RECORD OF ACADEMIC ACHIEVEMENT

The Evergreen State College - Olympia, Washington 98505

OFFICIAL TRANSCRIPT DOCUMENT

Rexroth, Michael Anthony — A00326438

Last, First Middle — Student ID

DEGREES CONFERRED:

Bachelor of Arts — Awarded 16 Jun 2017

TRANSFER CREDIT:

Start	End	Credits	Title
09/2006	06/2011	90	**Pierce College**

EVERGREEN UNDERGRADUATE CREDIT:

Start	End	Credits	Title
09/2015	06/2016	48	**Computer Science Foundations** *5 - Introduction to Programming in Java I* *5 - Introduction to Programming in Java II* *4 - Introduction to Functional Programming in Haskell* *4 - Digital Logic and Computer Organization* *4 - Computer Architecture and Assembly Programming* *4 - Microcontroller Applications* *4 - Discrete Mathematics I* *4 - Discrete Mathematics II* *4 - Discrete Mathematics III* *3 - Problem Solving Workshop* *3 - Seminar: Ethics and Technology* *4 - Seminar: Computational Thinking*
09/2016	03/2017	28	**Computability and Language Theory** **4 - Formal Languages I* **4 - Formal Languages II* **4 - Artificial Intelligence* **4 - Functional Programming in Haskell I* **4 - Functional Programming in Haskell II* **4 - Programming Language Design I* **4 - Programming Language Design II*
01/2017	03/2017	4	**Unix System Administration** **4 - Unix System Administration*
04/2017	06/2017	16	**Computing Practice and Theory: Consumer Behavior** *4 - Statistics* *4 - Data Mining* *4 - Consumer Behavior* *4 - Project: Data Mining and Analytics*

Cumulative

186 Total Undergraduate Credits Earned

Rexroth, Michael Anthony — A00326438
Last, First Middle — Student ID

Starting at Evergreen I brought with me a background of work and life experience. Through running-start, I had managed to complete my Associates with ease, however lack of money prevented me from continuing my education path. Instead I found work and started a family of my own. I knew I had more in me though. I excelled in school, and was always quick to learn in class. This led me to Evergreen, a local school that I could afford. The location and price is what brought me to its beautiful campus. The unexpected teaching curriculum and programs available were what kept me passionate in my choice of majors. I've always been passionate about science and technology. Their capabilities to help everyone in so many ways have kept me captivated since childhood. After two years of carefully balancing my family, school, and work life, I've completed my Bachelor's Degree majoring in Computer Science (CS).

During my first year at Evergreen I dove straight in with a 3-quarter program purely devoted to my planned major, Computer Science Foundations (CSF). Beginning at the bottom we learned the basics of CS including truth tables, discrete mathematics, and building our logic gates up to a basic system with the support of Logisim. Continuing up the line we learned programming at the low level with Jasmin, Big-O calculations, Intermediate Java, and in Evergreen tradition, held a quarter long seminar on Computer Ethics. To complete our broad foundation in CSF we studied Graphing in mathematics, building with Arduinos, Intermediate Haskell, and the all-important Induction proofs. With the completion of CSF I had a strong foundation to jump into different sections of my major.

With the beginning of year two I began the study of Computability and Language Theory. My main objective in this class being a stronger understanding of programming and what is possible with it. The studies began with Programming Languages Principles and Paradigms. The focus of this thread was to study how languages are defined and implemented. I completed this by learning about lexical analysis, parsing, and type systems through creating said components for a C-lite language. Additionally, through study and assignments I gained an understanding of how high-level languages gain their meaning and are interpreted. This also gave a strong foundation for the different paradigms of programming languages, seeing how the creation of the choices made leading up to the constructor can strongly change the language. Our second intense thread during this class was Formal Languages where we studied the classification of languages, abstract machines, string/language recognition, regular & context-free grammars, Turing machines, the Halting problem, and everything between. Completing this 2-quarter class includes the study of Artificial Intelligence with its neural networks, Advanced Haskell, and a study on Linux System Administration.

To complete my final year at Evergreen I split off to gain a foundation in a connected and important field, Data Mining. I learned how to analyze Big Data to find unseen systems and the traps to avoid while doing so. This was complemented with an understanding of Statistics and Consumer Behavior. Finally, to use what I've learned, we studied the following programs; R, Weka, & Tableau. Bringing all this knowledge together I completed this quarter by assisting Lutheran Community Services North West, a non-profit organization, by analyzing their available data and presenting to them my findings.

