Teaching Portfolio
Sterling McLeod
PhD Student
GAANN Research Fellow
smcleo12 at uncc dot edu
Contents
Introduction .................................................................................................................. 3
Teaching Philosophy .................................................................................................... 3
Experience .................................................................................................................... 4
Instructor ....................................................................................................................... 4
  Topics in Computer Science: Robot Navigation (Spring 2016) .................................. 4
  Operating Systems and Networking (Fall 2013) ....................................................... 6
Teaching Assistantships ............................................................................................... 7
  Design and Analysis of Algorithms (Spring 2014) .................................................. 7
  Intelligent Robotics (Spring 2015) ........................................................................... 7
  Intelligent Systems (Fall 2015) ................................................................................. 8
Grading ......................................................................................................................... 8
GAANN ......................................................................................................................... 9
Presentations ................................................................................................................. 9
Seminars/Workshops .................................................................................................... 10
  Problem Based Learning to Promote Student Engagement .................................. 10
  Active Teaching with Technology ......................................................................... 10
  Using Concept Maps to Promote Critical Thinking ............................................ 10
  Engagement through Whole Group or Small Group Class Discussions ............ 11
  Best Practices for Structuring Group Work .......................................................... 12
Class Observations ...................................................................................................... 13
  Fall 2014: Introduction to Computer Science II, Celine Latulipe and Bruce Long. 13
  Spring 2015: Human-Computer Interfaces, Mary Lou Maher ............................. 13
Service ......................................................................................................................... 14
  President – CCI Grads 2015-2016 Academic Year ............................................. 14
  Vice Chair for Judging and Awards – UNC-Charlotte Graduate Research Symposium 2016 .......................................................... 14
  General Volunteer – UNC-Charlotte Graduate Research Symposium 2015 ....... 15
  Editor – History of Robotics Project ...................................................................... 15
  S.T.A.R.S. Outreach Program .............................................................................. 15
  CPCC Geekfest ...................................................................................................... 16
Appendix ....................................................................................................................... 16
Teaching Observations ................................................................................................. 16
  UNC-Charlotte Computer Science Department ..................................................... 16
Introduction
My teaching portfolio is a comprehensive document of all my activities that I feel have helped foster my teaching abilities. My teaching career is still very new as I am still a graduate student. Most of my experience is as a teaching assistant although I do have a small amount of experience as an instructor. I have been a Teaching Assistant for three courses and independently taught one course. These courses include both undergraduate and graduate courses with medium class sizes (20-50 students). I feel that I’ve gained a huge amount of experience from just these few semesters. As a result of these early experiences, I have become increasingly interested in active teaching techniques. I hope to independently teach another course next school year and implement a problem-based learning pedagogy to better engage students.

I am currently being trained in higher-education pedagogy through the GAANN Fellowship program. This training has opened doors for me to learn about active-teaching techniques, attend seminars on higher-education teaching, discuss pedagogical research with education professionals, and other various activities that help develop my teaching skills. I have tried to utilize this training to learn as much as possible about active teaching techniques and how to effectively incorporate them into any classroom.

In this portfolio one will find a brief statement on my teaching philosophy, an overview of my teaching career, and a list of activities that have honed my teaching abilities through the GAANN Fellowship program.

Teaching Philosophy
More than anything, I love to learn new things. This love drives me to teaching. I want to immerse myself in an environment based around learning and do whatever I can to help create a love of learning for others. Pursuing a career in education seems the most promising path for me to help as many people as possible experience this happiness.

