EMORY

ROLLINS SCHOOL OF PUBLIC HEALTH

DEPARTMENT: BIOSTATISTICS AND BIOINFORMATICS

COURSE NUMBER: 531 **SECTION NUMBER: 2**

CREDIT HOURS: 2 **SEMESTER: FALL**

COURSE TITLE: SAS PROGRAMMING

CLASS HOURS AND LOCATION: M 1:00-2:50 PM

INSTRUCTOR NAME: PAUL WEISS

INSTRUCTOR CONTACT INFORMATION

EMAIL: paul.weiss@emory.edu

PHONE: (404)712-9641

SCHOOL ADDRESS OR MAILBOX LOCATION: GCR 308

OFFICE HOURS M 10:00 – 11:30

COURSE DESCRIPTION

This class is designed to help students master statistical programming in SAS. Students in this class will develop programming style and skills for data manipulation, report generation, simulation and graphing. This class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH). That being said, SAS is a primary data analysis and data management software system in use worldwide, particularly in public health settings. Students who master the skills offered in this course will have a much easier time completing the work for their thesis and will find themselves more ready for a public health career with a more analytical bent.

MPH/MSPH FOUNDATIONAL COMPETENCIES:

This class does not meet any foundational competencies as described by the Rollins School of Public Health.

CONCENTRATION COMPETENCIES:

This class does not meet any foundational competencies as described by the Rollins School of Public Health.

COURSE LEARNING OBJECTIVES:

Students in this class will learn how to use SAS's programming language to manipulate data and solve complex statistical problems using simulation.

EVALUATION

There will be 4 projects comprising 75% of the final grade each. Two exams will comprise the remaining 25% (midterm=10%, final=15%). The Base SAS Certification Exam is an optional test that will not count towards the student's grade. Students are encouraged to sit for the certification exam; we are pleased to offer the exam with an Emory discount of approximately 50% off the regular price. This Emory discount does not take the place of the one-time student discount already offered by SAS, so students may take the exam with us using our discount and a different exam (or a retake of the base exam) with the student discount at a later date. The SAS exam is optional and the resulting grade will not be figured into the students' final grades regardless of the result.

GRADING:

- [96 + A
- [91 – 96) A-
- [86 – 91) B+
- [81 – 86) B
- [76 – 81) B-
- [66 – 76) C
- Below 66 F

COURSE STRUCTURE

Lecture 1: Introductions (Syllabus, The RSPH Network, SAS)
Project handout distributed – Project #1 assigned

Project #1 covers merging and manipulating data for processing. Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this homework project does not address any specific competency. However, data manipulation is the most fundamental skill in any analytical field and an essential tool for working with data in Public Health.

Lecture 2: Introducing SAS Datasets (Reading in, Creating)

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. However, SAS is one of the most commonly used statistical packages for processing and analyzing data in the world. Being able to read in and process SAS datasets into the SAS system is a foundational skill upon which all other SAS skills are built.

Lecture 3: Sorting, Merging and Concatenating

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. However, datasets are often presented in multiple parts (e.g. multiple waves of a survey or various components of a clinical study linked by record number) and being able to process and assemble an analytical dataset from a collection of smaller files is an essential tool in biostatistical methods.

Lecture 4: Dates, Times and Datetimes

Project #1 due
Project #2 assigned

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. SAS handles dates and times differently from other computer packages and understanding how they work in SAS opens up large avenues of programming facility.

Project #2 covers report generation and table creation, two essential skills in analytics but not specifically identified as competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH).

Lecture 5: Putting, Data Management and Report Writing

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. However, being able to assemble a dataset for presentation of results and writing the results out to a data file or report are essential tools in every area of public health. Creating reports and filling in table shells by hand is an arduous process that is prone to error. The amount of work required to fill in a table shell and then double-check it for mistakes is time consuming and problematic. Being able to use SAS to create the numbers for the table and then fill it in as well saves time and improves efficiency on multiple levels. While these skills may be undervalued as competencies, they are extremely valuable in the working environment and mastering these skills will increase a student's value in the workforce.