With finishing my Bachelor's degree at Evergreen, I walk away with a broad foundation in Computer Science, a strong understanding of Programming, and the useful skills of Data Analysis. This added to my background of eight years actively working with people in retail, my studies in Digital Design at Peirce College, and of course my life experience as a Father gives me strong confidence for my future career.

Rexroth, Michael Anthony — A00326438
Last, First Middle — Student ID

April 2017 - June 2017: Computing Practice and Theory: Consumer Behavior

16 Credits

DESCRIPTION:
Faculty: Richard Weiss, Ph.D. and Jon Baumunk, J.D., M.S.A.

The Computing Practice and Theory: Consumer Behavior program was a full-time, 16-credit program for intermediate and advanced computer science and business students, which wove together the theory and practice of data science: statistics, consumer behavior, and data mining and analytics in the context of Big Data. Through seminar texts, writing assignments, quizzes, independent research, and group projects and presentations, students examined the question of how various organizations collect and analyze large amounts of data to discover or confirm patterns. Students worked in small groups to apply computing sub-disciplines of their choice to a programming and statistics project and presented their work both orally and in written reports at the end of the quarter.

Seminar texts included Leonard Mlodinow's "The Drunkard's Walk: How Randomness Rules Our Lives," Mark Maier's and Jennifer Imazeki's "The Data Game: Controversies in Social Science Statistics," Donella Meadows's "Thinking in Systems: A Primer," and Cathy O'Neil's "Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy," representing several disciplines. Other texts covered statistics, consumer behavior, and programming in R and Weka. In seminar, students discussed concepts from the readings and lectures. In addition to seminars and lectures on statistics, data mining, and consumer behavior, this project-oriented program's disciplinary components focused on: 1) machine learning and pattern recognition and 2) modeling and visualization.

This all-level program gave the upper division students opportunities to continue work begun in other computer science programs. Students met for lectures, seminars, workshops, and labs. Students worked independently and in small groups, collaboratively focusing on improving critical reading, thinking, and analytical skills. They presented their work in both written and verbal forms and were given weekly assessments. Students leaving this program should be better equipped to work with data and procedures in conducting business and public policy. They should also be better prepared for the quantitative and qualitative requirements of business and government.

EVALUATION:
Written by: Jon Baumunk, J.D., M.S.A. and Richard Weiss, Ph.D.

Michael, who goes by Rex, was a very engaged student, ready and willing to learn often challenging concepts of statistics, data mining, and consumer behavior, as well as take on a group project involving data mining and analytics. While Rex missed a few classes, Rex proved to be a dedicated and committed participant of the learning community and was always an active participant in small group discussions and seminar. Rex's work demonstrates insight and a rapidly growing and developing view of information and computerization in today's world.

Rex participated in a variety of class discussions and quizzes involving consumer behavior. They included topics such as marketing strategy; perceptual processes; memory and knowledge; motivation and behavioral learning; personality, self-concept, and psychographics; belief, attitude, and behavior formation and change; persuasive communications; decision processes; brand loyalty and satisfaction; situational and social influences; group, dyadic, and diffusion processes; popular culture; subcultures and demographics; neuroscience; and ethics. Rex turned in all eight quizzes. Rex's scores on the consumer behavior quizzes showed a good understanding of the material. This indicates Rex had carefully studied the assigned text covering consumer behavior. In addition, Rex was assigned a group project, which among other things, was designed to apply in a group setting Rex's teamwork and problem solving skills and knowledge of consumer behavior concepts. Rex's accomplishments demonstrate that Rex's

understanding of consumer behavior can be used to effectively work with businesses and other organizations.

While in seminar, Rex's contributions were insightful and heartfelt, and Rex seldom had trouble consolidating the material in a clear and cogent way. Rex certainly followed the discussion and not only consolidated Rex's point of view after hearing what was said by others, but typically added a new perspective to the discussion. This indicates a willingness to be open and grow Rex's ideas from the contributions of others. Rex's seminar papers indicate that Rex had read the assigned texts, captured the author's key ideas, and showed an understanding of the place and function of data mining and analytics in the context of Big Data.

In Rex's final group project, Rex's team divided the substantive work and presentation tasks well. Rex's contribution was very convincing and demonstrated that Rex could work effectively as a member of a team. Rex helped the team with Tableau, learning it and creating visually interesting and easy to use dashboards for the "client" organization, and worked well with teammates in building charts and graphs. In addition, Rex helped do research on comparable organizations, cleaned up data, contributed to weekly discussions of how to proceed, and helped delegate tasks for the team to complete by deadlines, including handing over a large chunk of work completed on researching the annual reports and finding more work to personally complete. By the end of the project, Rex became the primary liaison to the "client" organization, instructors, and the rest of the team, in order to plan the specifics for a smooth presentation. Rex also wrote the initial draft of the final report. Overall, Rex did very well on this assignment.