Another driving factor behind my desire to teach is that I want people to be better than I am. Hindsight is 20/20 and thus it’s easy to look back and pinpoint what I should have focused my past time on or what opportunities mattered more than others. I know I cannot change the past so I don’t dwell in reflection for self-loathing – I dwell in reflection so that I can one day pass on this knowledge. I’m not content with just having a new generation being just as good as its predecessor. I think each new generation should be better: more knowledgeable, more efficient, more open-minded, and more creative. Even so early on in my career, I feel that I have a wealth of wisdom on how to become the best engineer possible. Teaching is the best avenue to reach out to the next generation of computer science students and help them to become the best that they can be.
I do not approach teaching haphazardly. I strive to teach with purpose and finesse. To do so, I have adopted four principles to guide my teaching:

1) **Be Active**
   The evidence supporting active teaching is too strong to ignore. Whether it is a discussion, hands-on examples, or some other kind of non-lecture activity, it is nearly always more effective to spend class time with an active learning technique than a traditional, “information dump” lecture. There are, of course, some instances where lectures are still effective and should be used accordingly, but I believe that active teaching techniques should be an item in any instructor’s repertoire of techniques.

2) **Be Relevant**
   Students want to learn useful things. They want to know that they’re spending their time on something that will actually help them in the future. Traditionally, instructors have put a huge focus on theory and leave it up to the real-world to make the connections between theory and practice. I want to break this practice. I think that instructors should aim to make those connections in the classroom so that students can hit the ground running when they start working on real-world problems.

3) **Be Clear**
   Miscommunication is a problem in any discipline, but it is especially hazardous in a classroom environment. Miscommunication can affect what students interpret as important details regarding the material, causing them to miss main points. Or worse, they may interpret something incorrectly and learn incorrect information. Unlearning information requires deliberate and repetitive cognitive efforts on the learner, which can be difficult for anyone. Instructors should be conscious of this phenomenon in the classroom and strike a balance between relinquishing control for active learning and maintaining clarity in how students’ learn material.

4) **Be Thorough**
   Teaching is a very time-consuming process. An instructor must spend time searching for the best way to present material, the best method to evaluate learning, and keep up with the latest status of the material they are teaching. By its nature, knowledge is always expanding and instructors must always keep up. On top of this, the act of teaching requires many hours of planning, evaluating techniques and student learning. I want to always maintain thoroughness and rigor when teaching because I know that my students’ success depends significantly on me.

**Experience**

**Instructor**

**Topics in Computer Science: Robot Navigation (Spring 2016)**

I proposed this class as an addition to UNC Charlotte’s Computer Science curriculum for undergraduates. Our curriculum offers an Intelligent Systems focus, but contained no classes on robotics. I taught this class as a flipped classroom with 1-2 lectures mixed in. This was my first experience teaching a flipped classroom so there were many lessons learned throughout the semester.

I wanted the students to learn both theory and practice. Flipped classrooms help promote practical skills by engaging students with practical problems every class throughout the semester. In
terms of practical skills, I wanted students to become familiar with a software framework called Robot Operating System (ROS) because ROS is listed as required knowledge for many new software engineering positions at robotics companies. I found that learning a software framework was just as hard, if not harder, for the students than learning the robotics theory. Most of the time spent on assignments was on learning ROS. I also wanted my students to get hands-on experience on programming robotics software. I surveyed possible robot platforms that would be suitable for the class. After communicating directly with a handful of companies, I submitted a report to our department chair that the Turtlebot 2 platform was the most feasible platform. The Computer Science department then invested in four Turtlebot 2 robot platforms for the course.

The most consistent lesson during the semester was that groups matter a lot. The size of groups, the ability for the students to communicate and work together, and prior experience of the students in each group were all factors that significantly affected the class. I originally made groups of three people. The groups were fine at the beginning, but as time went on they degenerated into two people working well and one person not participating. After the semester midterm, I changed the group size to four students per group. This seemed to be much more effective because the groups could separate into sub-groups of two students that worked together. This also let the groups work on different things in parallel which increased productivity and forced them to communicate as a group more often.

Students had a hard time communicating throughout the whole semester. I swapped some students between groups to help the situation, but I couldn’t solve all communication issues. I think that this is largely something that needs to be addressed much earlier than a student’s junior or senior year of college. In future flipped classrooms, I will find a resource on group communication for the students to go through and be graded on in case any of them have never worked in a group before.

