
**Broader Terms:** [Mathematics](#);  
**Narrower Terms:** n/a  
**Related Terms:** [Trigonometry](#);  
**Used For:** [Differential Equations \(2004\)](#); [Precalculus](#);  
**Use Term:** n/a  
**Add Date:** 07/01/1966  
**Postings:** 1506

**Figure 2.6: ERIC Search Descriptors**

8. After selecting Calculus, the advanced search page is presented. This page makes it possible to enter as many search parameters as needed. It became clear that only searching for Calculus did not provide very many relevant sources. The search was adjusted.



**Advanced Search**

Use My Clipboard

Basic Search | **Advanced Search** | Search the Thesaurus

**Search for:** Descriptors (from Thesaurus)  [Use Thesaurus to help find Descriptors...](#)

AND

AND

AND

AND

[Add Another Row](#)

**Figure 2.7: Advanced Search**

9. In this search, two search bars were used with descriptors. In the first search bar, "Calculus" OR "College Mathematics" was used, and in the second search bar, "Student Placement" OR "Aptitude Tests" OR "Placement" OR "Screening Tests" was used. This informed the search engine that at least one

descriptor from the first field and at least one descriptor from the second were needed. After the information was entered, the “Search” button was selected.

The screenshot shows an "Advanced Search" interface. At the top, there are three tabs: "Basic Search", "Advanced Search" (which is selected), and "Search the Thesaurus". To the right of the tabs is a link that says "Use My Clipboard". Below the tabs, there are three search rows. The first row is labeled "Search for:" and has a dropdown menu set to "Descriptors (from Thesaurus)" with a downward arrow. The text input field contains the query: "Calculus" OR "College Mathema". Below this input is a link: "Use Thesaurus to help find Descriptors...". The second row starts with an "AND" dropdown, followed by another "Descriptors (from Thesaurus)" dropdown. Its text input field contains: "Student Placement" OR "Aptitud". Below this input is another link: "Use Thesaurus to help find Descriptors...". The third row starts with an "AND" dropdown, followed by a "Keywords (all fields)" dropdown. Its text input field is empty. To the right of this row is a link: "Add Another Row". At the bottom left of the form are two buttons: "Search" and "Clear".

**Figure 2.8: Advanced Search with Descriptors**

10. After pressing the search button, the researcher will see the search results page. On the results page will be different articles that best fit the search parameters. The search parameters used will be highlighted in yellow. Each result is an article and an abstract that briefly describes what the article is about. When finding an article of interest, the researcher will be able to download it directly from ERIC or have to request it from the WPI interlibrary loan system. Since the first article in this search was needed, the interlibrary loan system was used. To do this, “Find in a Library” was selected.



**Search Results**

Sort By: Relevance Use My Clipboard to print, email, export, and save records. 1 items in My Clipboard

Show: 10 | 20 results per page Now showing results **1-10** of **12**. [Next 10 >>](#)

**1. Secondary Options and Post-Secondary Expectations: Standards-Based Mathematics Programs and Student Achievement on College Mathematics Placement Exams (EJ790441)** Added

**Author(s):** [Davis, Jon D.](#); [Shih, Jeffrey C.](#) **Pub Date:** 2007-12-00  
**Source:** School Science and Mathematics, v107 n8 p336-346 Dec 2007 **Pub Type(s):** Journal Articles; Reports - Research  
**Peer-Reviewed:** Yes

**Descriptors:** [College Mathematics](#); [Placement](#); [Mathematics Achievement](#); [Academic Achievement](#); [Multiple Regression Analysis](#); [Calculus](#); [Algebra](#); [Secondary Education](#); [Gender Differences](#); [Statistical Studies](#); [Student Evaluation](#)

**Abstract:** Research on student achievement within the University of Chicago School Mathematics Project (UCSMP) and Core-Plus Mathematics Project (CPMP) at the secondary level is beginning to accumulate, however, much less is known about how prepared these students are for post-secondary education. Therefore this study involving students within one tracked school district used multiple linear regression to [e Show Full Abstract](#)

**Full-Text Availability Options:**  
[Help Finding Full Text](#) | [Find in a Library](#) | [Publisher's Web Site](#)

---

**2. Comparative Validity of the Descriptive Tests of Mathematical Skills (DTMS) and SAT-Mathematics (SAT-M) for Predicting Performance in Freshman College Mathematics Courses: Prefatory Report (ED502316)** Add

**Author(s):** [McLoughlin, M. Padraig M. M.](#); [Bluford, Dontrell A.](#) **Pub Date:** 2004-02-24  
**Source:** Online Submission, Paper presented at the Annual Meeting of the Eastern Educational Research Association (Clearwater, FL, February 12, 2004) **Pub Type(s):** Reports - Research; Speeches/Meeting Papers  
**Peer-Reviewed:** N/A

**Descriptors:** [African American Institutions](#); [Prediction](#); [Test Validity](#); [Calculus](#); [Algebra](#); [College Admission](#); [College Entrance Examinations](#); [Student Placement](#); [Black Colleges](#); [Regression \(Statistics\)](#); [Remedial Mathematics](#); [Mathematics Skills](#); [Scores](#); [College Mathematics](#); [Correlation](#); [Psychometrics](#)

**Abstract:** This study investigated the predictive validity of the Descriptive Tests of Mathematical Skills (DTMS) and the SAT-Mathematics (SAT-M) tests as placement tools for entering students in a small, liberal arts, historically black institution (HBI) using regression analysis. The placement schema is four-tiered: for a remedial algebra course, college algebra course, pre-calculus course, and a first ca [Show Full Abstract](#)

**Full-Text Availability Options:**  
 ERIC Full Text (194K)

Figure 2.9: Search Results

11. From this page, select “Interlibrary Loan Request”, enter in the appropriate library account information, and press request.

**FullTextFinder**

**Title:** Secondary Options and Post-Secondary Expectations: Standards-B  
**Source:** Davis, Jon. School Science And Mathematics Volume: 107 Issue: 8 (2007-01-01) ISSN: 0036-6

**Resources Located for this Citation**

**Get the Full Text**  
 Find this article in full text from Gale 1999-01-01 to present

**Check the Journals List**  
 See if WPI has this journal.

**Interlibrary Loan Request**  
[Interlibrary Loan Request](#)

**Search the Web**  
 Search for more info on this author in Google Scholar  
 Search for more info on this title in Google Scholar  
 Search Google Scholar for more info on this article title and journal title

**RefWorks**  
 Add the citation for this title to RefWorks.

**Check the WPI Library Catalog**  
 Search the library catalog for this item's title

Figure 2.10: Full Text Finder

(Education Resources Information Center) (WPI George C. Gordon Library, 2009)

## 2.3 Comparison of the WPI Placement Test with Other Schools' Exams

The current WPI online placement tests are different from the placement tests found at several other colleges. The first difference is that all of the other tests found used a multiple choices format, as opposed to WPI's open response format. Second, the overall level of difficulty appears to be higher for free response placement tests. Finally, the mathematical topics of the tests varied greatly among the different colleges' calculus placement tests.

Many other colleges use a multiple choice format for calculus placement tests. This format allows for automatic grading, but it also gives students extra information about the problem by providing them with a list of possible answers. These tests can be completed much more quickly than an open response test, if only because guessing is always an option. In addition, students employ many test-taking strategies while taking a multiple choice test, usually designed to find the correct answer with a minimal understanding of the actual mathematics. These strategies include: eliminating answers, working backwards from provided answers, and simply guessing (Figure 2.11). Some multiple choice exams include "none of the above" answers to help combat these testing strategies (Figure 2.12).

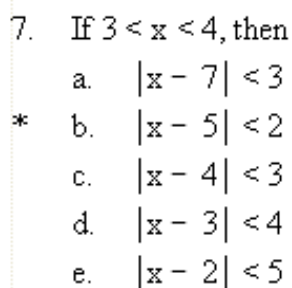
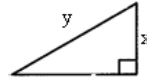
- 
7. If  $3 < x < 4$ , then
- a.  $|x - 7| < 3$
  - \* b.  $|x - 5| < 2$
  - c.  $|x - 4| < 3$
  - d.  $|x - 3| < 4$
  - e.  $|x - 2| < 5$

Figure 2.11: Multiple Choice Question

4. What is the area of the triangle shown below?



a.  $\frac{xy}{2}$

b.  $\frac{x(y-x)^2}{2}$

c.  $\frac{\sqrt{y^2-x^2}}{2x}$

\* d.  $\frac{x\sqrt{y^2-x^2}}{2}$

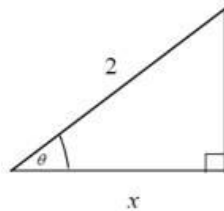
e. None of these

**Figure 2.12: Multiple Choice Question with “None of these” as a possible answer**

(Testing and Evaluation Services)

The WPI placement test, along with other free response placement exams, incorporates a higher level of difficulty than other forms of testing. This is a result of the student having to recall information from their own knowledge base and not multiple choice testing strategies. Unfortunately, these tests generally take longer and require more time to grade. Figure 2.13 is a trigonometry question from Temple University and Figure 2.14 is a trigonometry question from WPI’s Placement Test. There is a clear distinction between the difficulties of these two problems, largely due to the question format.

34. Find  $\cos \theta$ ,  $\sin \theta$ ,  $\tan \theta$ ,  $\cot \theta$ ,  $\sec \theta$  and  $\csc \theta$  where  $\theta$  is the angle shown in the figure below.



**Figure 2.13: Trigonometry Question**

((Temple University Russell Conwell Educational Services Center, 2006))

- (1 pt) If a right triangle has a hypotenuse of length 13 and one leg of length 5 and  $\theta$  is the angle between that leg and the hypotenuse, please
- a) Determine the length of the other leg.
  - b) Determine  $\cos(\theta)$
  - c) Determine  $\sin(\theta)$

**Figure 2.14: WPI Trigonometry Question**

(WPI WeBWorK, 2000-2006))

The mathematical subjects included in tests vary between schools. Some tests only include algebra and trigonometry while others, such as WPI, include algebra, geometry, trigonometry, and functions. Differences between tests are potentially due to the different curriculums offered at each school. WPI has an easier task because the only introductory mathematics topic considered is calculus.

## 2.4 Special Features of the WPI Placement Test

The implementation of the WeBWork module currently sets the WPI Calculus Placement Test apart from other placement tests. This system has been used at many universities around the country for algebra, calculus, statistics, and physics courses. It has also been implemented in the placement tests in 2007 and 2008. The test has two important and unique features.

The first feature is the open-response answer format. As discussed above, multiple choice exams can be good indicators of a student's test taking ability, but may not show the student's aptitude in the subject material. By using techniques discussed in the previous section, the incoming freshman may score higher on the placement test than actually reflects his or her math abilities. The goal of the placement test is to match the student into the calculus course that he or she is ready for, not to see how intelligent the incoming class is or how well they can perform on an entrance test.

The second WeBWork feature is randomly generated variations of questions. This is a more useful tool when using WeBWork for graded class assignments where cheating would be discouraged, but it will also have an appealing distribution in the questions. Coefficients in a problem can be selected randomly for each student, so that the basic concept tested is the same but the exact function or numerical answer is unique to the student.

The test is taken online and graded immediately. In fact, students see immediately whether they have given a correct answer for a problem and are given three attempts to enter a correct answer. Each day, the grades are downloaded and recommendations are sent to the students who have completed a test, providing them with placement test scores and recommended first math course. Though the process is automated, these emails are sent through the account of a faculty member. In this way, incoming students already have a personal contact at WPI and are free to ask questions and make comments.

There is an important additional feature in WeBWork. Each problem has an “Email Instructor” button that students can use to send questions to WPI. Many students use this button to get help with a problem, to clarify a question, or simply to ask when their test will be evaluated. Once again, the gain is getting the student connected to WPI.

### 3. Project Goals

One goal of this project is to compare the results of the second delivery of the Calculus Placement Tests to the results obtained in the first year. A second goal is to improve the existing Calculus Placement Test by simplifying the test format and improve the accuracy of the recommendations while maintaining the personal contact between faculty and the student. With these major goals in mind, the focus was placed on the following specific goals.

First, a basic statistical analysis is performed on the new data from the summer of 2008. The analysis uses methods identical to those used in the previous year. In addition, only a small portion of the statistical analysis will be repeated.

Second, the three Placement Tests will be combined into a single 34 question test. This will solve several problems with the current Calculus Placement Tests. These problems will be discussed later. (The change may also introduce new, unforeseen problems. These should be the study of a future IQP.)

Finally, the recommendation system can be improved by considering the student's Advanced Placement credit and his or her intended major at WPI. With this information, the recommendation can be tailored for each student.

To achieve these goals, further changes must be made to the macro that analyze student scores and sends the recommendation. These changes should reduce the confusion in the recommendation process, and maintain the personal contact between the incoming students and the faculty member managing the Calculus Placement system.

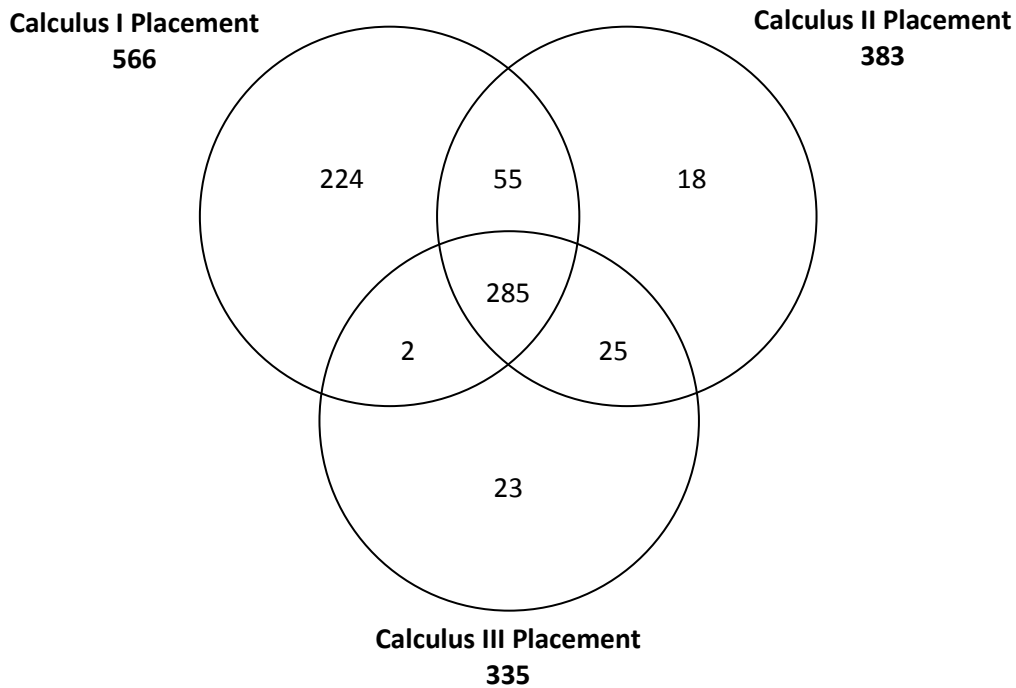
#### **4. Analysis of Scores on the Calculus Placement Tests:**

The original Calculus Placement Tests contained questions that test prerequisite material for the given class. The Calculus I Test confirms a student's background in Functions, Geometry, Algebra, and Trigonometry. The Calculus II Test confirms a student's knowledge of Calculus I (differentiation and applications). The Calculus III Test confirms a student's knowledge of Calculus II (integration and applications). Students have the option to take any combination of these three tests (or none at all). The primary goal of this section is to verify the results for 2007 with the data from 2008. All data from the 2007 Placement tests is taken from the 2008 IQP report (Leary & Smaltz, 2008) analyzing those test results.

Students were not required to finish any test. Students are, however, encouraged to complete tests by the Office of Academic Advising. In 2008, 632 students in the class of 2012 completed at least one test, which is 70% of the 907 students in the first-year class. In 2007, 532 of 805 students, 66%, completed at least one test. The slight improvement is encouraging.

Many of the 632 students completed more than one test; some even completed all three. The diagram in Figure 4.1 shows how many students completed each possible combination of Placement Tests.





**Figure 4.1: Number of students completing placement tests in 2008.**

It is interesting that only one number in the above Diagram has changed significantly in the last year. The number of students completing all three tests increased from 199 students to 285 students. All other numbers changed very little with a maximum increase of 13 in any single area. Table 4.1 below compares the data.