In summary, Rex clearly met the expectations for this program. Through Rex's demonstrated understanding of statistics and consumer behavior, well-written seminar papers on current issues in data science from different perspectives, work in various aspects of data mining and analytics, and contributions to the discussions, Rex has demonstrated significant progress toward accomplishing the learning objectives of this program.

Rex did very well in statistics and data mining. His attendance was good. He submitted all of the homework and lab assignments. In general, he did very well on them. He took all of the quizzes and did an excellent job on many of them.

SUGGESTED COURSE EQUIVALENCIES (in quarter hours) TOTAL: 16

4 - Statistics
4 - Data Mining
4 - Consumer Behavior
4 - Project: Data Mining and Analytics

Rexroth, Michael Anthony — A00326438
Last, First Middle — Student ID

January 2017 - March 2017: Unix System Administration

4 Credits

DESCRIPTION:
Faculty: Neal Nelson, Ph.D.

In the **Unix System Administration** Individual Learning Contract, students studied from the text *System Administration, 3rd Edition: Tools and Techniques for Linux and Unix Administration* by Æleen Frisch in order to learn the fundamentals of system administration in a Unix environment. Topics included fundamentals of files, processes, and devices; common system administration tools and techniques; startup and shutdown of processes; managing user accounts and account groups; setting up and securing a network; and automation of administrative tasks. The students studied 6 to 8 chapters of the text, wrote chapter summaries, engaged in setup and regular experimentation on one or more sandbox systems, and worked on one or more projects in the course of their study.

EVALUATION:
Written by: Neal Nelson, Ph.D.

Rex (Michael) successfully completed the Unix System Administration ILC. He wrote chapter summaries for each chapter he studied and kept detailed notes of his experimentation with the various system administration commands. He completed two projects and and informative project summary for each. His first project was the installation and setup of sandbox Linux in a virtual machine (several times as he learned about mistakes during his second project). In his second project, Rex set up and configured several user accounts and groups for a hypothetical project development team with various levels of permissions.

SUGGESTED COURSE EQUIVALENCIES (in quarter hours) TOTAL: 4

*4 - Unix System Administration

* indicates upper-division science credit

Rexroth, Michael Anthony — A00326438
Last, First Middle — Student ID

September 2016 - March 2017: Computability and Language Theory

28 Credits

DESCRIPTION:
Faculty: Neal Nelson, Ph.D.; Sheryl Shulman, Ph.D.; and Richard Weiss, Ph.D.

Computability and Language Theory is a two-quarter advanced computer science program, covering the study of Programming Language Design, Formal Languages, Artificial Intelligence, and Haskell (a high-level functional programming language). Students completed some or all of the threads of study described below.

Formal Languages I & II

During the fall, the class studied formal languages using the textbook *Languages and Machines* by Thomas Sudkamp. In the fall, we covered the definition of regular sets, grammars, languages and their properties, context free grammars, normal forms for context free grammars, and finite automata and their properties. Students were evaluated on weekly chapter assignments and two exams.

During the winter, the class continued in the same textbook. Winter topics included the Pumping lemma for both regular and context free languages, push-down automata, Turing machines, Turing computable functions, the Chomsky hierarchy, decision problems, undecidability, primitive recursive functions, and some complexity topics using Turing machines. Students were evaluated on weekly chapter assignments and two exams.

Artificial Intelligence & Machine Learning

In Artificial Intelligence & Machine Learning, students surveyed several basic topics and techniques. In the fall quarter Artificial Intelligence, the students studied chapters 1 – 5 and chapter 18 in the textbook *Artificial Intelligence: A Modern Approach* by Russell and Norvig. The reading, lectures, and labs covered intelligent agents, uninformed and informed search, stochastic search, adversarial search, perceptron learning, and an introduction to neural networks. Students were evaluated based on five quizzes, a midterm exam, a final exam, four programming projects, six written homework assignments, and class participation. The projects involved simulating and applying search techniques to a vacuum agent in a partially observable environment, solving the 8-puzzle, playing Othello, and perceptron learning.