Prior experience with technical skills was a huge factor in the class. It made the groups disproportionate in terms of each person’s ability to contribute. This was a problem because not every student was learning as much as the other group members. Since active teaching is based around students learning through applying skills in class, the students with the most experience would end up doing all of the work while other students mostly just copied their work. The students with little experience would attempt to contribute, but they would be severely outpaced by their groupmates.

A lack of prior experience also severely limited what the students could complete in a reasonable amount of time. I expected to have some troubles in designing reasonable assignments, but I was surprised to find a handful of assignments that the students could not complete. Further, I found that students did not have time to reflect on the assignments. They would be so concerned with simply finishing an assignment that they would move on as fast as possible to the next part of an assignment after they finished something. They were just copy and pasting lines of code around rather than trying to concentrate and think about why a certain line of code worked. This makes me feel that there is a lot of value in homework assignments. The students can have 1-2 weeks to complete something rather than 3 hours. That should give them plenty of time to reflect on why their assignment is correct. If I were to teach the class over again, I would give assignments meant to introduce them to ROS as homework assignments. After 1-2 introductory assignments, I would then switch to a flipped classroom style.

The most important takeaway from this course was that it is easy to have an unequal amount of learning among the students. Some will learn a lot and others will be struggling behind. Flipped classrooms don’t solve the issue. It is a problem in any classroom, flipped or traditional.
Operating Systems and Networking (Fall 2013)

This was a junior and senior level science in the computer science curriculum at UNC-Charlotte. It was the first class I ever taught. I made many mistakes and learned a lot. I taught this class in a traditional lecture-based style. That was all I knew at the time – I had not been exposed to alternative pedagogies yet.

Even so, I tried to make the class as hands-on as possible. I wanted the focus to be on doing rather than memorizing. The main portion of grading was through programming assignments that had the students implementing simulations of various algorithms covered in the class. I wanted to reinforce the material I lectured on in class through these assignments. The purpose of these assignments was solely for the students to build more detailed mental models of the material after using it to solve problems, such as scheduling a list of jobs to process.

This was my very first teaching experience. I had no experience independently teaching or even being a TA before I taught this class. Thus, I made mistakes. One of these mistakes occurred when I gave my first assignment. The problem was that the majority of the students were not clear on the details of the problem and what I expected in terms of a solution. Out of twenty-five students, only around ten made a C or above. Most of the submissions were not just incorrect, but conceptually wrong. They were confused about the material and how it applied to the problem. They misunderstood how the material applied to the problem which means they likely formed an incorrect mental model or no model at all due to confusion. I realized that I needed to be much more articulate when describing the mechanics of problems, how the material should be applied, and what I expect in a solution. Ever since that first assignment, I have strive to explicitly define those three qualities of an assignment when giving them to students. I find that any miscommunication about how material applies to problems to be detrimental to learning, especially if one learns something that is incorrect. Therefore, I believe an instructor should be careful and thorough when designing assignments to minimize any miscommunication about what is expected.

I gave a final project for the students to complete. I wanted the students to choose their own project topic so my only strict requirement was that the project was relevant to operating systems in some way. One group chose Data Center Virtualization as their topic – something we did not talk about in the class. Two of the students were obtaining a VCA-DCV (VMware Certified Associate - Data Center Virtualization) certification while they were taking the class and they wanted to use the class project as a way to further their understanding of the topic before their exams. They did all of the research themselves and only consulted me on what they should be submitting for a grade. They implemented a set of virtual machines using VMware Workstation (part of what they need to know for VCA-DCV). They wanted to test their VMs by installing the Cryptolocker virus on all of the machines. However they discovered that the Cryptolocker virus has protections against being installed in a controlled environment (e.g.
VMs). More details on their work can be found in my portfolio’s Appendix. I emphasized this project because 1) it went beyond the scope of the class in two areas (VMs and security), 2) it was directly applicable and relevant to the student’s careers, and 3) their results required them to reflect on the material (as is essential to learning under PBL techniques) to understand why they could not install Crytolocker.

Teaching Assistantships
In all of my TA positions, I have been responsible for the following:

- Grading
- Proctoring exams
- Holding office hours to answer questions

However, each class is different and thus each assistantship was a different experience. Below I give a brief description of each class, any additional responsibilities required by me, and what I took away from each experience.