Lecture 6: Midterm Exam (one hour)
Introduction to DO statements, iterative coding and loops
Project #2 due

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH), the midterm exam does not address any specific competency. The exam will engage students' ability to read and diagnose code. They will be presented with a program (on paper) that is plagued with a number of coding errors. Students will identify the erroneous statements and provide a fix for each error. Students are awarded one point for each error they identify and an additional point if their fix is satisfactory. They will not receive the additional point if their fix does not work; they will also lose a point if they make a change to a statement that works that results in making the statement wrong. For example, if the student sees: IF x = 7 THEN y = "TRUE"; and changes the statement to: If X = 7 then Y = "TRUE"; they will not receive or lose any points. If the student sees IF x = 7 THEN y = "TRUE"; and changes the statement to: If (X = 7) {Y = "TRUE"}; then they will lose a point because the statement will no longer work in SAS. Being able to debug code is an essential skill for any programming language; a SAS master will be able to look at code in any setting and diagnose errors without having to rely on the enhanced SAS editor for hints on where the errors might be. The best SAS programmers will have an idea of where the error resides simply by anecdotal description of the problem. Our assessment at this juncture is a straightforward assessment of students' mastery of very basic concepts.

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH), looping does not address any specific competency. This is a more advanced concept in data manipulation and plays a huge part in simulation. Being able to create and control looping structures opens huge avenues for programming that would otherwise be unavailable to the neophyte programmer.

Lecture 7: Arrays
Project #3 assigned

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. However, arrays are extremely useful as data structures in a host of programming languages. SAS Arrays are static structures comprised of variables which may or may not exist at the time the array is specified. Being able to refer to variables in a list by index makes code more efficient and can reduce the number of data defining statements considerably.

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. Project #3 gives students the opportunity to write a simulation to calculate the winning probability of a common game of chance. Traditionally, we use two different games (Craps and Golo) alternating year by year. These games challenge students to write efficient code to simulate the game under a simple strategy and then modify the strategy and compare the results. Simulation is an essential tool in statistical programming for calculating probabilities in the absence of a closed form solution. Complex power problems, the impact of various assumptions and Bayesian posterior distributions are just a few examples of where simulation could provide an adequate answer when more mathematical options fail.

Lecture 8: Macro

Project #4 assigned

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. The SAS Macro language is an advanced topic that allows programmers to create modular code and improve their efficiency at a slight cost in speed. The macro language allows programmers to replace hard-coded values with macro variables and assign their value once rather than having them assigned in multiple places directly. Cutting and pasting code is a common practice that is fraught with peril. Changing a “1” to a “2” in 30 places in a copied block of code and then replicating the methodology for an additional 20 blocks leads to long programs and many errors to debug. Using a macro to call the block multiple times and change the value in a single parameter assignment makes the code easier to read and leaves much less room for error. Macro improves coding style and efficiency but the change in processor leads to a decrease in speed. Students will learn about the macro processor and the tradeoff as they master this very useful advanced topic.

Project #4 does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH). This project will allow students to develop a macro that calls their simulation from project #3 and allow the user to define the parameters of the game under multiple strategies and such. This is the most challenging project of the semester and is worth the least amount of points due to its degree of difficulty and timing.

Lecture 9: IML

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. The SAS Interactive Matrix Language is an advanced topic that allows programmers to manipulate data in matrices and vectors. This topic is not assessed directly by the projects or exams.

Lecture 10: Graphics

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. The SAS System for Graphics is an advanced topic that allows programmers to display data in a host of different methods. This topic is not assessed directly by the projects or exams.

Lecture 11: Miscellaneous topics / Catch-up
Project #4 due

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this lecture does not address any specific competency. We build in some room at the end of the semester to cover any topics we were unable to get to in the event of school closings and class cancellation beyond our control. Any topics covered in this class period will not be assessed on the final.