In the winter quarter, Machine Learning students studied the basic concepts of reinforcement learning. They learned about dynamic programming, Monte Carlo methods, and temporal difference learning. The text was *Reinforcement Learning (2nd ed)* by Sutton and Barto. The reading, homework, and exams focused on the first 6 chapters. Students were evaluated on 6 quizzes, a midterm and final exams. There were six homework assignments, which included programs in Python and written questions from the text. This part of the program also included background work in Bayesian probability.

Functional Programming in Haskell I & II

In the fall, students studied the advanced functional programming concepts and techniques using the Haskell programming language. Students guided through studies from a variety of resources accessible via haskell.org. Weekly topics included higher-order functions, programming with lazy evaluation, modules, abstract data types, algebraic data types, overloading, polymorphism, and type classes. Students were evaluated based on 8 programming assignments and two exams.

During the winter, students studied higher order functional combinators leading to programming and proofs with type classes and type class instances for functors, applicative functors, monoids, and a

variety of monads (including the state monad). Students also studied polymorphic type inference and practiced making effective use of types to better understand Haskell programs. Students were again evaluated based on 8 programming assignments and two exams.

Programming Language Design and Project

This component studied the issues involved in programming language design. Using the textbook *Programming Languages: Principles and Paradigms* by Allen Tucker and Robert Noonan, the class studied lexical analysis, parsing, type systems, how we give meaning to a program written in a high level language, interpretation, and the impact design decisions have on the language and the interpretation process.

In the project component of the program, students worked on implementing a small C-like language called C-lite in work that was coordinated with the Tucker and Noonan textbook study of programming language principles. Students were expected to implement five phases of an interpreter for the language: lexical analysis, parsing, static type checking, type transformation, and semantic interpretation. They were evaluated based on successful progress on the implementation project.

During the winter, some students extended their small C-lite language to include function declarations and function calls (C-liteF). They extended their type system to allow for functions, and substantially modified their interpreter to allow for multiple function definitions and a more complex idea of scope. Students also studied the untyped lambda calculus, and the simply typed lambda calculus, implementing an evaluator for the lambda calculus and a type inference algorithm. Students were evaluated based on their implementation progress in their C-lite interpreter and the lambda calculus as well as four chapter assignments and two exams.

EVALUATION:
Written by: Neal Nelson, Ph.D.; Sherri Shulman, Ph.D.; and Richard Weiss, Ph.D.

Michael Rexroth successfully completed the following portions of the computer science and mathematics program Computability and Language Theory. Michael's accomplishments in individual parts of the program are presented in detail below.

Formal Languages I
Michael had good attendance and completed both exams. He turned in 8 out of 9 assignments. Both his exams were significantly above the class average. He is making very good progress on the material and is well prepared for more advanced work.

Formal Languages II
Michael took both exams and turned in all of the homework assignments. Both of his exams were very good, well above the class average. He showed a very good understanding of turing machines, decidability, the halting problem, reducibility proofs, and primitive recursion. He is well prepared for further advanced work.

Artificial Intelligence
Michael's performance in the fall quarter of Artificial Intelligence was very good. He did an excellent job on the final exam, and most of his other test grades were good. He completed all of the programming and written assignments.

Functional Programming in Haskell I
Michael's performance in the fall quarter was excellent. Michael submitted 6 (of 7) assignments, achieved good results on the first exam, and excellent results on the second exam. Michael successfully

acquired a very good understanding of the functional programming paradigm in general and Haskell programming in particular.

Functional Programming in Haskell II
Michael's performance in the winter quarter was very good on the more difficult programming and proofs with various type class instances of functor, applicative, and monad. Michael submitted 6 (of 8) homework assignments, achieved good results on the midterm exam, and good to very good results on the final exam. Michael achieved a good understanding of the theory, abstractions, and techniques of more advanced functional programming.

Programming Language Design I
Michael had good attendance, turned in all chapter assignments, and has completed the project through the parser. His midterm was strong, but his final was below the class average. He showed a good understanding of the structure of the lexer and parser, and many of the issues of language design in an imperative language, but was weak at presenting the formal description. He is prepared for more advanced work.

Programming Language Design II
Michael completed both exams: both were excellent, substantially above the class average. He completed the project through the type checker (the lexer, parser, type checker) although the type checker did not display the type map, making it difficult to confirm its correctness. He completed the lambda calculus workshop showing good understanding of call-by-need as well as call-by-value beta-reduction. His exam also showed a good understanding of the Y-combinator for handling recursion as well as a basic understanding of the type inference algorithm and unification.