Design and Analysis of Algorithms (Spring 2014)
This class is meant for Computer Science sophomores, but also contains seniors from other disciplines if they are obtaining a minor in Computer Science.

The class content ranged from sorting algorithms to dynamic programming. We (the professor and I) tried to focus the class time on lectures with heavy interaction with the students. Class time was mostly a mix of discussion and lecture.

In this class, I had a huge amount of input. The professor and I went over the material and went back and forth on what material should be covered and why. I was responsible for designing first-drafts of the tests and assignments. Another responsibility I had for this class was holding virtual study sessions on Google Hangouts. All students were invited and we covered various data structures and algorithms talked about in the class. These study sessions supplemented the main class material with even more amounts of discussion coupled with application skills.

Intelligent Robotics (Spring 2015)
This was the first graduate course to be involved in. The course was taught through a traditional style of in-class lectures and out-of-class assignments and projects.

We utilized the ABB RobotStudio software to help the students visualize and understand manipulator kinematics. I was responsible for essentially everything involving that software for the class, such as:

- Learning the software
- Giving a tutorial lecture on how to use the software
- Working with our school’s technical support group to get the software added to lab machine images
- Providing simple CAD models for the students to use
- Outlining steps in order to build and use their own models

The first two homework assignments (out of three) focus on computing forward and inverse kinematics. The students were able to use RobotStudio to visualize the kinematics and configurations of the robot for these two assignments. It also served to help them check their answers. The software was also helpful for them on a large final project, as opposed to having to program a large OpenGL source.

**Intelligent Systems (Fall 2015)**

This is a graduate-level course offered at UNC-Charlotte. It is taught in a similar way as the Intelligent Robotics class discussed above. The class begins with informed searching and CSP algorithms (A*, hill climbing, etc), moves into first-order logic and reasoning, and ends on conditional independence and Bayesian networks.

I gave several guest lectures in this class. The topics I lectured on were A*, online planning, and conditional independence.

**Grading**

I include Grading as an independent discussion point in my portfolio because I have become increasingly interested in effective grading techniques since I started teaching. For my first assignment, my inexperience reared its face when I tried to grade. I started grading the assignments case by case and assigned grades as a holistic measure. I tried to see what they did and didn’t do, and to give a grade based on the overall quality of the assignment. After a few of these, I noticed that I began to change my grading habits. I wasn’t taking off for things I had previously taken off points for or I was changing the amount of points I took off. I knew I couldn’t do that so I erased all the grades and started again. This time I made an objective list of specific things to take off for and how many points they would be worth. When I followed my new rubric, my grades tanked. After 10 students, no one had gotten above a 75. I started again with a tweaked rubric. I eventually went through 3 tweaked versions before I had a tolerable distribution of grades.

During my first few semesters grading, almost every assignment I graded without prior guidelines, this routine (minus the first part) has followed. My first draft of a rubric is always flawed and I have to revise several times before being content with the grades. This struggle to grade well has sparked an interest in grading for me. I have realized that grading is a complex task and warrants a considerable amount of attention and energy to be done well.

Grading has several difficult tasks. First, it must accurately measure various aspects of student learning. There are many commonly used approaches to this – multiple-choice questions, short answer, open-ended problem solving, oral presentations, etc. Different approaches target different aspects of learning, different levels of understanding (i.e. levels of
Bloom’s taxonomy), and are better suited for different disciplines. Finding an effective way to measure student learning taking into account the levels of learning to target, the type of content being learned, and the discipline is definitely not a trivial task.

Grading also has the difficult job of combining multivariate information into a single, discrete variable. Even with the assumption that there is a perfect measurement of various tasks students perform, the question of how to combine these measurements is not clear.

I feel that grading is an area of teaching that often gets overlooked due to being too time-consuming for professors or TAs to give adequate attention. When given the chance to design assignments or tests and their grading rubrics, I always strive to evaluate students fairly and as accurately as possible by targeting several cognitive levels and considering various aspects of the assignment, material, and class.