Lecture 12: Final Exam (two hours)
Introduction to DO statements, iterative coding and loops
Project #2 due

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH), the midterm exam does not address any specific competency. The exam will engage students' ability to read and diagnose code. They will be presented with a program (on paper) that is plagued with a number of coding errors. Students will identify the erroneous statements and provide a fix for each error. Students are awarded one point for each error they identify and an additional point if their fix is satisfactory. They will not receive the additional point if their fix does not work; they will also lose a point if they make a change to a statement that works that results in making the statement wrong. For example, if the student sees: IF x = 7 THEN y = "TRUE"; and changes the statement to: If X = 7 then Y = "TRUE"; they will not receive or lose any points. If the student sees IF x = 7 THEN y = "TRUE"; and changes the statement to: If (X = 7) {Y = "TRUE"}; then they will lose a point because the statement will no longer work in SAS. Being able to debug code is an essential skill for any programming language; a SAS master will be able to look at code in any setting and diagnose errors without having to rely on the enhanced SAS editor for hints on where the errors might be. The best SAS programmers will have an idea of where the error resides simply by anecdotal description of the problem. Our assessment at this juncture is comprehensive; students will be presented with looping and macro code to diagnose and debug as well as the more fundamental concepts addressed in the base certification exam.

COURSE POLICIES

HOMEWORK POLICY:

Homework will be concentrated in two major term programming assignments. These assignments will allow you to apply the concepts we cover throughout the class. Final solutions to these assignments may include calculating simple statistics from a dataset, generating tables and reports, or building a simulation to explore some statistical phenomenon. Students will be graded on the accuracy of the presented information as well as the presentation of the program and results themselves. As programming is an art form, students will not be graded wholly on the efficiency of their program this semester, but on their creativity in applying what they've learned in solving the problem as well. Each assignment will have a number of deliverables assigned as separate projects and all requiring SAS in some way. You will need to decide the best way to solve the problems presented. You will turn in your program and sometimes some additional documentation depending on the required deliverables.

A midterm exam will be given in October and the final exam will be given on the last regular class day. These exams will allow students an opportunity to demonstrate SAS skills by debugging code written by another programmer. These exams will be administered by hard copy; students will not be allowed to use SAS to complete these exams.

The optional Base SAS Certification Exam will be given during the regular final exam schedule. We will set up quizzes on Blackboard that you can take any time during the semester. The quizzes and exams are good preparation for taking the SAS Certification exam. The SAS Certification exam and the quizzes will **not** figure into the course grade. Additional review sessions will be scheduled for students who are interested in taking the SAS exam. These sessions will be outside of normal class times. Students who pass the certification exam will automatically receive a 100 for the midterm or final exam.

IMPORTANT INFORMATION:

This class will serve as a prerequisite for BIOS 532 Statistical Computing. This class concentrates on statistical programming and not on data analysis. Students who are looking for a data analysis course should consider other electives in Biostatistics. This class is very computer intensive, since becoming familiar with PC SAS will prepare students as they start considering career options.

Statisticians analyze data. Programmers solve problems. Statistical Programmers solve data analysis problems. You may have been trained to think like a statistician – this class will try to get you to think like a programmer. Therefore, a statistical background is not essential for this class, but previous programming experience could come in very handy. People who have experience in object-oriented languages like C and C++ will find R and S-Plus much easier to pick up. People with experience in top-down languages like Pascal and BASIC will find SAS more to their liking.

As the instructor of this course I endeavor to provide an inclusive learning environment. However, if you experience barriers to learning in this course, do not hesitate to discuss them with me and the Office for Equity and Inclusion, 404-727-9877.

RSPH POLICIES

Accessibility and Accommodations

Accessibility Services works with students who have disabilities to provide reasonable accommodations. In order to receive consideration for reasonable accommodations, you must contact the Office of Accessibility Services (OAS). It is the responsibility of the student to register with OAS. Please note that accommodations are not retroactive and that disability accommodations are not provided until an accommodation letter has been processed.

Students who registered with OAS and have a letter outlining their academic accommodations are strongly encouraged to coordinate a meeting time with me to discuss a protocol to implement the accommodations as needed throughout the semester. This meeting should occur as early in the semester as possible.