SUGGESTED COURSE EQUIVALENCIES (in quarter hours) TOTAL: 28

*4 - Formal Languages I
*4 - Formal Languages II
*4 - Artificial Intelligence
*4 - Functional Programming in Haskell I
*4 - Functional Programming in Haskell II
*4 - Programming Language Design I
*4 - Programming Language Design II

* indicates upper-division science credit

Rexroth, Michael Anthony — A00326438
Last, First Middle — Student ID

September 2015 - June 2016: Computer Science Foundations

48 Credits

DESCRIPTION:
Faculty: Neal Nelson, Ph.D., Sherri Shulman, Ph.D., Richard Weiss, Ph.D., Adam King, Ph.D.

In the **Computer Science Foundations** program students engaged in intensive study of introductory and intermediate undergraduate computer science. Students were evaluated on the basis of attendance, participation in program activities, written work (including computer programs and executable logic models), and performance on examinations. Students completed some or all of the various parts of the program described below.

In **Introduction to Programming in Java I** the students studied the basics of programming using Java constructs (including methods, data types, conditionals, expressions, looping, recursion, and the basics of arrays). This background should give students the mental tools to quickly learn many other languages. Programming is all about learning how to express a process for doing something in a precise and unambiguous way. Programming can be a lot like solving a puzzle. And like a puzzle, when you get it right, you know it – that makes programming fun, challenging, and rewarding. There was no required programming text for this class, and one of the focuses of the class was to learn minimal constructs while maximizing problem solving time. This thread did not expect students to have any prior programming experience and only demanded a math background up through high school algebra. In this thread, students had seven programming labs, five quizzes, a midterm and a final.

In **Introduction to Programming in Java II** the students completed their study and exploration of basic Java constructs, including single and multi-dimensional arrays. Students studied how arrays functioned as objects, as well as how strings worked as objects. Students programmed a “hang-person” game that was their first “full” program. Students then went on to study the general object-oriented features of Java including classes, instances, addresses/pointers, the heap, error handling, garbage collection, constructors, encapsulation, inheritance, polymorphism, static vs. non-static, private vs. public, this, shadowing, and overriding. Students demonstrated creativity with a computer art lab, and they wrote a cellular-automata “game of life” simulator. In this thread, students had seven programming labs, five quizzes, a midterm and a final project.

Introduction to Functional Programming in Haskell introduced students to the basics of functional programming in Haskell. Using the online textbooks *Programming in Haskell* by Graham Hutton, *Learn you a Haskell for Great Good* by Miran Lipovaca, and lecture notes from the University of Virginia online resource by Nishant Shukla, the class covered Haskell algebraic types, recursion, strong typing, higher order functions and type classes. Evaluation was based on two exams and nine programming labs.

In **Digital Logic and Computer Organization** students studied the organization and logic of simple central processing units from the digital logic level to the instruction set architecture level. Topics included representation of data in binary and hex, combinational logic, sequential logic, register transfer level data path architecture, and the instruction set level architecture. Students attended weekly lectures on fundamental concepts and structures and studied from the preliminary textbook *Digital Logic and Computer Organization* by Neal Nelson to complete problem assignments and workshops. Students consolidated their understanding of computer organization in weekly labs by building and simulating a graduated series of logic modeling projects using a logic-modeling program (Logisim). The lab work concluded with a complete logic model and simulation of a very simple Von Neumann style computer. Students were evaluated on their attendance and participation in classroom learning activities, their weekly homework assignments, weekly labs, two examinations, and the final Computer Modeling Project.

Rexroth, Michael Anthony — A00326438
Last, First Middle — Student ID

Computer Architecture and Assembly Programming extended the study of machine organization from the register-transfer level architecture to the instruction set architecture and assembly language level of abstraction using the Java Virtual Machine (JVM) with the Jasmin JVM assembler. Students studied architecture and assembly language programming of the Java Virtual Machine language, covering Chapters 1 to 5 and 10 of the textbook *Computer Organization and Assembly Language* by Patrick Juola. Topics included JVM implementations of Java arrays, records, classes, objects and method calls as well as the usual low-level assembly programming. There were two examinations, weekly textbook assignments, and eight hands-on JVM assembly language programming laboratory assignments.

Microcontroller Applications was a project-oriented exploration of controller applications using the Arduino controller products and development software. Students learned how to plan, research, build, troubleshoot, demonstrate, and document small microcontroller applications. Students acquired individual learning skills by reading technical documentation and experimenting with the microcontroller hardware and software products. Students learned how to apply their knowledge by building and demonstrating their microcontroller projects. The class followed a few weeks of common introductory projects to build initial knowledge and experience and then were expected to complete one or more individual or group projects. Assessment was based on the students' weekly project planning and progress reports plus completed project demonstrations.