**GAANN**

In Fall 2014, I was awarded a fellowship by the Graduate Assistance in Areas of National Need (GAANN) program funded by the Department of Education. This fellowship is meant to train top graduate students in pedagogy for future careers in higher education. I receive this training in several ways. Firstly, I am given one-on-one training from a teaching mentor. I serve as a Teaching Assistant each semester for my mentor and help in designing assignments, giving guest lectures, grading, and managing a class. Secondly, I attend an education seminar course that is focused on recent literature in pedagogy. I am able to participate in discussions and present pedagogical work that I find interesting. Other opportunities for training under the GAANN Fellowship are observing classes of education researchers and attending education workshops.

**Presentations**

I have presented the following papers at the GAANN Teaching Seminar:

1. **Learning by Doing: An Empirical Study of Active Teaching Techniques**
   Jana Hackathorn, Erin D. Solomon, Kate L. Blankmeyer, Rachel E. Tennial, Amy M. Garczynski

2. **Everything You Need To Know About Developing A Grading Plan for Your Course (Well, Almost)**
   James O. Hammons & Janice R. Barnsley

3. **Why Problem-Based Learning Works: Theoretical Foundations**
   Rose M. Marra, David H. Jonassen, Betsy Palmer, Steve Luft
4. **Critical Thinking Goals, Outcomes, and Pedagogy in Senior Capstone Courses**  
   Marilyn Lockhart, Kenneth W. Borland Jr.

**Seminars/Workshops**

The Center for Teaching and Learning at UNC-Charlotte hosts workshops focused around pedagogy in higher education. The workshops are hosted by faculty and target an audience of instructors and teaching assistants. Below is a list of each workshop I have attended and what I learned from each one.

**Problem Based Learning to Promote Student Engagement**

I learned the difference between *problem-based* learning and *project-based* learning. The main difference is that project-based learning is usually composed of a few long-term learning exercises whereas problem-based learning can be done in small time intervals. So rather than having 1-4 projects throughout a class, you could implement a problem-based learning activity every week throughout a semester.

I also learned that students need to understand why PBL is useful. Many students feel that the teacher should be teaching them. They don't like the idea of paying so much money for tuition and then teaching themselves. If students don't understand why they're doing PBL, they are less motivated to work through problems and may feel bitter about the subject matter due to the “bad experience” they had in the class.

**Active Teaching with Technology**

I learned about some applications (some on Moodle, some not) that I had not used before. Mahara is an e-portfolio technology that allows students to organize and present their work. This would be good for students to show their knowledge by showing the progress of some project. SurveyShare allows students to answer small questions that can help the instructor see how well the class understands a topic. Saba Meeting is a web conferencing software that can be used within Moodle. This makes it easy to set up virtual office hours or virtual study-sessions.

**Using Concept Maps to Promote Critical Thinking**

The main take-away from this event was that there can be a massive difference in concept maps submitted by two different people and that's fine because the learning can still be solid in both maps. In fact, those differences can be utilized by putting students in groups to create concept maps. I really loved this idea because it causes students to discuss how ideas relate to one another and makes them defend their understanding of the material. Explaining yourself forces students to articulate their thoughts and the process of articulation can help solidify connections in learning for someone.
I think it would be interesting to have concept maps graded by other students. It would expose students to even more ideas of how to understand material and make them critically think about the material to determine what connects should be valid or not. That could be followed up by asking them all to create a new concept map after grading the other students' maps.

It would also be useful to have class discussions about specific concept maps. The basis for its usefulness is basically the same as making them grade concept maps, but the instructor can pick and choose which ones to discuss to highlight certain ideas.

Engagement through Whole Group or Small Group Class Discussions

Discussions are one of the key active teaching techniques. I want discussions to be a part of every class I teach in the future. Thus, I got a lot of value out of this workshop. Some of the significant lessons from this workshop were:

1) Do some kind of warm up activity before doing a full-on discussion

Student participation can be improved if you do some kind of warm-up before starting an in-depth discussion. The main purpose can be viewed as priming the students on the topic they are about to have a discussion on.