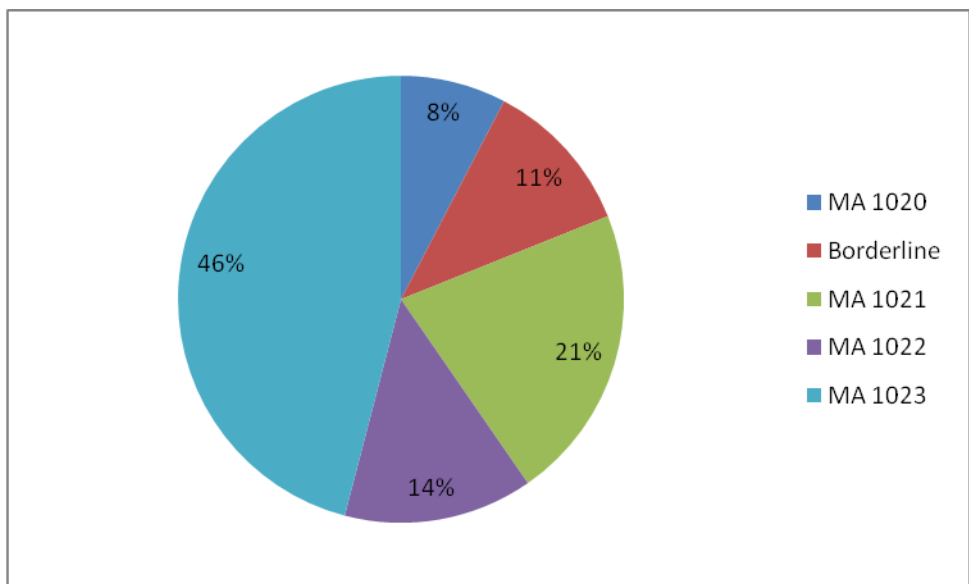
**Table 4.1: The number of students taking each combination of tests in 2007 and 2008.**

	Calculus I	Calculus II	Calculus III	Calculus I+II	Calculus I+III	Calculus II+III	Calculus I+II+III
<b>2008</b>	224	18	23	55	2	25	285
<b>2007</b>	223	9	22	58	9	12	199
<b>Difference</b>	1	9	1	3	-7	13	86

There were 224 students who completed only the Calculus I Test and 27 of these students received a perfect score but chose not to complete any other test. There were an additional 81 students who also passed with a score of 13 or higher and chose not to complete any other test. While these

numbers have increased slightly from 2007, the proportion of students in this group has dropped from 20% to 17% in 2008. This, too, can be viewed as a success. However, this is still a large number of students who could have received a poor recommendation because they did not attempt further tests.

Figure 4.2 below shows the final recommendations given in 2008. Note that the recommendation is often altered by faculty when students provide extra information which may result in a different recommendation (e.g. a student receives AP credit for Calculus I, II, and III if they get a 4 or 5 on the BC exam). For this reason, all recommendations involving Calculus IV are grouped together with Calculus III for this study. Also note that the instructors edit recommendations based on the Placement test results.



**Figure 4.2: Recommendations in 2008.**

This information closely resembles that of 2007. No area increased by more than 3%, with MA1022 (Calculus II) showing the greatest change of 3%. However, the only area to decrease, MA 1021 or Calculus I, dropped by 8%. This is a much more significant change. This may be a natural effect of the smaller proportion of students taking only the Calculus I Test. There should also be a certain

amount of variation between the two data sets. This may account for the significant change, but other circumstances may have had an effect as well. Table 4.2 below compares the percentages of recommendations in 2007 and 2008.

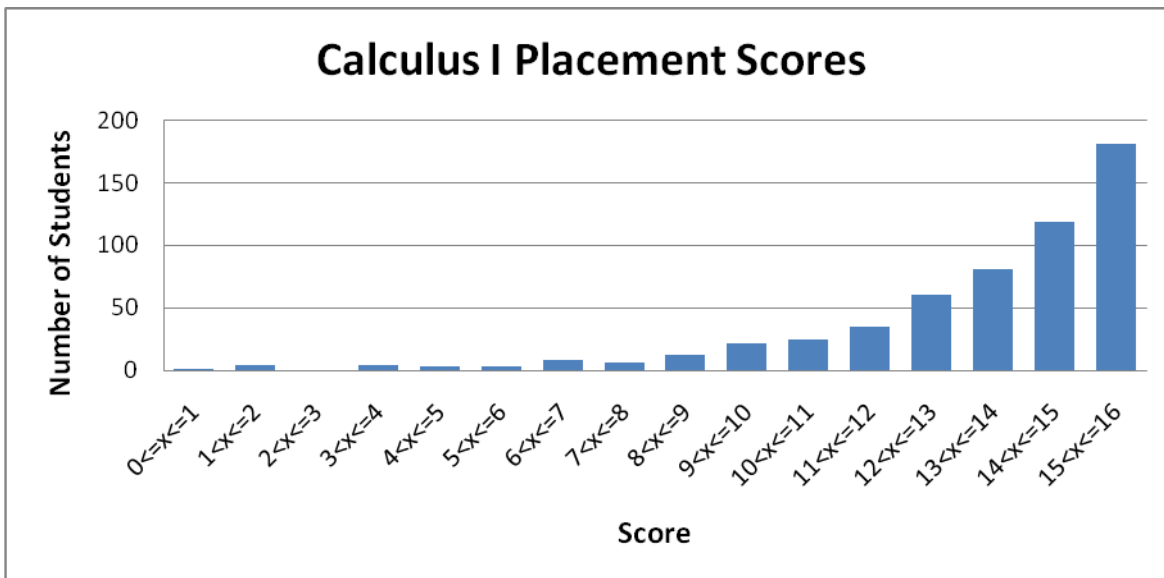
**Table 4.2: The distribution of recommendations in 2007 and 2008.**

	<b>MA1020</b>	<b>Borderline</b>	<b>MA1021</b>	<b>MA1022</b>	<b>MA1023</b>
<b>2008</b>	8%	11%	21%	14%	46%
<b>2007</b>	6%	10%	29%	11%	44%
<b>Difference</b>	2%	1%	-8%	3%	2%

Figure 4.3 below shows the distribution of scores for the Calculus I Placement Test. Clearly, the more common scores are very high. 21% of Calculus I Test takers achieved a perfect score. In comparison, 27% of Calculus I Test takers achieved a perfect score in 2007.<sup>3</sup> Usually, an approximately normal distribution seems more appropriate. However, this heavily left skewed distribution is desired for the Calculus I Placement Test. It is expected that most students will do well on the test. These students are comfortable with the prerequisite material for Calculus I. Students not comfortable with pre-calculus material are not admitted to WPI.

---

1. Note that the statistic 27% of students received a perfect score on the Calculus I Placement Test in 2007 was inaccurately reported in the 2008 IQP report on the Calculus Placement Tests. The 2008 report mistakenly reports scores greater than 15 of 16 points as a perfect score. Thus 27% is likely a high estimate. Note that the 2008 scores are accurately presented.



**Figure 4.3: Distribution of scores on the Calculus I Placement Test in 2008**

The tests were designed to give individual scores for Algebra, Functions, Geometry, and Trigonometry. Each of these subjects has four questions in the test to check that a student is proficient in that subject. As a result, a perfect score of 4 is possible for each of the four subjects, totaling 16 points. A student that scores a 2 or less in any subject is advised to review that particular subject in the Jump Start Modules, an internet-based review. Note that it is possible to pass the Calculus I Test (13 points or higher) and still be advised to take Jump Start.

Figure 4.4 below shows the distribution score for individual subjects in the Calculus I Placement Test. Note that 66 students took the Calculus II and/or Calculus III Tests without having taken the Calculus I Test. The students were placed in the “No Score” category. For each subject, the distribution of scores is heavily left-skewed and a perfect score of 4 is most common. The 2007 distribution of Calculus I sub-scores is nearly identical, excluding a change in scale for “Number of Students” and the proportional change of Geometry and Functions for scores in the range of (2,3]. The change of scale is required for such a comparison to accommodate the larger number of students taking the test. The

small changes can be considered random changes between 2007 and 2008. To further justify this idea, the proportion of students achieving a perfect score in Algebra increased by less than 1%, from 67.1% in 2007 to 68% in 2008<sup>4</sup>. Students were least successful in answering Geometry and Trigonometry questions. Trigonometry was the subject for which the greatest proportion of students (16.4% of those taking the test) received a Jumpstart recommendation. With less than half (43.6%) of test takers earning a perfect score in Trigonometry, this subject exhibited the smallest percentage of perfect scores. In 2007, 20.3% of test takers received a Jumpstart recommendation for Geometry (the subject for which the greatest proportion of students received a Jumpstart recommendation). Trigonometry was also the subject that exhibited the smallest percentage (42.1%) of perfect scores. As is discussed in the 2008 report, question 7 of the Calculus I Placement Test is particularly difficult, involving the Law of Sines.

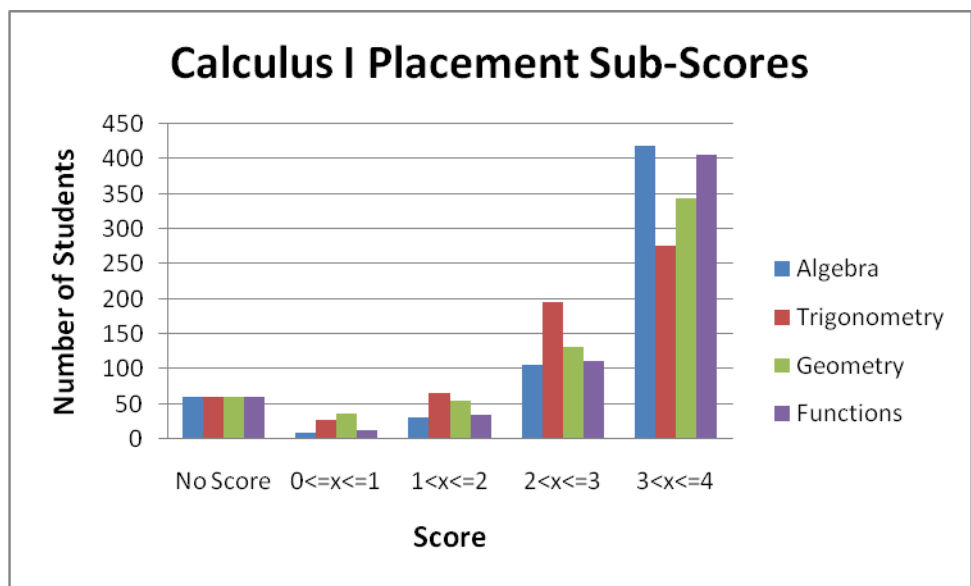


Figure 4.4: Distribution of scores on individual subject areas of the Calculus I Placement Test

<sup>4</sup> Note that the statistic 67.1% of students received a perfect score in Algebra on the Calculus I Placement Test in 2007 was inaccurately reported in the 2008 IQP report on the Calculus Placement Tests. The 2008 report mistakenly reports scores greater than 3 of 4 points as a perfect score. Thus 67.1% is likely a high estimate. Note that the 2008 scores are accurately presented.

Due to what seems to be significant differences in the various subject areas, it is useful to determine if these differences are statistically significant. A two-sampled t-test for each possible two subject combination is appropriate. The null hypothesis,  $H_0$ , is that the differences in the distribution of scores for the two different subjects were due to chance. The p-values for these tests are given in Table 4.3 below.

**Table 4.3: p-values for two-sample t-tests, examining differences between individual sections of the Calculus I Placement Tests in 2008**

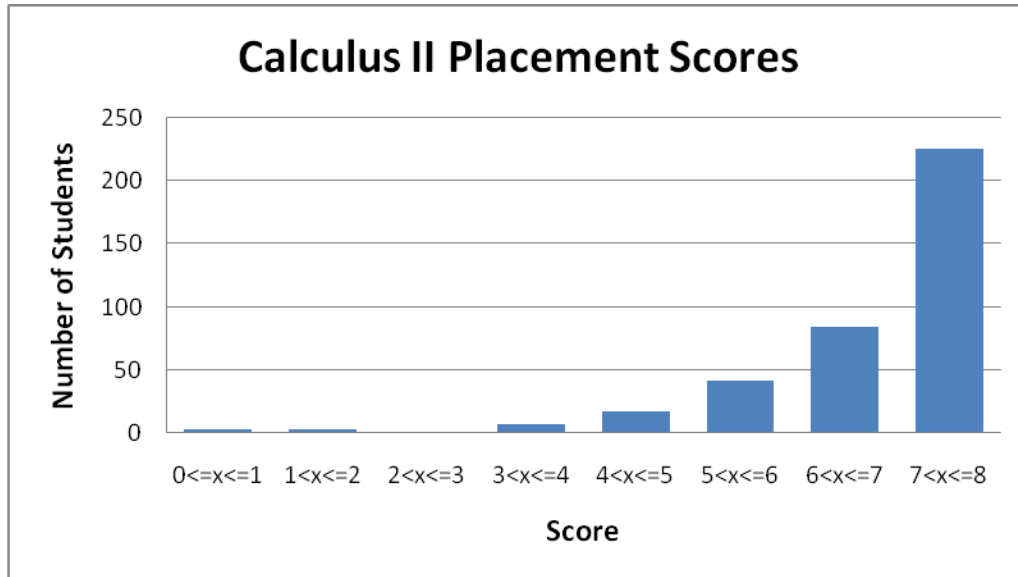
	<b>Algebra</b>	<b>Trigonometry</b>	<b>Geometry</b>	<b>Functions</b>
<b>Algebra</b>		$4.9484 \cdot 10^{-26}$	$2.6042 \cdot 10^{-15}$	$4.9320 \cdot 10^{-2}$
<b>Trigonometry</b>			$5.1847 \cdot 10^{-4}$	$9.95576 \cdot 10^{-25}$
<b>Geometry</b>				$8.14046 \cdot 10^{-14}$
<b>Functions</b>				

The results of the t-tests validate the supposition that student performance on Trigonometry and Geometry was significantly different from that of Algebra and Functions. The four relevant subject combinations, Geometry-Algebra, Geometry-Functions, Trigonometry-Algebra, and Trigonometry-Functions, certainly have small enough p-values to reject the null hypothesis. The Algebra-Functions combination has too high a p-value ( $p = 0.049319912$ ) to reject the null hypothesis. Trigonometry-Geometry, however, also has a small enough p-value ( $p = 0.00051847$ ) to reject the null hypothesis at the 1% significance level. While the statement that Trigonometry and Geometry are significantly different is valid, this statement is not as strong as the others. This last result is the only result that contradicts the 2007 report, which could not reject the same null hypothesis. The general results, however, remain the same: Trigonometry and Geometry are significantly more difficult than Algebra and Functions. Table 4.4 shows the p-values for the t-tests in the 2008 report.

**Table 4.4: p-values for two-sample t-tests, examining differences between individual sections of the Calculus I Placement Tests in 2007**

	Algebra	Trigonometry	Geometry	Functions
Algebra		$5.2817 \cdot 10^{-30}$	$6.0709 \cdot 10^{-22}$	$1.4757 \cdot 10^{-02}$
Trigonometry			$2.1717 \cdot 10^{-01}$	$3.4478 \cdot 10^{-21}$
Geometry				$2.1955 \cdot 10^{-17}$
Functions				

Figure 4.5 below shows the distribution of scores for students completing the Calculus II Placement Test. 383 students completed this test in 2008, compared to 278 students in 2007. As in 2007, this data is heavily left-skewed. These two data sets are extremely similar in certain aspects. The mode in both years was a perfect score of 8. The proportion of students passing this test (a score of 6 or higher) is 90.1% (345 students) in 2008 and 90.2% (251 students) in 2007.



**Figure 4.5: Distribution of scores on Calculus II Placement Test**

Figure 4.6 below shows the distribution of scores for students completing the Calculus III Placement Test. In 2008, 335 students took this test, an increase of 93 students from 2007. Of those 335 students, 288 were recommended for Calculus III, a drop of 7% from 2007.