In **Discrete Mathematics I** students learned some of the standard topics in Discrete Mathematics, including propositional logic, predicate logic, Boolean algebra, methods of proof, elementary set theory, functions, summations, number theory and computational complexity. Using the textbook *Discrete Mathematics and Its Applications*, 7th Edition, by Kenneth H. Rosen, the class covered chapters 1-4. Students submitted weekly problem sets and took five quizzes and two exams.

In **Discrete Mathematics II** students continued with some of the standard topics in Discrete Mathematics, including methods of proof, computational complexity, combinatorics, probability, and graph theory. Using the textbook *Discrete Mathematics and Its Applications*, 7th Edition, by Kenneth H. Rosen, the class covered chapters 5-7 and parts of chapter 10. Students submitted weekly problem sets and took five quizzes and two exams.

In **Discrete Mathematics III** students continued with some of the standard topics in Discrete Mathematics, including graphs, trees, relations, finite state automata and grammars. Using the textbook *Discrete Mathematics and Its Applications*, 7th Edition, by Kenneth H. Rosen, the class covered chapters 9, 11 and 13. Students submitted weekly problem sets and took six quizzes and two exams.

The **Problem Solving Workshop** thread focused on improving students' problem solving skills, concentrating primarily on the discrete math concepts introduced in lecture with some related programming exercises. Students worked through weekly workshops. Students were evaluated on attendance, participation, and weekly workshops.

In the **Ethics and Technology Seminar** students read *Ethics and Technology* by Herman T. Tavani. This thread introduced students to issues and controversies that comprise the relatively new field of cyberethics. The textbook examines a wide range of cyberethics issues—from specific issues of moral responsibility to broader social and ethical concerns that affect each of us in our day-to-day lives. Recent developments in machine ethics prompted students to consider questions about conventional conceptions of autonomy and trust. By the end of the quarter students developed an appreciation for the many ethical issues they will be confronted with and will have developed some tools allowing them to make responsible decisions in the future. Students were evaluated based on seminar attendance and participation, as well as two short papers, and one longer comprehensive paper.

In the **Computational Thinking Seminar**, students deepened their understanding of computation in a broader context. Examining basic models of computation (e.g. Turing machines, string re-writing systems, cellular automata) students explored how computational systems use syntactic manipulations on representations to draw semantic conclusions. This is both the philosophical and engineering basis of computer science. From this vantage point, students questioned how computation could be viewed as a model of the brain and mind. At issue were how first, second, and third person perspectives come into play with respect to this analysis. Students then looked at some of the paradoxes and contradictions that emerge from computational and axiomatic systems. It is hoped that this exploration stretched students' conceptions of computation and touched upon deep questions regarding who and what we are as beings, how we connect to each other, and how we understand the world around us. This was a participation and attendance based thread. Students primarily worked in groups solving problems, discussing the questions raised, and looking at how protocols, prototypes, and context all abstract from the particular content.

EVALUATION:
Written by: Neal Nelson, Ph.D., Sherri Shulman, Ph.D., Richard Weiss, Ph.D., and Adam King, Ph.D.

Rex (Michael) successfully completed the following portions of the computer science and mathematics program Computer Science Foundations. Rex's accomplishments in individual parts of the program are detailed below.

Introduction to Programming in Java I and II

In Introduction to Programming, Rex's overall performance was excellent. He put in a consistently strong effort and his attendance was superb. He was always conscientious in handing in his assignments. His quizzes, midterm, and final were all excellent, and he did demonstrate a rock solid understanding of the material. At this point, I feel his programming skills are very strong and that he is in a good position to continue on in this thread.

Rex did an excellent job this quarter in Introduction to Programming II. His attendance was very good and he was a positive presence in class. He was also helpful to other classmates. Rex handed in very strong labs on-time. His quizzes were nearly perfect as was his midterm. His final quiz showed his knowledge of the basics of objects and very good problem solving skills. It was a pleasure to have Rex in class and I enjoyed his intellectual rigor. He also suggested thoughtful ways to improve this class and I appreciate that feedback. Rex is in excellent shape to continue with programming in the Spring.