2) Use a circular or U-shaped seating setup

Humans naturally form circles when they are talking in groups. Providing that social geometry will help students feel more comfortable to talk.

3) Plan a question everyone can respond to

I thought this was a great idea and it pairs well with (1). Students will be more likely to participate if they feel that they have something useful to contribute. So in order to establish a positive attitude towards contributing, prompting them with an “easy” question will make feel good about participating. By “easy” I mean that the question should be something that either 1) should have an obvious answer to students or 2) be something that is easy to respond to because almost everyone has an opinion on.

4) Discuss how and why, not the facts

Discussions about facts are quite bland. In fact, talking about facts can resemble an information dump. Instead, focusing on “how” and “why” can make for much better discussions. Those type of questions are generally much more open-ended and can lead to many interesting points and ideas.

5) Hold students accountable for preparation

If students haven’t read material, they won’t have anything to contribute in a discussion. So finding a way to hold them accountable for preparing is essential to having involved
discussions with a large amount of participation and diverse ideas. An interesting idea that Dr. Maher does in her HCI class that I observed is this: the students are assigned reading to do outside of class. Then they must submit a handful of questions that could be used as quiz questions from that material. At the start of each class, they select 10 of the submitted questions and the students take a quiz from those questions. I think that is a fantastic way to hold students accountable. I'm sure there are other ways too. I think, though, that holding students accountable will require some graded activity. If students won't be graded on something, it's likely viewed as “extra” work that doesn't really matter. So in order to hold students accountable to prepare for discussions, some form of graded activity will have to be given.

Best Practices for Structuring Group Work

Group work is at the heart of active teaching techniques. Since I want to teach through active methods, this workshop was something I didn’t want to miss. I have highlighted some important points from this workshop.

1) **Do not let students self-select their own groups.**
   It is best to make the groups yourself as the instructor. There are several factors that can influence the success of a group. If the students know each other previously then other group members may feel left out. If there are huge differences in expertise levels in a group, that can lead to some students doing a majority of the work which means students will have significant differences in the learning process.

2) **Provide class time to get rid of scheduling issues**
   Students are busy. Most students are full-time and some students also work. It's unrealistic to think that groups will always be able to meet for a long enough time period to make significant progress on their work. Allotting time in class ensures that they will always have roughly an hour or more each week to all be together working.

3) **Have students anonymously evaluate each other**
   Groups are most successful when everyone participates. Groups should be able to inform other members that they need to participate more or put forth more of an effort. Many students will avoid the confrontation of discussing these issues so providing an anonymous avenue for them should help them express their feelings.

4) **Be transparent about why they are doing group work**
   Group work can be seen as a teacher being lazy. Students need to realize that group work can be extremely effective. They need to see the research that says they can learn better working amongst themselves than having an instructor lecture at them for 3 hours a week. If their perspective on group work is positive, then they should be much more willing to participate.

5) **Groups need to practice communication**
   As said earlier, a significant number of students will try to avoid confrontation rather than state their opinions about how the group is working. That is not a sign of a healthy group. Students need to be able to express their feelings on how each member is
performing. Being able to evaluate a group and each member in it is a very desirable quality in the real world. So we should try to get the students to practice communicating. This can be done by regular exercises where students go around and express how they feel the group is performing and articulate why each significant point about the group's progress is happening.

Class Observations
Fall 2014: Introduction to Computer Science II, Celine Latulipe and Bruce Long

The class was an interesting thing to observe. It seemed highly unstructured because there was much less control exerted by the instructors.

The students called the instructors by their first names. One of the instructors says she works hard to learn everyone's name. She believes that increases the students' sense of accountability. I thought this was a great point. Informality in a classroom makes things more fun and friendly. The students may become attached to the teacher and class which makes them not want to disappoint the instructor.

They did clicker quizzes every Friday for half of the class. This serves two purposes — it reinforces concepts with the students and gives them chances to work in groups.