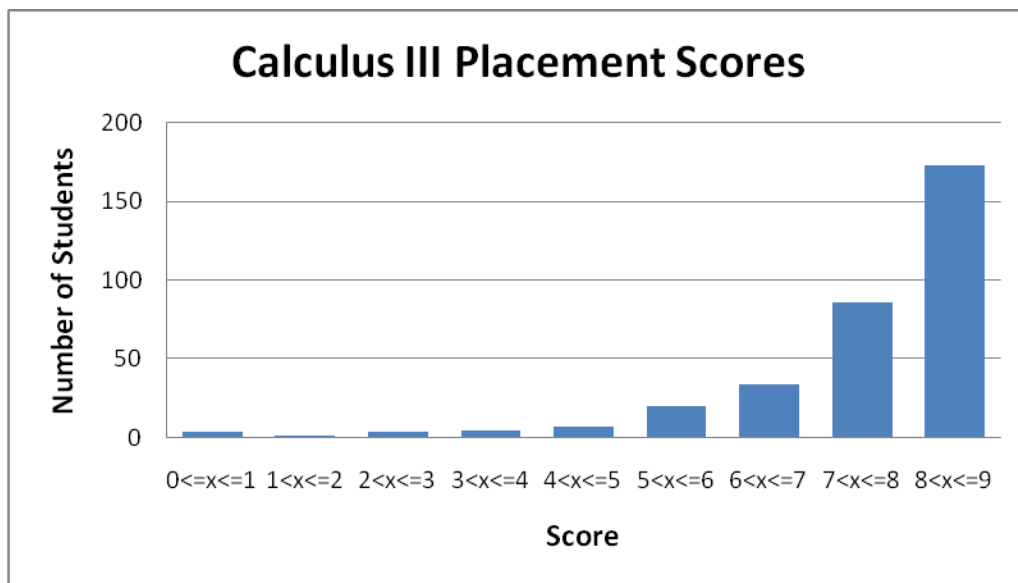


Figure 4.6: Distribution of scores on Calculus III Placement Test in 2008,

The number of students passing the Calculus II and III Tests may be quite high. However, significantly fewer students took these tests and most of those who completed these two tests also took the Calculus I Placement Test. Many students may have felt less comfortable taking these tests for many different reasons. With this in mind, it is more likely that those taking the Calculus II and III Tests would achieve higher scores.

While the number of students passing all three tests may still seem very high, the tests are checking prerequisite material for classes. Therefore, it is desirable to have a high number of passing results. As a result, it is more significant to determine how these tests predict success in courses at WPI.

The primary goal of this section was and is to determine if previously generated analyses of the Calculus Placement Tests are accurate. All data and analysis of the 2007 test results are taken from the

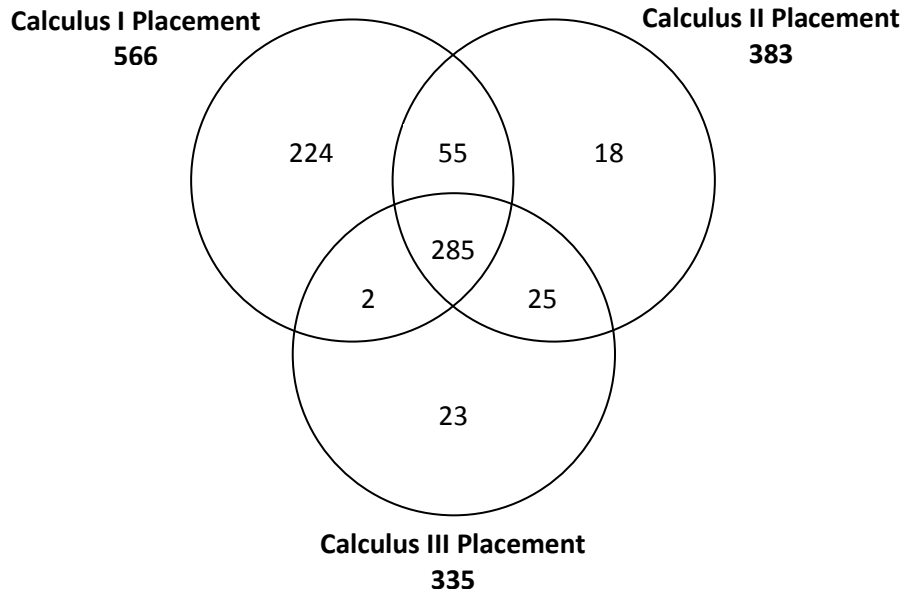


2008 IQP report for comparison. The results of this study show that the 2008 report has generated a very accurate analysis. There are very few disparities between the two data sets. The disparities that do exist are minor and do not show any significant changes (Leary & Smaltz, 2008).

## 5. Merging the Three Tests

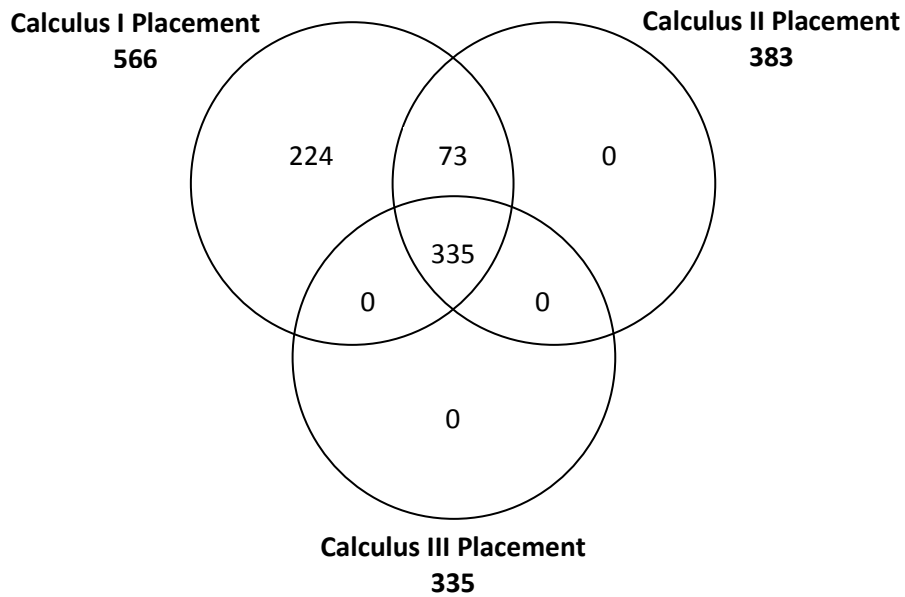
The current format of the placement tests creates certain situations which lead to confusing recommendations. For example, a student received one (automatic) recommendation at the completion of the Calculus I Placement Test. If they went on to take the Calculus II and Calculus III tests and did well, they would receive a new email with a different recommendation. Another problem would arise when a student completed only one test, for example the Calculus II Placement test. The faculty at WPI would not know if they planned to go on and take the Calculus III test or should receive a recommendation immediately. This confusion was the primary motivation combining the three separate tests into a single test.

Another important problem is the unfortunate case when a student has a good understanding of certain elements of Calculus, but not the pre-requisite material. There have been many cases in the past when the incoming student is very well-educated in differentiation and integration but has limited knowledge of geometry, functions, algebra, and/or trigonometry. Therefore, it is important for the Digital Advising Algorithm (on page 35) to have information on the incoming freshman's math skills as tested in the Calculus I Placement Test. The following is a Venn diagram displaying which placement tests were taken in 2007 by the students.



**Figure 5.1: Number of students completing each possible combination of the three Placement Tests**

As figure 5.1 illustrates, there were students who may have only taken the more advanced tests without having attempted the lower level placement test or tests. Figure 5.2 below is a Venn diagram that would be preferred to the figure above.



**Figure 5.2: Ideal number of students completing each possible combination of the three Placement Tests**

Figure 5.2 is derived from the same data as Figure 5.1. The difference between them is that certain regions were combined together. Any student who took the Calculus III Placement Test was combined into the center region, which denotes having taken all three tests. Those who had only taken the Calculus II Placement Test were combined in the region that denotes having taken the Calculus I and II Tests. Those who had only taken the Calculus I test were left alone. To clarify, Figure 5.2 was not actual data, but an ideal situation.

The original tests included a 16 question Calculus I Placement Test covering pre-calculus material. There are four questions for each of the following subject areas: Algebra, Geometry, Trigonometry, and Functions. The Calculus II Test included 9 questions covering differentiation and the 8 question Calculus III Placement Test evaluated the student's integration skills.

There are two proposed courses of action to correct the problems discussed. The first solution maintains the existence of three tests. The Calculus I Test would cover Algebra, Geometry, Functions, and Trigonometry. The Calculus II Test would cover Algebra, Geometry, Functions, Trigonometry, and Differentiation. The Calculus III Test would cover Algebra, Geometry, Functions, Trigonometry, Differentiation, and Integration. Therefore, there would be a 16 question Calculus I Placement Test, a 25 question Calculus I and II Placement Test combination, and a 33 question Calculus I, II, and III Placement Test. This solution maintains the stability of having three tests. The incoming student can choose what he or she feels comfortable with. This seems to solve the problem of receiving multiple recommendations. It also ensures that each student must test his or her knowledge of pre-requisite mathematics material.

A potential flaw in this solution is the possibility of a student choosing one test, completing that test, and later deciding to take a different test instead. The confusion of multiple recommendations is still a possibility. Similarly, if a student is unsure of which test to take and decides to take a placement

test that they are overqualified for, then the student may receive a good score and may either be recommended to take a course that will not pose a challenge or they will want to take the next placement test. The inconvenience is that they will have to retake the test that they just finished in order to reach the questions that they have not yet seen.

The second solution is to combine all of the tests into a single 33 question Calculus placement test. This solution requires that every student take the same test and skip problems if he or she is not familiar with the material. There is no chance that a student would receive multiple recommendations. This also eliminates the confusion regarding which test to take. This solution is not without its drawbacks. The test will be considerably longer. The student is going to see that the test is 33 questions long and may become discouraged. Students may also be discouraged if they come across a section that is difficult. For many students, the most difficult questions are the earliest questions (the pre-calculus material) and so they may not continue to work the problems which are actually the easiest for them.

It was decided to implement the second solution and create a single Calculus Placement Test using WeBWork. In order to accommodate the new test format of the WPI Calculus Placement Test, the Mail Merge macro had to be modified. The old Mail Merge macro only created email responses for the Calculus I test and it merely stored the data on the students' results for the Calculus II and III Placement Tests. This was inefficient because it required the Admin to handwrite emails for students who took the Calculus II and III Placement Test. The new Mail Merge macro records data from Calculus I, II, and III at the same time and outputs a single automated email, providing a recommendation for the student based on all three test scores.

The macro had to be modified in three key areas to accomplish this. First, it needed to be able to split up and store the test scores from the three tests. Second, it had to generate a single

recommendation using the stored data. Third, it had to include wording so the email could automatically give any recommendation.

In order to split up and store the three test scores from a single excel spreadsheet, we first devised a method to read through the data separately. It then stored the data separately using newly declared variables. An added difficulty was getting the stored data to correctly output to the excel sheet, but this was solved by modifying the print function currently being used.

To generate only one recommendation for each student, it was determined which single recommendation to give to each combination of scores. Currently, the student is recommended for the highest level class that they pass the corresponding test for, and is recommended to take Jumpstart if they were weak in any of the Calculus I areas of Algebra, Geometry, Functions, or Trigonometry. This was determined to be the best method because a student who can pass one of the Calculus placement exams should be able to pass its corresponding class. Also, it is likely that a student may become out of practice with material they learned earlier, and therefore shouldn't be put in a class they may be over-prepared for. In order to accommodate for this, a system where the student would only receive their highest recommendation was put in place using an if-then statement.

With multiple types of recommendations being covered by a single Mail Merge macro, wording had to be carefully selected and included so that the recommendation would always be clear. The excel spreadsheet and email was heavily edited for this purpose. The new wording for the Calculus recommendation is much more concise, providing test scores and the recommendation in bullet form, with a brief introduction and some important information to close.

## 6. Digital Advising Algorithm

### 6.1 Recursive Algorithm for Determining Recommended Course Background

Every course at WPI has recommended background associated with it. The recommended background is a course or courses that should be taken and passed before attempting the course in question. For example, Calculus II or an equivalent is the recommended background for Calculus III and Calculus I or an equivalent is the recommended background for Calculus II. Many majors require mathematics courses, but this information can be difficult to find. Therefore, a solution was devised.

A Perl script was written in order to facilitate the gathering of information on which math classes are needed for each major. As an input, the script takes in a course number and then outputs all the courses that are recommended for that course, along with all other recommended courses. This allows the script to find recommended math courses hidden under many layers of recommended background courses. Therefore, all that needs to be input into the program are the courses that are required for the major.

The program accomplishes this process by going through the online course catalog and parsing out the recommended courses for each course. It accomplishes this by identifying each individual course segment on the page, and then pulls out the course name and its requirements. A course can be broken down into its title, description, and recommended background, all of which make up a course segment. This information is relatively easy to parse out due to the way in which the html is formatted. The course description can then be removed, leaving only the recommended background and the course title. The recommended background can then be parsed to be only course numbers, while the course number can be pulled out of the segment title.

The script subsequently stores this information in a hash table. The key value for the hash table is a course number, which returns an array of course numbers for the recommended background courses. The script then iterates through the array, using the values in the array as keys to the hash table, which pulls out more recommended background courses. The script stops iterating when the current course has no recommended background courses. It does this by simply using the following recursive algorithm:

```
String Recursive_Class_Search(HashTable courseHash,String courseName,String courseRequirement)
{
    Array courses = courseHash{courseName};
    Foreach(String recommended_course in courses)
    {
        courseRequirement = courseRequirement + recommended_course + " ";
        courseRequirement = Recursive_Class_Search(courseHash,
                                                    recommended_course, courseRequirements);
    }
    return courseRequirements;
}
```

This algorithm returns a string containing the course numbers for all the recommended background courses. After returning the script, it would then print out the recommended background and prompt the user for more input.



## 6.2 An Analysis of Recommended Course Groupings

The recommendation process is to be modified for individual majors. The planned method was to group majors together based on the mathematics courses they require and provide recommendations to each group. First a spreadsheet was created to compile all majors at WPI and the mathematics courses required for each. The information in this spreadsheet came from the WPI 2008 – 2009 course catalogue. Many entries from the course catalogue were vague about the possible requirement of Calculus III, MA1023, and Calculus IV, MA1024. In these cases, the catalog stated that Differential and Integral Calculus were mandatory. This can be interpreted as Calculus I and II or as all four courses in the Calculus sequence. Figure 6.1 below shows the spreadsheet. Then each major’s mathematics requirements were compared and majors were grouped accordingly. The intention was to provide the same recommendation for “A” and “B” term for any major in a given group.

	Total Units(max)	Calculus I	Calculus II	Calculus III	Calculus IV	Ordinary Differential Equations	Discrete Mathematics	Statistics I	Statistics II	Probability for Applications	Probability	Linear Algebra
Aerospace Engineering	5/3	X	X	X	X	X						
Biology and Biotechnology	????											
Biomedical Engineering	2/1	X	X	X	X	X		X				
Chemical Engineering	2/1	X	X			X						
Biochemistry	4/3	X	X	X	X							
Chemistry	5/3	X	X									
Civil Engineering	3/1	X	X			X		X	X	X		
Computer Science	7/3							1	1	2	2	
Electrical and Computer Engineering	3/1	X	X	X	X	X	X	1		1		
Humanities and Arts	2/3											
IMGD	1/3	1	1	1	1	1	1	1	1	1	1	
Industrial Engineering	7/3	X	X	X	X	X		X		X		
Professional Writing	????											
International Studies	1/3											
Liberal Arts and Engineering	5/3	X	X	X	X			1		1		
Management	4/3	X	X					X	X			
Management Engineering	4/3	X	X					X	X			
Management Information Systems	4/3	X	X					X	X			
Manufacturing Engineering	8/3	X	X	X	X	X						
Mathematical Sciences	7/1											
Mechanical Engineering	2/1	X	X	X	X	X						X
Physics	3/1											
Robotics Engineering	7/3	X	X	X	X	X	X	1		1		
Economic Science	2/1	X	X	X	X			X	X			
Environmental Policy	5/3	X	X	X	X			X				
Psychological Science	4/3	X	X					X	X			
Society/Technology Policy	1/1											
System Dynamics	2/1	X	X	X	X	X		X				

Figure 6.1: Database of Majors and associated required mathematics courses

After compiling the database, the next course of action was to arrange them into exact matches. Some majors simply require a strong math background. This includes Mathematical Sciences and Physics, which were grouped together. Likewise, all classes with a very low number of required math

credits were also put in the same group. Figure 6.2 shows the 6 groups created. Figure 6.3 displays entirely unique cases.

Group Number	1	2	3	4	5	6
Listed	2	2	3	3	2	5
Majors	Biochemistry	ECE	Management	Aerospace	Mathematical Sciences	Biology
	Chemistry	RBE	MGE	Chemical Engineering	Physics	IMGD
			MIS	Manufacturing Engineering		Humanities and Arts
						Professional Writing
						International Studies

**Figure 6.2: Major groups according to required mathematics courses**

Unique
11
Biomedical Engineering
Civil Engineering
Computer Science
Industrial Engineering
Mechanical Engineering
Liberal Arts and Engineering
Economic Science
Environmental Policy
Psychological Science
Society/Technology Policy
System Dynamics

**Figure 6.3: Unique Cases in Perfect Matching**

The benefit to grouping majors seems to be minimal. Most groups only contain two or three majors. There are eleven majors that could not be matched with any others for math requirements. This shows that grouping classes together by exact matches is impractical.

A salvage attempt was made to group majors by recommendations for A and B term. The results of this method are shown below. Figure 6.4 displays groups based on common recommendations. Figure 6.5 displays the unique cases.