Introduction to Functional Programming in Haskell

Michael completed both exams and 8 out of 9 labs. He had very good attendance. His work on exams was excellent. Both his midterm and final were significantly above the class average. His lab work was excellent; all the labs were well presented and mostly without errors. He has learned the significant basics of functional programming in Haskell and more and is well prepared for more advanced work.

Digital Logic and Computer Organization

Rex mastered all of the knowledge and skills covered in fall quarter Digital Logic and Computer Organization. Rex consistently attended the 9 lab sessions. He submitted all 9 (of 9) lab reports and all 8 (of 8) homework assignments. He achieved excellent results on the first exam and also excellent results on the second exam. He successfully completed the final comprehensive computer modeling project that integrated all of the lab work for the quarter. Rex demonstrated an excellent understanding of the organization of a Von Neumann processor architecture at the digital logic level.

Computer Architecture and Assembly Programming

Michael's performance in the winter quarter Computer Architecture was excellent. Michael consistently attended labs, submitted all 9 (of 9) homework assignments, and submitted all 9 (of 9) labs. Michael's first exam results were very good and second exam results were excellent. Michael demonstrated an excellent understanding of the assembly language programming level of computer architecture.

Microcontroller Applications

Rex successfully completed and demonstrated a series of Arduino microcontroller projects involving a variety of hardware and control software including LEDs, servos, and a clever hand-built capacitive touch sensing circuit. Rex did project work in a group of three students and demonstrated effective collaboration skills during the project planning and development. The project group did excellent research into capacitive touch sensing and built a delightful 7-key piano that worked simply with pencil-drawn keys on a sheet of paper (the graphite was the touch-sensitive medium). Rex consistently submitted the weekly planning and progress reports and completed a final report documenting the project work and indicating good progress in learning to research, design, build, and troubleshoot simple microcontroller applications.

Discrete Mathematics I, II, and III

Rex did an excellent job in Discrete Mathematics I. His midterm and final were both excellent. He also did an excellent job on the quizzes. He submitted all of the homework assignments and they were generally very good. His attendance was excellent, and he was engaged in class.

Rex did an excellent job in Discrete Mathematics II. His midterm and final were both excellent. He also did an excellent job on the quizzes. He submitted all of the homework assignments and they were generally very good. His attendance was excellent, and he was engaged in class.

Rex did very well in Discrete Mathematics III. His midterm and final were both very good. He also did an excellent job on most of the quizzes. He submitted all of the homework assignments and they were generally very good. His attendance was very good, and he was engaged in class.

Problem Solving Workshop

Michael attended problem solving sessions regularly and turned in seven of the eight problem solving workshops, demonstrating good progress.

Seminar: Ethics and Technology

Rex was a consistently valuable participant in seminar discussions. He was attentive to ideas and viewpoints brought forth in discussions and he demonstrated a good understanding of the ethical frameworks used in evaluating ethical issues. Rex turned in all 3 papers and weekly responses. His papers were all good. His papers often raised provocative points. His final paper was about the net neutrality. He mixed his views on network neutrality with his values such as freedom of speech, political participation, democratization, investment opportunities, encouraging innovation, etc. His essay would have been improved if he had concentrated on how network neutrality might achieve those goals, arguing from a consequentialist ethical view.

Seminar: Computational Thinking

In the seminar on Computational Thinking, Rex first explored how all computational systems use syntactic manipulations on representations to draw semantic conclusions. This is both the philosophical

Rexroth, Michael Anthony — A00326438
Last, First Middle — Student ID

and engineering basis of the field Rex has been studying this year. From this starting point, Rex was asked not to demonstrate select understanding of the content that was presented, but instead to use the content as a springboard to open-ended discussions, meditations, and group problem solving. It is hoped that this exploration stretched Rex's conceptions of computation and touched upon deep questions regarding who and what we are as beings, how we connect to each other, and how we understand the world around us. Rex regularly attended the seminar and was an active participant. Rex was a positive presence, often asking good questions and adding interesting insights. Rex has a strong analytical mind and it helped the class and his work groups. I also appreciate that Rex was very open-minded regarding the material being presented. Rex worked well with others and he was a pleasure to have in class.

SUGGESTED COURSE EQUIVALENCIES (in quarter hours) TOTAL: 48

5 - Introduction to Programming in Java I
5 - Introduction to Programming in Java II
4 - Introduction to Functional Programming in Haskell
4 - Digital Logic and Computer Organization
4 - Computer Architecture and Assembly Programming
4 - Microcontroller Applications
4 - Discrete Mathematics I
4 - Discrete Mathematics II
4 - Discrete Mathematics III
3 - Problem Solving Workshop
3 - Seminar: Ethics and Technology
4 - Seminar: Computational Thinking

EVERGREEN

The Evergreen State College • Olympia, WA 98505 • www.evergreen.edu

EVERGREEN TRANSCRIPT GUIDE

Accreditation: The Evergreen State College is fully accredited by the Northwest Commission on Colleges and Universities.