Contact Accessibility Services for more information at (404) 727-9877 or accessibility@emory.edu. Additional information is available at the OAS website at http://equityandinclusion.emory.edu/access/students/index.html

Honor Code

You are bound by Emory University's Student Honor and Conduct Code. RSPH requires that all material submitted by a student fulfilling his or her academic course of study must be the original work of the student. Violations of academic honor include any action by a student indicating dishonesty or a lack of integrity in academic ethics. *Academic dishonesty refers to cheating, plagiarizing, assisting other students without authorization, lying, tampering, or stealing in performing any academic work, and will not be tolerated under any circumstances.*

The RSPH Honor Code states: "Plagiarism is the act of presenting as one's own work the expression, words, or ideas of another person whether published or unpublished (including the work of another student). A writer's work should be regarded as his/her own property."
(http://www.sph.emory.edu/cms/current_students/enrollment_services/honor_code.html)

COURSE CALENDAR

Tentative Lecture Outline

Week 1: Introductions (Syllabus, the RSPH Network, SAS)
Week 2: SAS Datasets (Reading in and creating)
Week 3: Sorting, Merging and Concatenating
Week 4: Dates, Times and Datetimes
Week 5: Putting, Data Management and Report Writing
Week 6: The DO Statement, Iterative Coding and Loops
Week 7: Arrays
Week 8: Macro
Week 9: IML
Week 10: Graphics
Week 11: Miscellaneous Topics / Catch-up
Week 12: **FINAL EXAM IN CLASS**
Week 13: SAS Certification Exam (optional)

COURSE OUTLINE

Lecture 1: Introductions (Syllabus, The RSPH Network, SAS)

Project handout distributed – Project #1 assigned

Project #1 covers merging and manipulating data for processing. Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this homework project does not address any specific competency. However, data manipulation is the most fundamental skill in any analytical field and an essential tool for working with data in Public Health.

Lecture 2: Introducing SAS Datasets (Reading in, Creating)

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. However, SAS is one of the most commonly used statistical packages for processing and analyzing data in the world. Being able to read in and process SAS datasets into the SAS system is a foundational skill upon which all other SAS skills are built.

Lecture 3: Sorting, Merging and Concatenating

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. However, datasets are often presented in multiple parts (e.g. multiple waves of a survey or various components of a clinical study linked by record number) and being able to process and assemble an analytical dataset from a collection of smaller files is an essential tool in biostatistical methods.

Lecture 4: Dates, Times and Datetimes

Project #1 due
Project #2 assigned

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. SAS handles dates and times differently from other computer packages and understanding how they work in SAS opens up large avenues of programming facility.

Project #2 covers report generation and table creation, two essential skills in analytics but not specifically identified as competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH).

Lecture 5: Putting, Data Management and Report Writing

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. However, being able to assemble a dataset for presentation of results and writing the results out to a data file or report are essential tools in every area of public health. Creating reports and filling in table shells by hand is an arduous process that is prone to error. The amount of work required to fill in a table shell and then double-check it for mistakes is time consuming and problematic. Being able to use SAS to create the numbers for the table and then fill it in as well saves time and improves efficiency on multiple levels. While these skills may be undervalued as competencies, they are extremely valuable in the working environment and mastering these skills will increase a student's value in the workforce.

Lecture 6: Midterm Exam (one hour)
Introduction to DO statements, iterative coding and loops
Project #2 due

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH), the midterm exam does not address any specific competency. The exam will engage students' ability to read and diagnose code. They will be presented with a program (on paper) that is plagued with a number of coding errors. Students will identify the erroneous statements and provide a fix for each error. Students are awarded one point for each error they identify and an additional point if their fix is satisfactory. They will not receive the additional point if their fix does not work; they will also lose a point if they make a change to a statement that works that results in making the statement wrong. For example, if the student sees: IF x = 7 THEN y = “TRUE”; and changes the statement to: If X = 7 then Y = “TRUE”; they will not receive or lose any points. If the student sees IF x = 7 THEN y = “TRUE”; and changes the statement to: If (X = 7) {Y = “TRUE”}; then they will lose a point because the statement will no longer work in SAS. Being able to debug code is an essential skill for any programming language; a SAS master will be able to look at code in any setting and diagnose errors without having to rely on the enhanced SAS editor for hints on where the errors might be. The best SAS programmers will have an idea of where the error resides simply by anecdotal description of the problem. Our assessment at this juncture is a straightforward assessment of students’ mastery of very basic concepts.