Groups	1	2	3
Listed	18	10	3
Majors	Aerospace Engineering	Management	IMGD Art
	Biochemistry	MGE BME	IMGD Technical
	Chemical Engineering	MGE CEE	International Studies
	Chemistry	MGE ECE	
	Civil Engineering	MGE MAE	
	Computer Science	MGE ME	
	ECE	MGE OM	
	Industrial Engineering	MIS	
	Liberal Arts and Engineering	Psychological Science	
	Manufacturing Engineering		
	Mathematical Sciences		
	Mechanical Engineering		
	Physics		
	Robotics Engineering		
	Economic Science		
	Environmental Policy		
	System Dynamics		

**Figure 6.4: Recommendation Based Grouping**

Unique
Biology and Biotechnology
Humanities and Arts
Professional Writing
Society/Technology Policy

**Figure 6.5: Unique Cases**

This approach seems to work better. There exists a concern with this method that once the recommendation exceeds Calculus III in “A” term, the recommendation becomes unreliable. The problem becomes more complicated when considering Advanced Placement credit that students can obtain during high school.

These problems quickly become too difficult to handle. Grouping majors together in this fashion is too complicated and inconvenient. The problem grows when handling unique cases. It was decided that an easier, more reasonable system is to treat every major as a unique case, storing its required information in a database.

### 6.3 Placement Test Algorithm Explained through Flowcharts

A series of flowcharts was created to assist with the programming of the Digital Advising Algorithm. The flowcharts were designed to be in modular form. This design makes the flowcharts easier to follow, more efficient, and more manageable. The first flowchart calls for the student's math background, specifically including AP classes. The second flowchart is the original recommendation breakdown. The final flowchart in the making is the Jumpstart Analysis module.

The first flowchart, on page 51, receives input from the student detailing any AP Calculus Exams taken and the student's intended major. Up to four majors may be selected. There will be one final recommendation email. It will not only include the recommendation, but will also include "remaining math classes recommended for selected majors." This will be a list of all the classes that the major requires. To make this clearer to the new student, any class that they have already received AP credit for will be excluded from the remaining classes list.

The second flowchart, on page 52, is the module of fast recommendation. It breaks each test into pass/fail margins. It also groups the questions into topics. Every topic will also be given a pass/fail margin. If any topic is failed, then an attachment on the output email will be included recommending that the student review that particular topic and that they may use the "Jumpstart" program in order to prepare. There are four ideal cases in this stage. Each consists of passing up to a certain test and failing the remaining. Two examples of this are passing all of them, or passing Calculus I and II but not the Calculus III Test. These will make fast recommendations possible. If a student gets 100% on all of the tests, it is evident that he or she is ready to take Calculus III. No further review is required. If a student fails all of the tests, then it would be beneficial for them to take the pre-calculus class, MA1020. For any anomalies such as a student who passed the first and third test but failed the second, the analysis of the student will be sent to another method.

All irregular cases will be subject to further review. In this case, the administrator hand-reviewed the tests. This can become tedious and timely for the advisor. It also is likely to be a similar decision every time. Therefore, we will apply that mode of thinking to the selection software. This will take all of the pass/fail information broken down into topics. It will then take the selected major and the important topics for that major. If there is a failed topic that is necessary for their major, they will be recommended for the class that will teach them that topic again. An example of this would be a physics major who is struggling with derivatives. It is more important for them to learn the topic than to start in an advanced class. Their email would therefore tell them that they may be ready for Calculus III, but they may need additional assistance with derivatives. Therefore, they have the choice of taking the Jumpstart program for rules of differentiation or going back to take Calculus I.

The flowcharts are the early description of the Digital Advising Algorithm. It was later decided that several key changes were necessary. The algorithm is described fully on page 42, with the required changes.

## **7. Improvements to Recommendation System**

### **7.1 Introduction to WeBWorK**

WeBWorK is an open source application designed to deliver homework problems to students over the web. The system is written in Perl, and runs using mod\_perl, an apache module that allows Perl scripts to be run by the Apache web server. This allows WeBWorK to dynamically generate content and makes it easy to add new problems and tests to the system.

### **7.2 Problems with WeBWorK**

The use of WeBWorK for an online calculus placement test is not perfect. The system is designed to deliver homework problems and tests to students. An online placement test seems similar to a test or homework assignment, but there is a fundamental difference. Generally, the results of a homework assignment or exam are used to determine a student's grade in a course. A placement test, on the other hand, is designed to determine what course a student should take.

WeBWorK has excellent tools for determining the grades of a large number of students, but it lacks the ability to give the user immediate feedback based upon the test. This means that a person has to look over all of the students' results and make a recommendation, which can sometimes take a while. To solve this problem, an extension of WeBWorK was created to allow a student to get an immediate preliminary recommendation.

### **7.3 Things no one told you about WeBWorK**

During the course of this Project, a few interesting traits of WeBWorK were discovered. Explaining these should help give insight into our solution to WeBWorK.

WeBWorK uses Perl macros in its problems to do most of the work. It uses these macros to randomize the variables in problems, parse answers, give hints, and much more. However, when `mod_perl` first starts, it needs to know all the Perl scripts that will be used while it is running. `Mod_perl` checks Perl's `@INCLUDE` (which lists the directories where Perl scripts are located) for the scripts it will be running and for all of the scripts they depend on. During this time, it compiles these scripts and stores them to memory. To install a new script, the Apache server needs to be restarted so that `mod_perl` can go through this process again.

Perl has a built-in function called `eval`, which runs an input string as if it were Perl code. WeBWorK uses this function to run macros, as it allows the user to add more macros or change existing macros while the system is running. The problem with this system is that it allows anyone with administrator access to a WeBWorK course to run arbitrary code on the server, a major security breach. WeBWorK therefore runs all problems inside of a sandbox. While running in this sandbox, the macro is able to make any changes it wants inside of its area, but it is unable to access anything outside. The macro has no access to outside modules and all data gets sent directly back to WeBWorK. All it can do to interact with the outside is give the user a link to follow.

Another interesting fact about WeBWorK is that any requests going to POST or GET requests coming from a WeBWorK domain are intercepted by WeBWorK. This means that any data sent from a WeBWorK page by normal means can't escape WeBWorK, and will probably cause WeBWorK to throw the user an error.

## 7.4 The Requirements

The upgrades to WeBWorK were required to meet certain specifications. First, it needed to collect Major Information from the student and store it to the database. The new script also needed to know which AP tests the students had taken as well as the students' results on the Placement Test. The

upgraded script would then use this information to present a recommendation of courses to the student.

## 7.5 The Options

A few options were considered before a final one was chosen. The first solution was to make a small extension to WeBWork such that the user was routed to the recommendation script by WeBWork itself using WeBWork's own internal mechanism and its database connection. This proved to be unfeasible because of how WeBWork handles the routing of users to pages. The solution would have required extensive changes to WeBWork that would prevent easy upgrades to later versions of the software.

The second solution was to create a special final problem and use WeBWork's macro system to gain access to internal WeBWork systems. This solution also proved to be unfeasible because macros do not have access to any external systems, WeBWork or otherwise, except for a few very specific ones. It would be possible to create a way for macros to access the WeBWork database, but it would be a major security issue.

The third solution was to use a WeBWork macro to collect the data and then send it to an external CGI script to process the data. The advantage of this solution is that no changes have to be made to WeBWork itself, meaning the system should continue to work even if WeBWork is upgraded. The disadvantage is that the script has no access to internal WeBWork variables and systems, including its database connection and security protocols.

## 7.6 The Solution

A two step solution was chosen for the final script. In the first step, a WeBWork macro is embedded within a WeBWork problem. This macro dynamically generates a JavaScript form containing



all available majors. The user is then able to select between one and four of these majors by checking the check box adjacent to them. The user then selects the AP courses they have taken and presses the submit button.

When the submit button is pressed, the JavaScript first checks to make sure that between one and four majors have been selected. If more than four or less than one majors have been selected, the JavaScript opens an error window that explains to the user that only one to four majors should be selected.

If the correct number of majors has been selected, the JavaScript then goes on to check which majors have been selected. An array of courses is generated by the Perl macro, and this same array is used by the Perl CGI script that will be receiving the data. Since both sources use the same array of major names, we are able to build a Boolean array of majors to send to the CGI script. Each index in the Boolean array corresponds to the same index in the major array. The Boolean array is initialized so that each value in the array is zero. When a major is selected in the form, the corresponding index in the Boolean array is set to be one. When a major is unselected, the corresponding value is set to be false. The AP information is handled the same way.

The JavaScript now generates a link to the CGI script. Included in this link is the major selection Boolean array, the AP information Boolean array, the student's user name, the current course they are in, the current test they are working on, and their session key. All of this information is used by the CGI script later on.

After submit has been clicked, the user follows the generated link automatically. The user's web browser sends an HTTP request to the Apache web server in order to get the page to be displayed to the user. The CGI script intercepts this request.

The CGI script first parses out the set of information sent to it by the Perl macro. Once it has that, it immediately parses out the student's user name, the course they were in, the test they were taking, and their session key. It uses this information to connect to WeBWorK's database to check that the session key it was given is valid for the current user. If it is not, then an error message is displayed and the script proceeds no further. If the user's information is valid, the script again connects to the WeBWorK database to obtain the student's test scores. The script then uses the majors Boolean array to create an array of the majors that the student selected. The script now has all of the information it needs to make the recommendation.

For each major selected, the script does the following: It first takes the current major from the array and uses it as a key in a hash table containing the courses that need to be taken by each major. It will remove courses from this list based on AP scores and calculus placement scores. Any courses removed from the list have html tags inserted to make them appear crossed out. This way, students know which courses they don't have to take.

The first courses removed are courses the student no longer needs to take due to their AP test scores. Next, courses are removed from the string based on their Calculus placement test scores. Finally, the remaining courses are added to the string that will be returned as html. This will be the page that the user sees.

Lastly, the script stores the major selection information to the WeBWorK database under the recitation field. The user then sees a webpage displaying the major information they selected along with the courses they have been recommended for. (WPI WeBWorK, 2000-2006)

## 8. Conclusion and Recommendations:

### 8.1 Review of Accomplishments

The Calculus Placement IQP of 2009 was an attempt to improve the performance of the WPI Online Calculus Placement Tests. The goals were to first duplicate the previous Calculus Placement IQP study and then make improvements to the system. The statistical analysis of the results for 2008 was found to be consistent with the results found in 2007.

Significant improvements have been made to the WPI Calculus Placement Test. First, the three Placement Tests have been consolidated into a single Calculus Placement Test. Students no longer receive multiple recommendations. The algorithms that previously only existed for automatically sending an email to students who had completed the Calculus I Placement Test were expanded to automate the recommendation system for the combined Placement Test. The recommendation in the email strictly recommends the courses that the student is ready to take and is independent of AP credit and intended major.

The final improvement was the background questionnaire, which was added to the end of the test. This allows the Digital Advisor Algorithm to make its recommendation based on the courses and recommended background that the student needs to fill his or her degree requirements. It also accounts for AP tests that the student has taken and will exclude those classes from the recommendation. This recommendation is provided to the test taker immediately in an html page. It is entirely separate from the recommendation sent via email. This update will be incorporated into the Calculus Placement Test during a system update at the end of the summer in 2009.

These improvements will allow for a smoother recommendation process. They will require less time and effort from the faculty members that run the program. This will also allow for a more personal and accurate course recommendation to incoming freshmen.

## **8.2 Recommendations for the Future**

### **8.2.1 Performance Comparison**

While it may seem obvious that a single test and recommendation message will improve the system, the data should be analyzed when the testing is completed in the summer of 2009. It is possible that the longer test will lead to a lower participation rate. It is also possible that even among students who do start the test, more may give up on parts of the test and receive a recommendation that is lower than expected. Therefore we recommend that a performance analysis be performed.

### **8.2.2 Compliance Comparison**

There should be a study of the percentage of students who took the recommendation given and who followed the recommended courses for “A” and “B” term. This should be evaluated against the data from the previous two years. If the percentage of students following the recommendation has increased in 2013, it can be inferred that the new format is an improvement and should be kept.

### **8.2.3 Jumpstart Redesign**

The Jumpstart review site should be updated and redesigned. Certain aspects of the review material are unclear and poorly formatted.

Jumpstart makes good use of certain tools. Primarily, the site uses examples effectively and consistently. Examples are a valuable tool for clarifying complicated or difficult-to-follow theorems. They show less directly what the theorem states. However, they also show exactly how the theorem is used in application. The examples are also well positioned within the text, placed immediately after the explanation of the theorem. Often times, however, the theorem is stated after a set of examples. Though this layout seems sensible now, grouping all text independent of the set of examples may seem more logical when considering a new design of the website.

The material is often confusing when viewed as a teaching tool. This is especially so when considering that Jumpstart exists for those struggling in certain mathematical topics. It is possible that these “bad examples” are simply good for some students and not for others. Different people learn in different manners. For this reason, it may be practical to provide two explanations of certain topics and/or replace others. If two explanations are provided for the same topic, it is necessary to be very clear that two full explanations are given for those who do not understand the first. The designer may even wish to physically separate the two.

Currently, Jumpstart’s structure is simple and contains very few web pages. There exists a simple hierarchy, but it completely separates the review material from the Jumpstart tests. The review material is also packed into a single, very long page of text and images. The hierarchy needs to be redesigned and the review materials should also be separated into multiple pages for clarity. A simple approach for spreading out the review is to create two pages per topic: topic review and examples for that topic. The separation of review material will also clarify when a new topic will be covered. Currently, it seems unclear when the reader is being introduced to new information. Navigation bars need to be used both for the subject (e.g. Algebra) and the specific topics (e.g. Fractions). This is a useful tool for any web site. Specifically for Jumpstart, students who need to review very few key topics can easily find that topic instead of filtering through the entire subject review for a specific section as they now must. Analogously, the review pages could also contain hyperlinks to and from specific questions in the Jumpstart exams. These tools could have a significant impact on the effectiveness of the Jumpstart program.

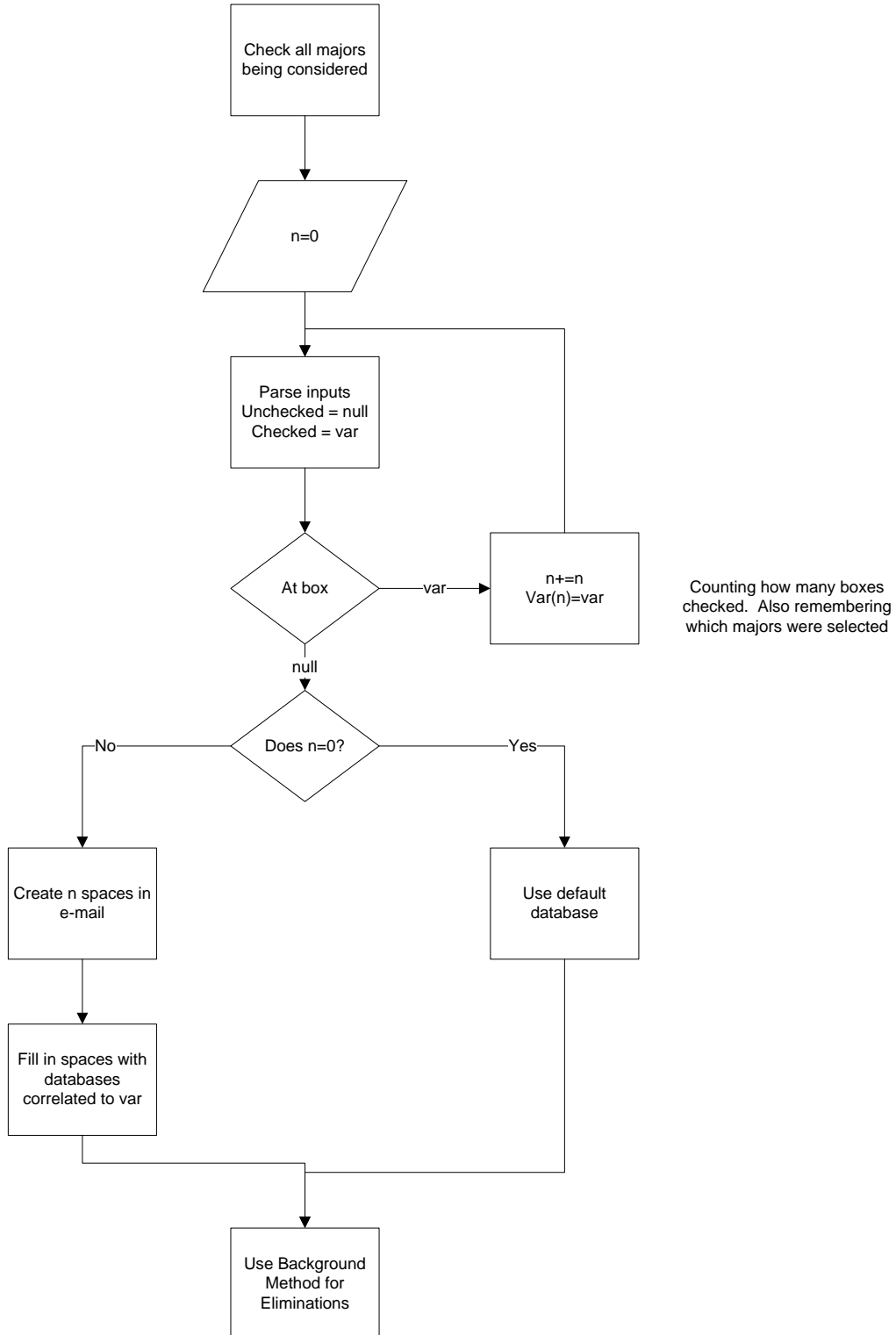
#### **8.2.4 Using Major Data**

The tool to collect student major and AP course information has been developed and should be put in place with the new installation of WeBWork during A and B terms of 2009.