The instructors told me that this style of teaching requires much more planning than in a traditional classroom. This is surprising because from an outside perspective it seems that the teachers do less than normal which would make you think it requires less planning. The material needed isn't less than in a normal class — it is the same amount or more of a different type of material. Rather than preparing hours of lectures, they prepare numerous quizzes, lab assignments, guided-programming labs, etc.

Spring 2015: Human-Computer Interfaces, Mary Lou Maher

I learned quite a lot from this event. The number-one observation was that the groups tended to work in small sub-groups of 2-4 people. Even though they could talk to (at most) nine others, most people stuck close to their neighbors and talked to the rest of the group only for confirmation and progress updates. Although the goal is probably to have all of the students from a group interacting as one big group, it seems the more efficient method is to have the groups separate into smaller groups that work on different aspects of the project, and then communicate their progress to the other sub-groups. It was interesting to me to see that happening naturally in a classroom environment.

The class was almost never quiet. As said earlier, there was over 100 students so someone was always talking. The teacher had to use a microphone to speak to the whole class. I found that surprising, but I quickly realized why it was needed. Without it, she never would
have spoken over all of the students. And I think they wanted to facilitate a relaxed classroom environment. Forcing everyone to be quiet for parts of the class would take away from that friendly feeling. I wonder if the flipped classroom style would still be effective with a smaller class size.

Some members of a group assumed leadership roles. When I asked the groups about their projects, generally one student would do all of the talking. I don't think the class required the groups to have leaders, but it just happened naturally. That group-dynamic was interesting because it means that different members of the group may have a different understanding or perspective on the project. I wonder how that would affect their understanding of the material they are meant to be learning.

I also noticed that the graduate students asked for help much more. I'm not sure why that was, but it stood out to me a lot.

Service

President – CCI Grads 2015-2016 Academic Year

I currently serve as the President of the graduate student organization, CCI Grads, which is for graduate students in the College of Computing and Informatics. The organization has been a part of our college for many years. As President of the group, I wear many hats. My responsibilities include (but are not limited to):

- Manage the planning of all the events
- Coordinate with the graduate student government
- Help with securing travel funding for our graduate students
- Serve as a spokesman for the graduate student population in our department

Vice Chair for Judging and Awards – UNC-Charlotte Graduate Research Symposium 2016

I am currently serving as the Vice Chair for Judging for this year’s GRS at UNC-Charlotte. My responsibilities include:

- Developing rubrics (papers, posters, and talks)
- Recruiting judges
- Providing instructions and training for judges
- Overseeing tabulation of scores
- Overseeing award generation
- Implementing a thank-you process for judges

Along with these judging-specific responsibilities, I am part of the main GRS committee. As part of that committee, I help find solutions for the problems and vote on event-planning decisions.
General Volunteer – UNC-Charlotte Graduate Research Symposium 2015

I was a volunteer for the UNC-Charlotte GRS in 2015. The event is a small symposium for graduate students of NC schools to come and present their work. Paper presentations and poster sessions are held all day. Faculty from UNC-Charlotte serve as judges and score each person. At the end of the day, the winners are presented with a prize.

In the morning, I ran the poster sessions for the Science and Engineering research posters in the morning. I managed the timer, informed (yelled at) the presenters and judges of five and one minute warnings, when the judges were to switch, and when to begin each session.

That experience was a lesson in improvising. I did not expect to be the one managing the session. But not all of the volunteers showed up so one of us that were there had to fill the role of doing the actual timing and stopping/starting all of the sessions. I'm generally a reserved person so I felt uncomfortable taking control of a room with my voice so often. I gave it a legitimate effort though and I think it went very well.

In the afternoon, I was a floating guide. I welcomed visitors and showed them to the rooms they were assigned to. This was, again, a lesson in improvisation. I was the only floating guide to show up so I was all over the place trying to direct people. The rooms were actually very difficult to find so I had to walk people all the way there, leaving other people behind. We eventually printed out more signs and I moved to the second floor permanently. Visitors could make their way to the second floor easily and I could spend less time showing them to their rooms. Everything went smoothly after that and I could relax some.