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH), looping does not address any specific competency. This is a more advanced concept in data manipulation and plays a huge part in simulation. Being able to create and control looping structures opens huge avenues for programming that would otherwise be unavailable to the neophyte programmer.

Lecture 7: Arrays

Project #3 assigned

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. However, arrays are extremely useful as data structures in a host of programming languages. SAS Arrays are static structures comprised of variables which may or may not exist at the time the array is specified. Being able to refer to variables in a list by index makes code more efficient and can reduce the number of data defining statements considerably.

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. Project #3 gives students the opportunity to write a simulation to calculate the winning probability of a common game of chance. Traditionally, we use two different games (Craps and Golo) alternating year by year. These games challenge students to write efficient code to simulate the game under a simple strategy and then modify the strategy and compare the results. Simulation is an essential tool in statistical programming for calculating probabilities in the absence of a closed form solution. Complex power problems, the impact of various assumptions and Bayesian posterior distributions are just a few examples of where simulation could provide an adequate answer when more mathematical options fail.

Lecture 8: Macro

Project #4 assigned

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. The SAS Macro language is an advanced topic that allows programmers to create modular code and improve their efficiency at a slight cost in speed. The macro language allows programmers to replace hard-coded values with macro variables and assign their value once rather than having them assigned in multiple places directly. Cutting and pasting code is a common practice that is fraught with peril. Changing a “1” to a “2” in 30 places in a copied block of code and then replicating the methodology for an additional 20 blocks leads to long programs and many errors to debug. Using a macro to call the block multiple times and change the value in a single parameter assignment makes the code easier to read and leaves much less room for error. Macro improves coding style and efficiency but the change in processor leads to a decrease in speed. Students will learn about the macro processor and the tradeoff as they master this very useful advanced topic.

Project #4 does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH). This project will allow students to develop a macro that calls their simulation from project #3 and allow the user to define the parameters of the game under multiple strategies and such. This is the most challenging project of the semester and is worth the least amount of points due to its degree of difficulty and timing.

Lecture 9: IML

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. The SAS Interactive Matrix Language is an advanced topic that allows programmers to manipulate data in matrices and vectors. This topic is not assessed directly by the projects or exams.

Lecture 10: Graphics

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH) this topic does not address any specific competency. The SAS System for Graphics is an advanced topic that allows programmers to display data in a host of different methods. This topic is not assessed directly by the projects or exams.

Lecture 11: Miscellaneous topics / Catch-up
Project #4 due

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on

Education for Public Health (CEPH) this lecture does not address any specific competency. We build in some room at the end of the semester to cover any topics we were unable to get to in the event of school closings and class cancellation beyond our control. Any topics covered in this class period will not be assessed on the final.

Lecture 12: Final Exam (two hours)
Introduction to DO statements, iterative coding and loops
Project #2 due

Since this class does not directly satisfy any competencies as defined by the Department of Biostatistics and Bioinformatics, the Rollins School of Public Health or the Council on Education for Public Health (CEPH), the midterm exam does not address any specific competency. The exam will engage students' ability to read and diagnose code. They will be presented with a program (on paper) that is plagued with a number of coding errors. Students will identify the erroneous statements and provide a fix for each error. Students are awarded one point for each error they identify and an additional point if their fix is satisfactory. They will not receive the additional point if their fix does not work; they will also lose a point if they make a change to a statement that works that results in making the statement wrong. For example, if the student sees: IF x = 7 THEN y = "TRUE"; and changes the statement to: If X = 7 then Y = "TRUE"; they will not receive or lose any points. If the student sees IF x = 7 THEN y = "TRUE"; and changes the statement to: If (X = 7) {Y = "TRUE"}; then they will lose a point because the statement will no longer work in SAS. Being able to debug code is an essential skill for any programming language; a SAS master will be able to look at code in any setting and diagnose errors without having to rely on the enhanced SAS editor for hints on where the errors might be. The best SAS programmers will have an idea of where the error resides simply by anecdotal description of the problem. Our assessment at this juncture is comprehensive; students will be presented with looping and macro code to diagnose and debug as well as the more fundamental concepts addressed in the base certification exam.