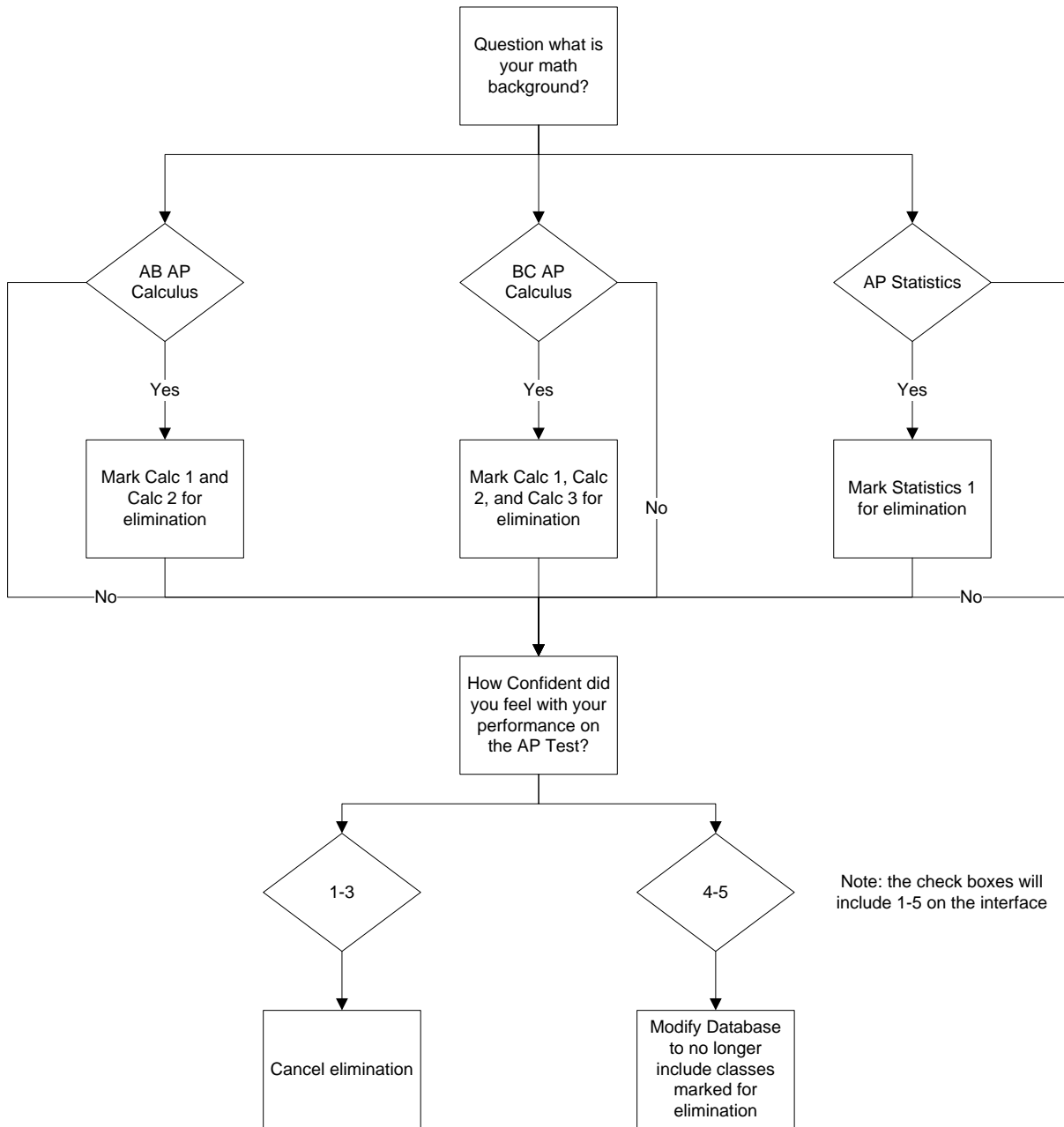
## 9. Appendix

### 9.1 Digital Advising Algorithm Flowcharts

#### 9.1.1 Flowchart 9.1: Inputs

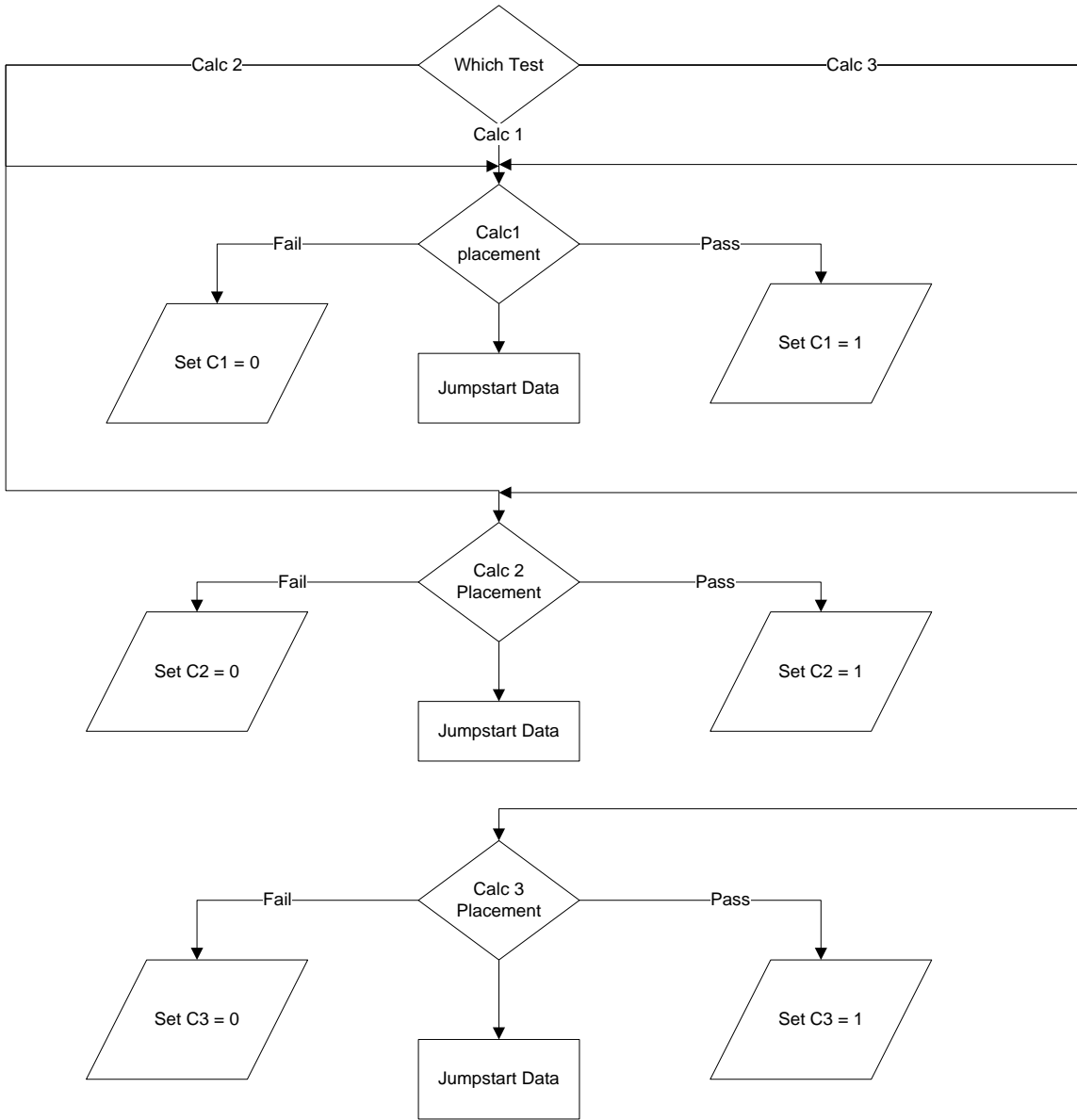


### 9.1.2 Flowchart 9.2: Background Method for Course Elimination

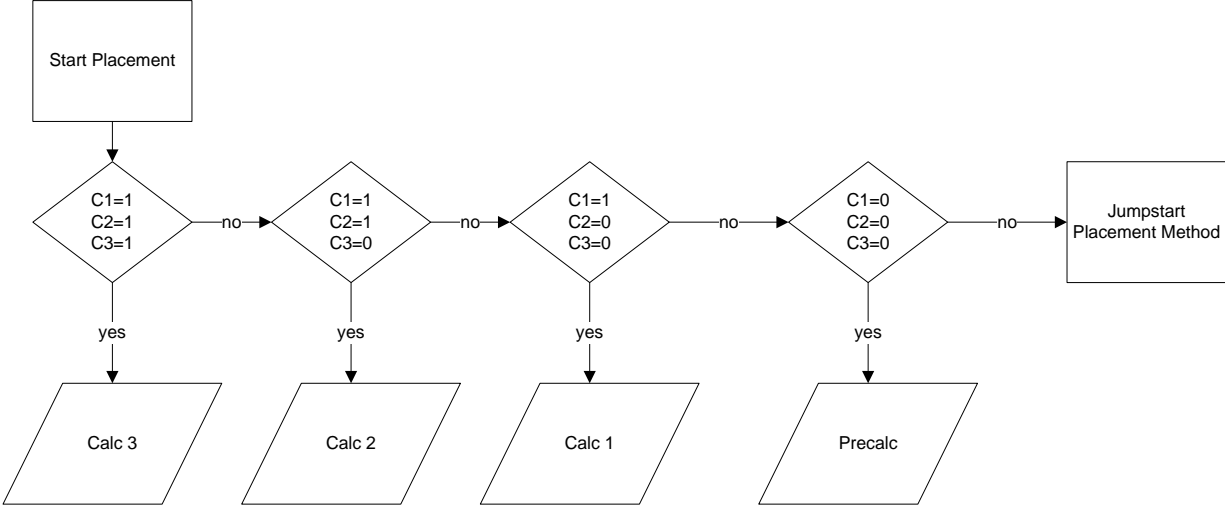




### 9.1.3 Flowchart 9.3: Pass or Fail the Tests



9.1.4 Flowchart 9.4: Recommendation



## 9.2 Perl CGI Script for Major Recommendations and Web Page Generation

```
package IQPTest;
use strict;
use Apache2::RequestRec ();
use Apache2::RequestIO ();
use Apache2::Const -compile =>':common';
use Apache2::Const -compile =>':log';
use Apache2::Log;
use APR::Const;
use WeBWorK;
use WeBWorK::Request;
use WeBWorK::DB;
use DBI;
require "/opt/webwork/pg/macros/IQP.pl";

#####
#Description: This subroutine is called by the Apache server when a request #
#going to the /IQPTest url is recieved                                     #
#                                                                           #
#This subroutine generates sends the HTML generated by the course       #
# recommendation algorithms to the user                                  #
#Arguments:                                                               #
#                                                                           #
# ApacheRequest $apache An apache request object                        #
#                                                                           #
#Returns: A constant that tells the apache server the reques was handled #
# properly.                                                               #
#####
sub handler
{
    #Get the apache request object
    my $apache = shift;
    #$out is the html generated by the recomendationAlgorithm()
    my $out = recomendationAlgoritim($apache);
    #Set the content type to be html so it displays correctly
    $apache->content_type('text/html');
    #Put the new html into the apache request object so it can be displayed
    in the users browser
    $apache->puts('
        <HTML><BODY>
        ' . $out . '
        </BODY></HTML>');
    return Apache2::Const::OK;
}

#####
#Descripton: The recomendationAlgorithm() determines the first year courses#
#a user should take                                                     #
#based upon there Calculus Placement test results, their AP credit,     #
#and there choice                                                         #
# of majors.                                                             #
#                                                                           #
#Arguments:                                                               #
#                                                                           #
#
```

```

#      ApacheRequest $apache An apache request object                                     #
#                                                                                       #
#Returns: String $htmlReturn containin the HTML generated by the algoritim, #
#which also contains the recomended courses                                         #
#####
sub recomendationAlgoritim
{
    #get the apache request object
    my $apache = shift;
    #get the Args field of $apache, within which is the data generated by
    the IQP webwork macro
    my $args = $apache->args();
    # The split args function takes in the $args string and finds the
    $user,$course,and $test substrings, which it then
    # returns. $user is the current users username. $course is the name of
    the webwork course which directed the user
    # here, and $test is the name of the individual test they took.
    my ($user,$course,$test,$key) = splitArgs($args);
    return 'Error, bad authentication' unless
        authenticateUser($user,$course,$test,$key);

    #the args string is limited by the & charachter. Split will take the
    string a return and return substrings based
    #on the given delimiter, in this case &.
    my @info = split /\&/, $args;
    #the first substring should be a string of 0's and 1's representing the
    selected majors. The position of these
    #values in the string correspond to there location in the majors array.
    DeterminMajors() will return an array of
    #Majors corresponding to the location of the ones in the input string.
    my @majors = determineMajors($info[0]);
    # $courses is a pointer to a hash table, in which the key is the major
    name, and the value stored under that key
    # is a list of required courses for that major. courseRequiredByMajor
    generates that hash table and returns a
    # pointer to it.
    my $courses = coursesRequiredByMajor();

    # @testInfo is an array of test scores storing what the user got on
    each section of the calc placement test. Field
    # 0 is the calc 1 score, field 1 is the calc 2 score, and field 2 is
    the calc 3 score. getTestResults() uses the
    # $user,$course, and $test fields to connect to the WeBWork database to
    get that information.
    my @testInfo = getTestResults($user,$course,$test);
    # @testInfo stores an array of boolean values that tell whether the
    user passed the calc 1,2,or3 placement test.
    my @calc = generateCalcScoreArray(@testInfo);
    # $courseForElim is a string containing containing the courses the user
    does not have to take given there AP
    # test scores. $info[1] is a string of 0's and 1's whos locations
    correspond with there locations in the ap test
    # array.
    my $courseForElim = backgroundMethodForCourseElimination($info[1]);
    # $htmlReturn is a string containing the HTML that will be returned by
    this function.
}

```

```

my $htmlReturn = '';
my $dataToStore = '';
#we now generate the recommended classes the student needs to take for
each selected major
foreach(@majors)
{
    $dataToStore = $dataToStore." ".$_;
    $htmlReturn = $htmlReturn.<H3>For a '._.' major we recommend the
        following courses:</H3>';
    #we set $coursesNeeded to be the value of the current major in the
    @major array. This value is a string
    # containing the classes the student needs.
    my $courseNeeded = ${$courses}{$_};
    #$coursesNeededAfterRemoval contains the string of courses the
    student needs, now courses the student does not
    #need will have a line through them when displayed.
    my $courseNeededAfterRemoval =
        removeCourses($courseNeeded,$courseForElim);
    #$coursesNeededAfterCalcTests contains the final recommendation, with
    any course that the student does not need
    #having a line through it to show that it is not necessary. This is
    job is done by the modifyCourses subroutine.
    my $coursesNeededAfterCalcTests =
        modifyCourses(\@calc,$courseNeededAfterRemoval);
    #here we actually add the recommendation to the string that the
    subroutine returns
    $htmlReturn = $htmlReturn.$coursesNeededAfterCalcTests.<br><br>';
}
insertData($user,$course,$dataToStore);
#return the HTML.
return $htmlReturn;
}

sub backgroundMethodForCourseElimination
{
    # $apData a string of 1's and 0's containing information on which ap
    tests were taken, 1 meaning it was taken
    my $apData = shift;
    # $courseForRemoval is an empty string where we will be storing the
    courses that the user does not need to take
    my $courseForRemoval = "";
    # $ABAPScore is a boolean indicating whether the ABAP test was taken
    my $ABAPScore = substr($apData,0,1);
    # $BCAPScore is a boolean indicating whether the BCAP test was taken
    my $BCAPScore = substr($apData,1,1);
    my $APStatsScore = substr($apData,2,1);

    if($ABAPScore > 0 && $BCAPScore <= 0)
    {
        $courseForRemoval = $courseForRemoval."Calculus 1,Calculus 2,";
    }
    elsif($BCAPScore > 0)
    {
        $courseForRemoval = $courseForRemoval."Calculus 1,Calculus 2,Calculus
3,";
    }
}