Degrees Awarded: The Evergreen State College awards the following degrees: Bachelor of Arts, Bachelor of Science, Master of Environmental Studies, Master of Public Administration and Master In Teaching. Degree awards are listed on the Record of Academic Achievement.

Educational Philosophy:

Our curriculum places high value on these modes of learning and teaching objectives:

- Interdisciplinary Learning
- Collaborative Learning
- Learning Across Significant Differences
- Personal Engagement
- Linking Theory with Practical Applications

Our expectations of Evergreen Graduates are that during their time at Evergreen they will:

- Articulate and assume responsibility for their own work
- Participate collaboratively and responsibly in our diverse society
- Communicate creatively and effectively
- Demonstrate integrative, independent, critical thinking
- Apply qualitative, quantitative and creative modes of inquiry appropriately to practical and theoretical problems across disciplines, and,
- As a culmination of their education, demonstrate depth, breadth and synthesis of learning and the ability to reflect on the personal and social significance of that learning.

Our students have the opportunity to participate in frequent, mutual evaluation of academic programs, faculty and students. In collaboration with faculty and advisors, students develop individual academic concentrations.

Academic Program

Modes of Learning: Evergreen's curriculum is primarily team-taught and interdisciplinary. Students may choose from among several modes of study:

- **Programs:** Faculty members from different disciplines work together with students on a unifying question or theme. Programs may be up to three quarters long.
- **Individual Learning Contract:** Working closely with a faculty member, a student may design a one-quarter-long, full-time or part-time research or creative project. The contract document outlines both the activities of the contract and the criteria for evaluation. Most students are at upper division standing.
- **Internship Learning Contract:** Internships provide opportunities for students to link theory and practice in areas related to their interests. These full- or part-time opportunities involve close supervision by a field supervisor and a faculty sponsor.
- **Courses:** Courses are 2-6 credit offerings centered on a specific theme or discipline.

The numerical and alpha characters listed as Course Reference Numbers designate modes of learning and are in a random order.

Evaluation and Credit Award:

Our transcript consists of narrative evaluations. Narrative evaluations tell a rich and detailed story of the multiple facets involved in a student's academic work. A close reading of the narratives and attention to the course equivalencies will provide extensive information about student's abilities and experiences. Students are not awarded credit for work considered not passing. Evergreen will not translate our narrative transcript into letter or numeric grades.

Transcript Structure and Contents: The Record of Academic Achievement summarizes credit awarded, expressed in quarter credit hours. Transcript materials are presented in inverse chronological order so that the most recent evaluation(s) appears first.

Credit is recorded by:

Quarter Credit Hours:	Fall 1979 to present
Evergreen Units:	1 Evergreen Unit (1971 through Summer 1973) equals 5 quarter credit hours
	1 Evergreen Unit (Fall 1973 through Summer 1979) equals 4 quarter credit hours

Each academic entry in the transcript is accompanied by (unless noted otherwise):

- The Program Description, Individual Contract or Internship Contract which explains learning objectives, activities and content of the program, course or contract.
- The Faculty Evaluation of Student Achievement provides information on specific work the student completed and about how well the student performed in the program or contract.
- The Student's Own Evaluation of Personal Achievement is a reflective document written by the student evaluating his or her learning experiences. Students are encouraged but not required to include these documents in their official transcript, unless specified by faculty.
- The Student's Summative Self Evaluation is an optional evaluation summarizing a student's education and may be included as a separate document or as a part of the student's final self- evaluation.

Transfer credit for Evergreen programs, courses and individual study should be awarded based upon a careful review of the transcript document including the course equivalencies which are designed to make it easier for others to clearly interpret our interdisciplinary curriculum. These course equivalencies can be found at the conclusion of each of the Faculty Evaluation of Student Achievement.

The college academic calendar consists of four-eleven week quarters. Refer to the college website (www.evergreen.edu) for specific dates.

This record is authentic and official when the Record of Academic Achievement page is marked and dated with the school seal.
All information contained herein is confidential and its release is governed by the Family Educational Rights and Privacy Act of 1974 as amended.
If, after a thorough review of this transcript, you still have questions, please contact Registration and Records: (360) 867-6180.