```

```

        if($APStatsScore > 0)
        {
            $courseForRemoval = $courseForRemoval."Statistics 1";
        }
        return $courseForRemoval;
    }
}

sub removeCourses
{
    my $courses = shift;
    my $removal = shift;
    my @coursesT = split /,/, $courses;
    my @removalT = split /,/, $removal;
    my $count1 = 0;
    my $count2 = 0;
    my $ret = '';
    for($count1 = 0; $count1 < scalar @coursesT; $count1++)
    {
        for($count2 = 0; $count2 < scalar @removalT; $count2++)
        {
            if($coursesT[$count1] eq $removalT[$count2])
            {
                $coursesT[$count1] = '<del>'.$coursesT[$count1]. '</del>';
            }
        }
    }
    for($count1 = 0; $count1 < scalar @coursesT; $count1++)
    {
        $ret = $ret.$coursesT[$count1].',';
    }
    return $ret;
}

#returns refrence to courses
sub coursesRequiredByMajor
{
    my %courses;
    my $calc12 = "Calculus 1,Calculus 2";
    my $calcSeries = $calc12.",Calculus 3,Calculus 4";
    my $odeSeries = $calcSeries.',Ordinary Differential Equations';
    my $stats12 = "Statistics 1,Statistics 2";
    foreach(returnMajors())
    {
        $courses{$_} = "1";
    }
    $courses{"Aerospace Engineering"} = $odeSeries;
    $courses{"Biology and Biotechnology"} = "";
    $courses{"Biomedical Engineering"} = $odeSeries.",Statistics 1";
    $courses{"Chemical Engineering"} = $calc12.
        ",Ordinary Differential Equations";
    $courses{"Biochemistry"} = $calcSeries;
    $courses{"Chemistry"} = $calc12;
    $courses{"Civil Engineering"} = $calc12.",Ordinary Differential
        Equations, ".$stats12.",Probability for Applications";
    $courses{"Computer Science"} = "Discrete Mathematics,
        ".$stats12.",Probability for Applications,Probability";
    $courses{"Electrical and Computer Engineering"} = $odeSeries.

```

```

        ",Discrete Mathematics,Statistics 1,Probability for
        Applications";
$courses{"Humanities and Arts"} = "$calc12";
$courses{"IMGD"} = "";
$courses{"Industrial Engineering"} = $odeSeries.",Statistics
        1,Probability for Applications";
$courses{"Professional Writing"} = "";
$courses{"International Studies"} = "";
$courses{"Liberal Arts and Engineering"} = $calcSeries.",Statistics
        1,Probability for Applications";
$courses{"Management"} = $calc12.", ".$stats12;
$courses{"Management Engineering"} = $calc12.", ".$stats12;
$courses{"Management Information Systems"} = $calc12.", ".$stats12;
$courses{"Manufacturing Engineering"} = $odeSeries;
$courses{"Mathematical Sciences"} = $odeSeries.",Statistics 1";
$courses{"Mechanical Engineering"} = $odeSeries.",Linear Algebra";
$courses{"Physics"} = $odeSeries;
$courses{"Robotics Engineering"} = $odeSeries.",Discrete
        Mathematics,Statistics 1,Probability for Applications";
$courses{"Economic Science"} = $calcSeries.", ".$stats12;
$courses{"Environmental Policy"} = $calcSeries.",Statistics 1";
$courses{"Psychological Science"} = $calc12.", ".$stats12;
$courses{"Society/Technology Policy"} = "";
$courses{"System Dynamics"} = $odeSeries.",Statistics 1";
return \%courses;
}

sub getDBHandle
{
    my $database_dsn = "dbi:mysql:webwork";
    my $database_username = "webworkWrite";
    my $database_password = "password";
    my $database_debug = 0;
    return DBI->connect($database_dsn,$database_username,
        $database_password,{RaiseError => 1,
        AutoCommit => 0});
}

sub getTestResults
{
    my $user = shift;
    my $course = shift;
    my $test = shift;
    my $dbh = getDBHandle();
    my @returnArray;
    my $sth = $dbh->prepare("SELECT problem_id, status FROM ".$course.
        "_problem_user WHERE user_id=? AND set_id=?");
    $sth->execute($user,$test);
    while( my @row = $sth->fetchrow_array)
    {
        $returnArray[$row[0]] = $row[1];
    }
    return @returnArray;
}

sub insertData
{

```

```

my $user = shift;
my $course = shift;
my $commentInfo = shift;
my $dbh = getDBHandle();
my $sth = $dbh->prepare("UPDATE ".$course."_user SET recitation=? WHERE
                        user_id=?");

$sth->execute($commentInfo,$user);
$dbh->disconnect;
}

sub authenticateUser()
{
    my $user = shift;
    my $course = shift;
    my $test = shift;
    #my $key = shift;
    my $dbh = getDBHandle();
    my $sth = $dbh->prepare("SELECT key_not_a_keyword FROM ".$course."_key
                            WHERE user_id=?");

    $sth->execute($user);
    my $ret = "";
    while( my @row = $sth->fetchrow_array)
    {
        $ret = $ret . ' ' . $row[0];
    }

    $dbh->disconnect;
    return $ret;
}

sub splitArgs()
{
    my $args = shift;
    $args =~ /\[?|&]user=(\w*)/;
    my $user = $1;
    $args =~ /\&\/webwork2\/(\w*)\/(\w*)/;
    my $course = $1;
    my $test = $2;
    $args =~ /\[?|&]key=(\w*)/;
    my $key = $1;
    return $user,$course,$test,$key;
}

sub testResultsDebug()
{
    my @testResults = @_;
    my $i = 0;
    my $st = '';
    foreach(@testResults)
    {
        $st = $st.'

```



```

sub getCalcTestScore
{
    my $testInfo = shift;
    my $startQuestion = shift;
    my $endQuestion = shift;
    my $i;
    my $average = 0;
    for($i = $startQuestion; $i < $endQuestion; $i++)
    {
        $average += ${$testInfo}[$i];
    }
    $average = $average / ($endQuestion-$startQuestion);
    return $average;
}

sub getCalc1TestScore
{
    my @testInfo = @_ ;
    my $startQuestion = 1;
    my $endQuestion = 17;
    return getCalcTestScore(\@testInfo, $startQuestion, $endQuestion);
}

sub getCalc2TestScore
{
    my @testInfo = @_ ;
    my $startQuestion = 17;
    my $endQuestion = 25;
    return getCalcTestScore(\@testInfo, $startQuestion, $endQuestion);
}

sub getCalc2TestScore
{
    my @testInfo = @_ ;
    my $startQuestion = 17;
    my $endQuestion = 25;
    return getCalcTestScore(\@testInfo, $startQuestion, $endQuestion);
}

sub getCalc3TestScore
{
    my @testInfo = @_ ;
    my $startQuestion = 25;
    my $endQuestion = 33;
    return getCalcTestScore(\@testInfo, $startQuestion, $endQuestion);
}

sub generateCalcScoreArray
{
    my @testInfo = @_ ;
    my $failure = .70;
    my @calc = (0,0,0);
    if(getCalc1TestScore(@testInfo) >= $failure)
    {
        $calc[0] = 1;
    }
    if(getCalc2TestScore(@testInfo) >= $failure)

```

```

    {
        $calc[1] = 1;
    }
    if(getCalc3TestScore(@testInfo) >= $failure)
    {
        $calc[2] = 1;
    }
    return @calc;
}

sub modifyCourses
{
    my $calc = shift;
    my $courses = shift;
    #if you pass calc 1 test
    if(${ $calc}[0] == 1)
    {
        #and you pass the calc 2 test
        if( ${ $calc}[1] == 1)
        {
            #and you pass the calc 3 test
            if(${ $calc}[2] == 1)
            {
                #then we recommend you take calc 3, not calc one and two
                $courses = removeCourses($courses, 'Calculus 1,Calculus 2');
            }
            #else you still need to take calc 2
            else
            {
                $courses = removeCourses($courses, 'Calculus 1');
            }
        }
        #if you failed calc two
        else
        {
            #and you pass the calc 3 test
            if(${ $calc}[2] == 1)
            {
                #then we recommend you take calc 3, and review calc two
                $courses = removeCourses($courses, 'Calculus 1,Calculus 2');
                $courses = $courses."We recommend that you review Calculus 2
                    materials";
            }
            #else you need to take calc 1-3 so remove nothing
            else
            {
                }
            }
        }
        #so if you fail calc one
        else
        {
            #and you pass the calc 2 test
            if( ${ $calc}[1] == 1)
            {
                #and you pass the calc 3 test

```

```

        if(${ $calc}[2] == 1)
        {
            #then we recommend you take calc 3, not calc one and two and
            you review calc one
            $courses = removeCourses($courses,'Calculus 1,Calculus 2');
            $courses = $courses."We recommend that you review Calculus 1
            materials";

        }
        #else you still need to take calc 2
        else
        {
            $courses = removeCourses($courses,'Calculus 1');
            $courses = $courses."We recommend that you review Calculus 1
            materials";

        }
    }
    #if you failed calc two
    else
    {
        #and you pass the calc 3 test
        if(${ $calc}[2] == 1)
        {
            #then we recommend you take calc 3, and review calc two
            $courses = removeCourses($courses,'Calculus 1,Calculus 2');
            $courses = $courses."We recommend that you review Calculus 1
            and Calculus 2 materials";

        }
        #else you need to take Calculus one with Reliminary topics
        else
        {
            $courses = $courses.'We recommend you take Calculus 1 with
            Preliminary Topics';

        }
    }
}

return $courses;
}

sub determineMajors
{
    my $majorsString = shift;
    my @majors;
    my @listMajors = returnMajors();
    my $i = 0;
    my $pos = 0;
    while($pos != -1 && $i < 4)
    {
        $pos = index $majorsString,'1',$pos;
        if($pos != -1)
        {
            $majors[$i] = $listMajors[$pos];
            $i++;
            $pos++;
        }
    }
}

```

```
    }  
    return @majors;  
}  
1;
```

### 9.3 Perl Macro Developed for WeBWork

```
#####
#####
###
### This macro was written as part of the WPI Online Calculus Placement
###
### Test IQP. The macro generates HTML and javascript that is used to get
###
### information about the users majors of choice and AP test information
### of WeBWork and to an outside script.
### out
###
### Author: Ryan Bedell
###
### Modified: Thursday, March 5, 2009
###
#####
#####

#####
#Description: This subroutine writes HTML and javascript to the screen #
#Arguments: none #
#Returns: none #
#####
sub generateText
{
    # Perl uses the '"' character as a string delimiter, so we use this
    # variable to represent it in strings.
    my $singleQuote = "'";
    # $table is a string that will eventually be the html output for the
    # form for the majors table
    my $table = '<form > <table>';
    # majors is an array of all majors
    my @majors = returnMajors();
    my $i = 0;
    # $size is the size of the @majors array
    my $size = scalar @majors;
    #This is the java script that handles the sending of the information
    # out tof the macro
    TEXT(generateJavaScript($size));

    #generate the table for choosing majors
    while($i < $size)
    {
        #create a new row in the table
        $table = $table . '<tr>';
        #create a cell in the row that has within it a checkbox. When
        #checked it calls the java function
        #createLink() with $i as the argument.
        $table = $table . '<td> <input type ="checkbox" name ="Major"
            value="'. $majors[$i] .' onclick="createLink(
                '$singleQuote.$i.$singleQuote.'"></td>';
        #create a cell in the row that displays the name of the major in
        # the @majors array at location $i
        $table = $table . '<td>'. $majors[$i] . ' ' . '</td>';
    }
}
```

```

#increment $i
$i++;
#create the second column in the table. We need to make sure that
we arn't at the end of
#@majors
if($i < $size)
{
    #create a cell in the row that has within it a checkbox. When
    checked it calls the java function
    #createLink() with $i as the argument
    $table = $table . '<td> <input type ="checkbox" name ="Major"
    value="'. $majors[$i].
    '"onclick="createLink('.$singleQuote.$i.$singleQuote.')">
    </td>';
    #create a cell in the row that displays the name of the major
    in the @majors array at location $i
    $table = $table . '<td>'. $majors[$i]. ' ' </td>';
}
#end the current row in the table
$table = $table . '</tr>';
#increment $i
$i++;
}
#end the table
$table = $table . '</table>';
#write text to problem
TEXT('
    <br>
    <br>  Select between one and four majors you are considering:
    <br>
    ');
#write $table to problem
TEXT($table);
#write text to screen
TEXT('
    <br>
    Select yes if you took the listed AP test:
    <br>
    <br>
    ');
#write AP table to screen
TEXT(generateApTableHTML());
#write the submit button linked to the sendData() function to screen
TEXT('<input type="button" value="Submit" onclick="sendData()">
    </form>');
}

#####
#Description: this returns an array of majors #
#Arguments: none #
#Returns: An array of majors #
#####
sub returnMajors
{
    my @majors =('Aerospace Engineering','Biology and

```

```

        Biotechnology','Biomedical Engineering',
        'Chemical Engineering','Biochemistry','Chemistry','Civil
Engineering','Computer Science',
        'Electrical and Computer Engineering','Humanities and
Arts','Industrial Engineering',
        'IMGD','Professional Writing','International
Studies','Liberal Arts and Engineering',
        'Management','Management Engineering','Management
Information Systems','Manufacturing Engineering',
        'Mathematical Sciences','Mechanical Engineering',
        'Physics','Robotics Engineering','Economic Science',
        'Environmental Policy','Psychological Science',
        'Society/Technology Policy','System Dynamics');
    return @majors;
}

#####
#Description: This returns the ap courses we care about#
#Arguments: none #
#Returns: An array of AP courses #
#####
sub returnAPInfo
{
    return ('AB AP Calculus','BC AP Calculus', 'AP Statistics');
}

#####
#Description: This should be called such that the html it generats is inside#
#a form. #
#Arguments: none #
# #
#Returns: a String containing the HTML for the Ap Credit Selection Table #
# #
#####
sub generateApTableHTML
{
    # $apTable is a string that will contain the information for the apTable
    my $apTable = '<table>';
    #Get the names of the AP tests we are looking for
    my @apTests = returnAPInfo();
    #Get the size of the @apTests array
    my $size = scalar @apTests;
    #A single quote is Perls string delimiter, so we need a variable for it
    if we want to use it in a string
    my $singleQuote = "'";
    # $i is a standard counting variable for a loop
    my $i = 0;
    #begin construction of the HTML for $apTable. A "." is used in Perl for
    string concatenation. <tr> creates a
    #row in the table.
    $apTable = $apTable . '<tr>';
    #loop through each ap test listed in @apTests. Arrays start there
    indexing at zero.
    while($i < $size)
    {
        #add a new HTML cell to the table containing the name of the AP test

```

```

    in the @apTests array at position $i
    $apTable = $apTable . '<td>----'. $apTests[$i]. '----</td>';
    #increment $i so we continue iterating throug @apTests
    $i++;
}
#We are done with the current row in the table, so we set close the
current row
$apTable = $apTable . '</tr>';
#We reset $i to zero so that we can use it to iterate again.
$i = 0;
#Create another row in the table;
$apTable = $apTable . '<tr>';
#Iterate through the @apTable array;
while($i < $size)
{
    #this line creates a new cell in the row. Inside that cell it
    creates a table, and inside that table it creates
    #a new row, inside of which it creates two new cells. In the first
    cell is a radio button that is used to signal
    #a users choice of whether they took the listed ap test and got a
    four or five. This radio button indicates no.
    #this radio button, when selected, calls the javascript function
    setRadio() with the arguments
    #the value of $i and 0. This radio button starts selected. The next
    cell contains the words no.
    $apTable = $apTable . '<td> <table><tr><td><input type="radio"
        name="'. $apTests[$i]. ' value="No" checked
        onclick="setRadio('.$singleQuote.$i.$singleQuote.',
        '$singleQuote.'0'.$singleQuote.')">
        </td> <td> No </td></tr></table></td>';

    #increment $i
    $i++;
}
#end the current row in the main table
$apTable = $apTable . '</tr>';
#reset $i
$i = 0;
#start a new row
$apTable = $apTable . '<tr>';
#Iterate through the @apTable array;
while($i < $size)
{
    #this line creates a new cell in the row. Inside that cell it
    creates a table, and inside that table it creates
    #a new row, inside of which it creates two new cells. In the first
    cell is a radio button that is used to signal
    #a users choice of whether they took the listed ap test and got a
    four or five. This radio button indicates yes.
    #this radio button, when selected, calls the javascript function
    setRadio() with the arguments
    #the value of $i and 1.The next cell contains the words yes.
    $apTable = $apTable . '<td> <table><td><input type="radio"
        name="'. $apTests[$i]. ' value="Yes"
        onclick="setRadio('.$singleQuote.$i.$singleQuote.',
        '$singleQuote.'1'.$singleQuote.')">
        </td> <td> Yes </td></table></td>';

    #Increment $i

```



```

        $i++;
    }
    #end the current row
    $apTable = $apTable . '</tr>';
    #end the current table
    $apTable = $apTable . '</table>';
    #return the string $apTable
    return $apTable;
}

#####
#Description: This script generates all the javascript used by this script #
#through the use of helper functions. #
# #
#Arguments: #
# #
# integer $size the size of the @majors array #
# #
#Returns: a string containing the HTML and javascript generated #
# #
#####
sub generateJavaScript
{
    #take in argument size, an integer representing the total number of
    majors
    my $size = shift;
    #the number of ap tests we consider
    my $apSize = 3;
    my $javaScript = '
        <script type="text/javascript">
            var toPass = new Array();
            var apCalcData = new Array();
            var size = '.$size.';
            var apSize='.$apSize.'
            var count = 0;
            while(count < size)
            {
                toPass[count] = "0";
                count=count+1;
            }
            count = 0;
            while(count < apSize)
            {
                apCalcData[count] = "0";
                count=count+1;
            }
        ';
    $javaScript = $javaScript . generateJSFcreateLink();
    $javaScript = $javaScript . generateJSFsendData();
    $javaScript = $javaScript . generateJSFgetURIInfo();
    $javaScript = $javaScript . generateJSFsetRadio();
    $javaScript = $javaScript . '</script>';
    return $javaScript;
}

```

```

#####
#Description: Generates the javascript function createLink(). Should only be#
#called from generateJavaScript()                                         #
#Arguments: none                                                           #
#                                                                           #
#Returns: a string containing the javascript generated                     #
#                                                                           #
#####
sub generateJSFcreateLink
{
    return      'function createLink(i)
                {
                    if(toPass[i] == "0")
                    {
                        toPass[i] = "1";
                    }
                    else
                    {
                        toPass[i] = "0";
                    }
                }';
}

#####
#Description: Generates the javascript function sendData(). Should only be #
#called from generateJavaScript()                                         #
#Arguments: none                                                           #
#                                                                           #
#Returns: a string containing the javascript generated                     #
#                                                                           #
#####
sub generateJSFsendData
{
    #This is the location of the outside script that the information is
    being sent to
    my $locationToSendData = 'http://127.0.0.1/IQPTTest';
    return 'function sendData()
            {
                var toSend = "";
                var x = 0;
                var numSelected = 0;
                for(x in toPass)
                {
                    toSend = toSend + toPass[x];
                    if(toPass[x] == "1")
                    {
                        numSelected = numSelected + 1;
                    }
                }
                toSend = toSend + "&";
                x = 0;
                for(x in apCalcData)
                {
                    toSend = toSend + apCalcData[x];
                }
                if(numSelected > 4)
                {

```

```

        alert("Too many majors have been selected. Please select
              between one and four majors.")
    }
    else if(numSelected == 0)
    {
        alert("You must select at least one major")
    }
    else
    {
        window.location =
            "'.$locationToSendData.'?" + toSend + "&" + getURIInfo();
    }
}';
}

#####
#Description: Generates the javascript function getURIInfo(). Should only be#
#called from generateJavaScript()                                           #
#Arguments: none                                                            #
#                                                                           #
#Returns: a string containing the javascript generated                      #
#                                                                           #
#####
sub generateJSFgetURIInfo
{
    return 'function getURIInfo()
        {
            var fullUrl = location.href;
            fullUrl = fullUrl.replace(/\?/, "&");
            fullUrl = fullUrl.replace(/http.*\//web/, "/web");
            return fullUrl;
        }';
}

#####
#Description: Generates the javascript function setRadio(). Should only be #
#called from generateJavaScript()                                           #
#Arguments: none                                                            #
#                                                                           #
#Returns: a string containing the javascript generated                      #
#                                                                           #
#####
sub generateJSFsetRadio
{
    return 'function setRadio(num,value)
        {
            apCalcData[num] = value;
        }';
}

#the file is required to return 1
1;

```

## 9.4 Excell Macro

### 9.4.1 Visual Basic Code after Update

'This is needed to create the mail merge later

```
Enum WdMailMergeComparison
    wdMergelfEqual
    wdMergelfGreaterThan
    wdMergelfGreaterThanOrEqual
    wdMergelfIsBlank
    wdMergelfIsNotBlank
    wdMergelfLessThan
    wdMergelfLessThanOrEqual
    wdMergelfNotEqual
End Enum
```

```
Sub Calc_1_Placement_Score_Analysis()
```

```
'-----
'Used to analyze student scores on the WPI Calculus 1 Placement Test
'Computes total score and four sub-scores, stores this information in Excel
'Creates an email merge in Word to each student who has just completed the test
'Email contains recommendations for math course selection and Jump Start programs
```

```
'Daniel Smaltz, May 2007
```

```
'Updated by Lawson Glidden, Alex Ramsay, and Ryan Bedell, May 2009
```

```
'-----
```

```
'Jump Start recommendation threshold
'At JS_failing or below, we recommend Jump Start
'Out of 4 possible points
Const JS_failing As Double = 2
```

```
'Precalculus/Calculus I/Calculus II/Calculus III recommendation thresholds
```

```
'At Precalc_passing or above, we recommend MA 1021
Const Precalc_passing As Double = 13
```

```
'At Precalc_failing or below, we recommend MA 1020
Const Precalc_failing As Double = 8
```

```
'At Calc2_passing or above, we recommend MA 1022, any score lower than 6 is failing
Const Calc2_passing As Double = 6
```

```
'At Calc3_passing or above, we recommend MA 1023, any score lower than 7 is failing
Const Calc3_passing As Double = 7
```

```

'Determine the number of students
Dim students As Integer
students = 0

Sheets("Data").Select
While Not IsEmpty(Range("A8").Offset(students, 0))
    students = students + 1
Wend

If students = 0 Then
    MsgBox ("The Data sheet is empty or improperly formatted. Paste all Webwork data into cell A1 of
"Data"")
    Exit Sub
End If

'Determine number of data sheets exist currently
'They are numbered sequentially as they are stored
Dim numberOfSheets As Integer

Sheets("List").Select
Range("Y1").Value = Range("Y1").Value + 1
numberOfSheets = Range("Y1").Value

'Format an output table in the new sheet "Output"
FormatOutput

'listStudents -> number of students in "List"
Dim listStudents As Integer
listStudents = 0

Sheets("List").Select
While Not IsEmpty(Range("A2").Offset(listStudents, 0))
    listStudents = listStudents + 1
Wend

Sheets("Data").Select

'Variables for processing data

Dim loginID As String, lastname As String, firstname As String
    'Student information
Dim algebra As Double, trig As Double
    'Scores on individual subjects
Dim geometry As Double, functions As Double
    'Scores on individual subjects, cont'd
Dim total As Double
    'Total score on Calc 1 test

```

```

Dim calc2total As Double
    'Total score on Calc 2 test
Dim calc3total As Double
    'Total score on Calc 3 test
Dim JSalgebra As String, JStrig As String
Dim JSgeometry As String, JSfunctions As String
    'JumpStart recommendations
Dim precalc As String, advisor As String, calc As String
    'Recommendations for calc/precalc
Dim calc2 As String, calc3 As String
    'Recommendations for calc 2 and 3
Dim i As Integer, j As Integer, k As Integer
    'Indices for "for" loops
Dim finished As Boolean
    'True if student has finished the test, i.e. has completed last problem
Dim currentStudent As String
    'stores the Login ID of the student currently being analyzed
Dim studentIsListed As Boolean
    'True if a student who has finished is already in the "List" sheet
Dim newStudents As Integer
    'Number of finished students who are not yet in "List"

```

```
newStudents = 0
```

```

'Data processing loop
For i = 1 To students
    Sheets("Data").Select
    finished = False
    studentIsListed = False

```

```

    algebra = 0
    trig = 0
    geometry = 0
    functions = 0
    total = 0
    calc2total = 0
    calc3total = 0
    calc = "N"
    calc2 = "N"
    calc3 = "N"
    precalc = "N"
    advisor = "N"

```

```

'Determine who has completed Problem 34 - this indicates that they have finished
If Range("DA8").Offset(i - 1, 0).Value = 1 Then
    finished = True
End If

```

If finished = True Then

```
'Determine if a student in "Output" is already in "List"  
'If not, include the student in "List"  
Sheets("Data").Select  
currentStudent = EliminateWhitespace(Range("A7").Offset(i, 0))
```

```
'Check if current student is already in the list  
Sheets("List").Select  
For k = 1 To listStudents  
    If currentStudent = Range("A1").Offset(k, 0) Then  
        studentIsListed = True  
    End If  
Next k
```

```
If studentIsListed = False Then  
    newStudents = newStudents + 1  
    Sheets("Data").Select
```

```
'Store student info in vars  
    loginID = EliminateWhitespace(Range("A8").Offset(i - 1, 0))  
    lastname = Range("A8").Offset(i - 1, 2)  
    firstname = Range("A8").Offset(i - 1, 3)
```

```
'Determine algebra score  
    For k = 1 To 4  
        If Range("F8").Offset(i - 1, 3 * (k - 1)).Value > 0 Then  
            algebra = algebra + Range("F8").Offset(i - 1, 3 * (k - 1)).Value  
        End If  
    Next k
```

```
'Determine trig score  
    For k = 5 To 8  
        If Range("F8").Offset(i - 1, 3 * (k - 1)).Value > 0 Then  
            trig = trig + Range("F8").Offset(i - 1, 3 * (k - 1)).Value  
        End If  
    Next k
```

```
'Determine geometry score  
    For k = 9 To 12  
        If Range("F8").Offset(i - 1, 3 * (k - 1)).Value > 0 Then  
            geometry = geometry + Range("F8").Offset(i - 1, 3 * (k - 1)).Value  
        End If  
    Next k
```

```
'Determine functions score  
    For k = 13 To 16  
        If Range("F8").Offset(i - 1, 3 * (k - 1)).Value > 0 Then
```

```

        functions = functions + Range("F8").Offset(i - 1, 3 * (k - 1)).Value
    End If
Next k

'Determine Calc1 score
'Subtract 1 because we don't want to count Problem 17

    total = algebra + trig + geometry + functions

'Determine Calc 2 score
For k = 17 To 24
    If Range("F8").Offset(i - 1, 3 * (k - 1)).Value > 0 Then
        calc2total = calc2total + Range("F8").Offset(i - 1, 3 * (k - 1)).Value
    End If
Next k

'Determine Calc 3 score
For k = 25 To 33
    If Range("F8").Offset(i - 1, 3 * (k - 1)).Value > 0 Then
        calc3total = calc3total + Range("F8").Offset(i - 1, 3 * (k - 1)).Value
    End If
Next k

'Determine recommendations
'Jump Start recommendations
JSalgebra = JumpStart(algebra, JS_failing)
JStrig = JumpStart(trig, JS_failing)
JSgeometry = JumpStart(geometry, JS_failing)
JSfunctions = JumpStart(functions, JS_failing)

'Precalc or Calculus?
If (total < Precalc_passing And total > Precalc_failing) Then
    advisor = "Y"
    precalc = "Y"
    calc = "N"
    calc2 = "N"
    calc3 = "N"
End If

If total <= Precalc_failing Then
    advisor = "N"
    precalc = "Y"
    calc = "N"
    calc2 = "N"
    calc3 = "N"
End If

If total >= Precalc_passing Then

```



```
    advisor = "N"  
    precalc = "N"  
    calc = "Y"  
    calc2 = "N"  
    calc3 = "N"  
End If
```

```
If calc2total >= Calc2_passing Then  
    advisor = "N"  
    precalc = "N"  
    calc = "N"  
    calc2 = "Y"  
    calc3 = "N"  
End If
```

```
If calc3total >= Calc3_passing Then  
    advisor = "N"  
    precalc = "N"  
    calc = "N"  
    calc2 = "N"  
    calc3 = "Y"  
End If
```

```
'Output all student info into a table in "Output"  
Sheets("Output").Select
```

```
Range("A1").Offset(newStudents, 0).Value = loginID  
Range("A1").Offset(newStudents, 1) = Range("A1").Offset(newStudents, 0) & "@wpi.edu"  
Range("A1").Offset(newStudents, 2).Value = lastname  
Range("A1").Offset(newStudents, 3).Value = firstname  
Range("A1").Offset(newStudents, 4) = Format(Date, "dd-mmm-yy")  
Range("A1").Offset(newStudents, 5).Value = algebra  
Range("A1").Offset(newStudents, 6).Value = trig  
Range("A1").Offset(newStudents, 7).Value = geometry  
Range("A1").Offset(newStudents, 8).Value = functions  
Range("A1").Offset(newStudents, 9).Value = total  
Range("A1").Offset(newStudents, 10).Value = calc2total  
Range("A1").Offset(newStudents, 11).Value = calc3total  
Range("A1").Offset(newStudents, 12).Value = JSalgebra  
Range("A1").Offset(newStudents, 13).Value = JSstrig  
Range("A1").Offset(newStudents, 14).Value = JSgeometry  
Range("A1").Offset(newStudents, 15).Value = JSfunctions  
Range("A1").Offset(newStudents, 16).Value = precalc  
Range("A1").Offset(newStudents, 17).Value = calc  
Range("A1").Offset(newStudents, 18).Value = calc2  
Range("A1").Offset(newStudents, 19).Value = calc3  
Range("A1").Offset(newStudents, 20).Value = advisor  
Sheets("Data").Select
```

```

'Copy their information into "List"
  For k = 0 To 20
    Sheets("Output").Select
    Range("A1").Offset(newStudents, k).Copy
    Sheets("List").Select
    Range("A1").Offset(listStudents + newStudents, k).PasteSpecial xlPasteValues
  Next k
End If
End If
Next i

```

'Now "Output" contains all students who are finished with the test AND not already listed  
'And the students in "Output" have been added to "List"

```

'Copy "Data" for record-keeping
Sheets("Data").Select
Cells.Copy
Sheets.Add
ActiveSheet.Name = "Data (" & numberOfSheets & ")"
Range("A1").PasteSpecial xlPasteAll
Range("A1").Select

```

```

'Clear "Data" for next use
Sheets("Data").Select
Cells.Clear
Range("A1").Select
Sheets("List").Select
Range("A1").Select

```

```

'It is possible that no new students have finished since data was last downloaded
If newStudents = 0 Then
  MsgBox ("No new students have completed the test since score data was last processed.")
  'Sheet "Output" no longer needed. Delete it.
  Application.DisplayAlerts = False
  Sheets("Output").Delete
  Application.DisplayAlerts = True
  Sheets("List").Select
  Range("A1").Select
  Exit Sub
End If

```

```

'Copy "Output" for record-keeping
Sheets("Output").Select
Cells.Copy
Sheets.Add
ActiveSheet.Name = "Output (" & numberOfSheets & ")"
Range("A1").PasteSpecial xlPasteValues
Columns("F:J").Select

```

```

Selection.NumberFormat = "0.0"
Columns("E:E").Select
Selection.NumberFormat = "dd-mmm-yy"
Range("A1").Select

'Sheet "Output" no longer needed. Delete it.
Application.DisplayAlerts = False
Sheets("Output").Delete
Application.DisplayAlerts = True

Sheets("Data (" & numberOfSheets & ")).Move Before:=Sheets("Data")
Sheets("Output (" & numberOfSheets & ")).Move Before:=Sheets("Data")

'-----
'MAIL MERGE WITH WORD
'-----

Dim WorkbookName As String, SheetName As String
WorkbookName = ThisWorkbook.FullName
Filename = ThisWorkbook.Name

Dim namelength As Integer, Start As Integer, SavePath As String
namelength = Len(WorkbookName) - Len(Filename)
SavePath = Mid(WorkbookName, 1, namelength)

SheetName = "`Output (" & numberOfSheets & ")$"
Dim roundingCode As String
roundingCode = "\#0.0"

'Read email text from Excel (Sheet "Email")
Dim eRecommendCalc As String

Sheets("E-mail").Select
eSalutation = Range("A2").Text
eIntroduction = Range("A4").Text
eCalc1A = Range("A6").Text
eCalc1B = Range("C6").Text
eRecommendPrecalc = Range("A74").Text
eRecommendCalc = Range("A62").Text
eRecommendCalc2 = Range("A64").Text
eRecommendCalc3 = Range("A66").Text
eRecommendAdvisorA = Range("A12").Text
eRecommendAdvisorB = Range("A14").Text
eSubscore = Range("A16").Text
eCalc2a = Range("A8").Text
eCalc2B = Range("C8").Text

eCalc3a = Range("A10").Text
eCalc3b = Range("C10").Text

```

```
eAlgebraScoreA = Range("A18").Text
eAlgebraScoreB = Range("C18").Text
eRecommendNoJSAlgebra = Range("A20").Text
eRecommendJSAlgebra = Range("A22").Text
```

```
eTrigScoreA = Range("A24").Text
eTrigScoreB = Range("C24").Text
eRecommendNoJSTrig = Range("A26").Text
eRecommendJSTrig = Range("A28").Text
```

```
eGeometryScoreA = Range("A30").Text
eGeometryScoreB = Range("C30").Text
eRecommendNoJSGeometry = Range("A32").Text
eRecommendJSGeometry = Range("A34").Text
```

```
eFunctionsScoreA = Range("A36").Text
eFunctionsScoreB = Range("C36").Text
eRecommendNoJSFunctions = Range("A38").Text
eRecommendJSFunctions = Range("A40").Text
```

```
eConclusionA = Range("A42").Text
eConclusionB = Range("A44").Text
eConclusionC = Range("A68").Text
eConclusionD = Range("A70").Text
eSummer = Range("A46").Text
eClosing = Range("A48").Text
eSignatureA = Range("A50").Text
eSignatureB = Range("A52").Text
eSignatureC = Range("A54").Text
eTab = Range("A72").Text
```

```
'Create a Word document, and make it a mail merge document
```

```
Set wrdApp = CreateObject("Word.Application")
```

```
wrdApp.Visible = True
```

```
Set wrdDoc = wrdApp.Documents.Add
```

```
wrdDoc.Select
```

```
Set wrdSelection = wrdApp.Selection
```

```
Set wrdMailMerge = wrdDoc.MailMerge
```

```
Set wrdMergeFields = wrdMailMerge.Fields
```

```
'Enter text and mail merge fields into Word document
```

```
wrdDoc.MailMerge.MainDocumentType = wdEMail
```

```
wrdDoc.MailMerge.OpenDataSource Name:=WorkbookName, _
```

```
SQLStatement:="SELECT * FROM" & SheetName, SQLStatement1:="", SubType:= _
```

```
wdMergeSubTypeAccess
```

```
wrdApp.Selection.TypeText "Dear "
```

```

wrdDoc.MailMerge.Fields.Add wrdApp.Selection.Range, "First_Name"
wrdApp.Selection.TypeText ","
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText eIntroduction
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText Chr(11)

wrdApp.Selection.TypeText eTab
wrdApp.Selection.TypeText eCalc1A
wrdApp.Selection.TypeText " "
wrdDoc.MailMerge.Fields.Add wrdApp.Selection.Range, "Calc_1_score" & roundingCode
wrdApp.Selection.TypeText " "
wrdApp.Selection.TypeText eCalc1B
wrdApp.Selection.TypeText " "
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText eTab
wrdApp.Selection.TypeText eCalc2a
wrdApp.Selection.TypeText " "
wrdDoc.MailMerge.Fields.Add wrdApp.Selection.Range, "Calc_2_score" & roundingCode
wrdApp.Selection.TypeText " "
wrdApp.Selection.TypeText eCalc2B
wrdApp.Selection.TypeText " "
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText eTab
wrdApp.Selection.TypeText eCalc3a
wrdApp.Selection.TypeText " "
wrdDoc.MailMerge.Fields.Add wrdApp.Selection.Range, "Calc_3_score" & roundingCode
wrdApp.Selection.TypeText " "
wrdApp.Selection.TypeText eCalc3b
wrdApp.Selection.TypeText " "
wrdApp.Selection.TypeText Chr(11)
'-----
' Begin the Calc Recommendation
'-----

wrdApp.Selection.TypeText eTab
wrdDoc.MailMerge.Fields.AddIf wrdApp.Selection.Range, "Calc3", wdMergeIfEqual, _
"Y", "", eRecommendCalc3, "", ""

wrdDoc.MailMerge.Fields.AddIf wrdApp.Selection.Range, "Calc2", wdMergeIfEqual, _
"Y", "", eRecommendCalc2, "", ""

wrdDoc.MailMerge.Fields.AddIf wrdApp.Selection.Range, "Calc", wdMergeIfEqual, _
"Y", "", eRecommendCalc, "", ""

wrdDoc.MailMerge.Fields.AddIf wrdApp.Selection.Range, "Precalc", wdMergeIfEqual, _
"Y", "", eRecommendPrecalc, "", ""

```

wrdApp.Selection.TypeText Chr(11)

wrdDoc.MailMerge.Fields.AddIf wrdApp.Selection.Range, "Advisor", wdMergelfEqual, \_  
"Y", "", eRecommendAdvisorA, "", ""

wrdDoc.MailMerge.Fields.AddIf wrdApp.Selection.Range, "Advisor", wdMergelfEqual, \_  
"Y", "", " ", "", ""

wrdDoc.MailMerge.Fields.AddIf wrdApp.Selection.Range, "Advisor", wdMergelfEqual, \_  
"Y", "", eRecommendAdvisorB, "", ""

'-----

'End of Calc Recommendation

'-----

wrdApp.Selection.TypeText Chr(11)

wrdApp.Selection.TypeText eSubscore

wrdApp.Selection.TypeText Chr(11)

wrdApp.Selection.TypeText Chr(11)

wrdApp.Selection.TypeText eTab

wrdApp.Selection.TypeText eAlgebraScoreA

wrdApp.Selection.TypeText " "

wrdDoc.MailMerge.Fields.Add wrdApp.Selection.Range, "Algebra\_Score" & roundingCode

wrdApp.Selection.TypeText " "

wrdApp.Selection.TypeText eAlgebraScoreB

wrdApp.Selection.TypeText " "

wrdApp.Selection.TypeText Chr(11)

wrdApp.Selection.TypeText eTab

wrdApp.Selection.TypeText eTrigScoreA

wrdApp.Selection.TypeText " "

wrdDoc.MailMerge.Fields.Add wrdApp.Selection.Range, "Trig\_Score" & roundingCode

wrdApp.Selection.TypeText " "

wrdApp.Selection.TypeText eTrigScoreB

wrdApp.Selection.TypeText " "

wrdApp.Selection.TypeText Chr(11)

wrdApp.Selection.TypeText eTab

wrdApp.Selection.TypeText eGeometryScoreA

wrdApp.Selection.TypeText " "

wrdDoc.MailMerge.Fields.Add wrdApp.Selection.Range, "Geometry\_Score" & roundingCode

wrdApp.Selection.TypeText " "

wrdApp.Selection.TypeText eGeometryScoreB

wrdApp.Selection.TypeText " "

wrdApp.Selection.TypeText Chr(11)

```
wrdApp.Selection.TypeText eTab
wrdApp.Selection.TypeText eFunctionsScoreA
wrdApp.Selection.TypeText " "
wrdDoc.MailMerge.Fields.Add wrdApp.Selection.Range, "Functions_Score" & roundingCode
wrdApp.Selection.TypeText " "
wrdApp.Selection.TypeText eFunctionsScoreB
wrdApp.Selection.TypeText " "
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText Chr(11)
```

```
wrdApp.Selection.TypeText eConclusionA & " " & eConclusionB
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText Chr(11)
```

```
wrdApp.Selection.TypeText eConclusionC & " " & eConclusionD
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText Chr(11)
```

```
wrdApp.Selection.TypeText eSummer
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText Chr(11)
```

```
wrdApp.Selection.TypeText eClosing
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText Chr(11)
```

```
wrdApp.Selection.TypeText eSignatureA
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText eSignatureB
wrdApp.Selection.TypeText Chr(11)
wrdApp.Selection.TypeText eSignatureC
```

```
MsgBox ("A mail merge to " & newStudents & " students has been configured in a new Word document.")
```

```
'Save the Word document
wrdApp.ChangeFileOpenDirectory SavePath
wrdDoc.SaveAs "Mail Merge (" & numberOfSheets & ")", wdFormatDocument
```

```
Sheets("Output (" & numberOfSheets & ")").Select
```

```
'Save the Excel spreadsheet
ActiveWorkbook.Save
```

```
End Sub
```

```
Public Function JumpStart(ByVal subject As Double, JS_failing As Double) As String
```

```

If subject <= JS_failing Then
    JumpStart = "Y"
Else
    JumpStart = "N"
End If
End Function

```

```

Private Sub FormatOutput()
'Creates the sheet "Output" to place score information for newly added students
'Adds a header row to "Output" labeling each column

```

```

Sheets.Add
ActiveSheet.Name = "Output"

Range("A1").Offset(0, 0) = "Login ID"
Range("A1").Offset(0, 1) = "E-mail"
Range("A1").Offset(0, 2) = "Last Name"
Range("A1").Offset(0, 3) = "First Name"
Range("A1").Offset(0, 4) = "Date Added"
Range("A1").Offset(0, 5) = "Algebra Score"
Range("A1").Offset(0, 6) = "Trig Score"
Range("A1").Offset(0, 7) = "Geometry Score"
Range("A1").Offset(0, 8) = "Functions Score"
Range("A1").Offset(0, 9) = "Calc 1 Score"
Range("A1").Offset(0, 10) = "Calc 2 Score"
Range("A1").Offset(0, 11) = "Calc 3 Score"
Range("A1").Offset(0, 12) = "JS Algebra"
Range("A1").Offset(0, 13) = "JS Trig"
Range("A1").Offset(0, 14) = "JS Geometry"
Range("A1").Offset(0, 15) = "JS Functions"
Range("A1").Offset(0, 16) = "Precalc"
Range("A1").Offset(0, 17) = "Calc"
Range("A1").Offset(0, 18) = "Calc2"
Range("A1").Offset(0, 19) = "Calc3"
Range("A1").Offset(0, 20) = "Advisor"
End Sub

```

```

Public Function EliminateWhitespace(ByVal strString As String) As String
'This code obtained from http://www.vba-programmer.com/
'To replace all occurrences of one string in another
'Load every incoming character into an array
Dim MyCharArray()
For i = 1 To Len(strString)
    ReDim Preserve MyCharArray(i)
    MyCharArray(i) = Mid(strString, i, 1)
Next

'Scan the array and eliminate anything not an alpha or numeric character

```



```

For i = 1 To UBound(MyCharArray())
  If MyCharArray(i) Like "[!0-9A-Za-z._]" Then
    MyCharArray(i) = ""
  End If
Next

```

```

'Build the output string
Dim strTemp As String
For i = 1 To UBound(MyCharArray())
  If Len(MyCharArray(i)) > 0 Then
    strTemp = strTemp & MyCharArray(i)
  End If
Next

```

```

'Exit the function
EliminateWhitespace = strTemp
End Function

```

### 9.4.1 Excel Data Sheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	1 test1	test1	student			0	0	3	0	0	3	0	0	3	0	0	3	0	0
2	2 test2	test2	student			0	0	3	0	0	3	0	0	3	0	0	3	0	0
3	3 test3	test3	student			0	0	3	0	0	3	0	0	3	0	0	3	0	0
4	4 test4	test4	student			0	0	3	0	0	3	0	0	3	0	0	3	0	0
5	5 test5	test5	student			1	1	0	1	1	0	1	1	0	1	1	0	1	1
6	6 test6	test6	student			1	1	0	1	1	0	1	1	0	1	1	0	1	1
7	7 test7	test7	student			1	1	0	1	1	0	1	1	0	1	1	0	1	1
8	8 test8	test8	student			1	1	0	1	1	0	1	1	0	1	1	2	0	1
9	9 test9	test9	student			1	1	0	1	1	0	1	1	0	1	1	0	0	0
T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	
3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0	3
3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0	3
3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0	3
3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0	3
0	1	1	0	1	2	0	1	1	0	1	1	0	1	1	0	1	1	1	0
0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0
0	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0
0	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0
3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0	3
AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	
0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0
0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0
0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0
0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0
1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	3
1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	3
1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	3
1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	3
1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	3
1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	3
BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	
0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0
0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0
1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	3
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0
0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0
0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	3	0	0	0
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1

BZ	0	CA	0	CB	3	CC	0	CD	0	CE	3	CF	0	CG	0	CH	3	CI	0	CJ	0	CK	3	CL	0	CM	0	CN	3	CO	0	CP	0	CQ	3	CR	0
	1		1		0		1		1		0		1		1		0		1		1		0		1		1		1		1		1		2		1
	0		0		3		0		0		3		0		0		3		0		0		3		0		0		3		0		0		3		0
	1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1
	0		0		3		0		0		3		0		0		3		0		0		3		0		0		3		0		0		3		0
	1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1
	0		0		3		0		0		3		0		0		3		0		0		3		0		0		3		0		0		3		0
	1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1
	0		0		3		0		0		3		0		0		3		0		0		3		0		0		3		0		0		3		0
	1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1
	1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1
	1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1		1		0		1
CS	0	CT	3	CU	0	CV	0	CW	3	CX	0	CY	0	CZ	3	DA	1	DB	3	DC	0	DD	1														
	1		1		1		1		0		1		1		0		1		1		1		0		10												
	0		3		0		0		3		0		0		3		1		3		0		9														
	1		0		1		1		0		1		1		0		1		1		0		18														
	0		3		0		0		3		0		0		3		1		1		0		17														
	1		1		1		1		0		1		1		0		1		1		0		25														
	0		3		0		0		3		0		0		3		1		1		0		25														
	1		0		1		1		0		1		1		0		1		1		0		34														
	1		1		1		1		0		1		1		0		1		1		0		30														

## 9.4.2 Excel List Sheet

E-mail	Last Name	First Name	Date Added	Algebra Score	Trig Score	Geometry Score	Functions Score	Calc 1 Score	Calc 2 Score	Calc 3 Score
1@wpi.edu	test1	student	11-May-09	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2@wpi.edu	test2	student	11-May-09	0.0	0.0	0.0	0.0	0.0	0.0	9.0
3@wpi.edu	test3	student	11-May-09	0.0	0.0	0.0	0.0	0.0	8.0	0.0
4@wpi.edu	test4	student	11-May-09	0.0	0.0	0.0	0.0	0.0	8.0	9.0
5@wpi.edu	test5	student	11-May-09	4.0	4.0	4.0	4.0	16.0	0.0	0.0
6@wpi.edu	test6	student	11-May-09	4.0	4.0	4.0	4.0	16.0	0.0	8.0
7@wpi.edu	test7	student	11-May-09	4.0	4.0	4.0	4.0	16.0	8.0	0.0
8@wpi.edu	test8	student	11-May-09	4.0	4.0	4.0	4.0	16.0	8.0	9.0
9@wpi.edu	test9	student	11-May-09	4.0	0.0	4.0	4.0	12.0	8.0	9.0

JS Algebra	JS Trig	JS Geometry	JS Functions	Precalc	Calc1	Calc2	Calc3	Advisor
Y	Y	Y	Y	Y	N	N	N	N
Y	Y	Y	Y	N	N	N	Y	N
Y	Y	Y	Y	N	N	Y	N	N
Y	Y	Y	Y	N	N	N	Y	N
N	N	N	N	N	Y	N	N	N
N	N	N	N	N	N	N	Y	N
N	N	N	N	N	N	Y	N	N
N	N	N	N	N	N	N	Y	N
N	Y	N	N	N	N	N	Y	N

## Works Cited

Education Resources Information Center. (n.d.). *Eric Home*. Retrieved December 2008, from Eric: <http://www.eric.ed.gov/>

Leary, S., & Smaltz, D. (2008). *Analysis of the WPI On-Line Calculus Placement Tests*. Worcester, MA: WPI.

Temple University Russell Conwell Educational Services Center. (2006). *Summer Bridge Program 2009, Sample Math Placement Test*. Retrieved January 18, 2009, from Temple University Russell Conwell Educational Services Center: [http://www.temple.edu/rcc/math\\_placement.htm](http://www.temple.edu/rcc/math_placement.htm)

Testing and Evaluation Services. (n.d.). *Contents of the Mathematics Placement Test*. Retrieved January 18, 2009, from Testing and Evaluation Services: <http://testing.wisc.edu/math%20test.html>

WPI George C. Gordon Library. (2009, April 17). *WPI George C. Gordon Library*. Retrieved December 2008, from WPI George C. Gordon Library: <http://www.wpi.edu/Academics/Library/>

*WPI WeBWork*. (2000-2006). Retrieved November 2008, from WeBWork: <https://webwork.wpi.edu/>