Editor – History of Robotics Project

I was an editor for the History of Robotics project led by Indiana University. The project is a compilation of interviews from researchers that contributed to seminal works throughout the last few decades. I edited transcripts from three of the interviews. Most of the transcribers were not from a technical background. I went through the documents to correct jargon that was transcribed incorrectly and name-checked all the references.

S.T.A.R.S. Outreach Program

I was a member of the S.T.A.R.S. organization for three years 2010-2013. Most of my activities for STARS revolved around designing and creating modules that were meant to teach middle school students about technology. The module topics included website design, GameMaker, robotics, and participatory sensing.
CPCC Geekfest

The CPCC Geekfest is an all-day event to showcase engineering programs and projects of local organizations. I helped run a booth for UNC-Charlotte’s College of Computing and Informatics each year for 2010-2014. The goal was to reach out to students who wanted information about transferring or to high school students interested in UNC-Charlotte.

Appendix

Teaching Observations

UNC-Charlotte Computer Science Department

During the Spring 2016 semester, the Computer Science department of UNC-Charlotte observed one of the classes in my topics course titled “Robot Navigation”. Two experienced faculty sat in on a class and observed my teaching style. The class was a flipped-traditional hybrid and I gave a lecture during the observed class period on motion models for mobile robots. The material started with an overview of mobile robots, briefly explained control theory, and then went into motion models for differential drive robots. Next, we discussed car-like robots and ended the lecture on non-holonomic constraints.

At the end of class, the observers asked me to leave the room and asked the students various questions about the class and my teaching methods. A Performance Appraisal Form was submitted to the department chair and then I met with the observers to go over the form. The performance review can be seen over the next three pages.
### Instructional Performance Appraisal Form

**Evaluators:** Drs. Ken Chen and Aidong Lu  
**Instructor:** Sterling McLeod

**Date:** April 1, 2016  
**Time:** 9:30am  
**Class:** ITCS3050-2  
**No. Students Present:** 11

**INSTRUCTIONS**
- The evaluator is to assess the instructor’s performance with respect to the major functions of teaching listed below.
- The evaluator must add pertinent comments at the end of each major function for which an assessment of “Sometimes” or “Rarely” is given.
- The instructor is provided an opportunity to react to the evaluator’s ratings and comments.
- The evaluator and the instructor must discuss the results of the appraisal and any recommended actions pertinent to it.
- The instructor and the evaluator must sign the form in the assigned spaces.
- The form must be filed in the instructor’s personnel folder.

Check the appropriate boxes in categories 1 – 5 on this form.

#### 1. Classroom Management

<table>
<thead>
<tr>
<th>Category</th>
<th>Often</th>
<th>Some</th>
<th>Rarely</th>
<th>N/A</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Instructor makes good use of available time for teaching and keeps students on task.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2. Instructor stops inappropriate behavior promptly and consistently, yet maintains the dignity of the student.</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Instructor’s expectations are clearly explained when giving assignments and other directives.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Comments:**

The instructor started with explaining homework and continued to the lecture on differential drive.

Students have confirmed that the instructor’s expectations are clearly explained for homework, projects, and exams.

#### 2. Instructional Monitoring

<table>
<thead>
<tr>
<th>Category</th>
<th>Often</th>
<th>Some</th>
<th>Rarely</th>
<th>N/A</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Instructor poses questions clearly.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Instructor uses student responses to adjust teaching as necessary.</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Comments:**

The instructor posed several questions clearly during the lecture.

#### 3. Instructional Presentation
<table>
<thead>
<tr>
<th>1. Instructor links instructional activities to prior learning.</th>
<th><strong>Often</strong></th>
<th><strong>Some</strong></th>
<th><strong>Rarely</strong></th>
<th><strong>NA</strong></th>
<th><strong>Yes</strong></th>
<th><strong>No</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Students appear to comprehend what instructor is saying.</td>
<td><strong>X</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Instructor provides relevant examples and demonstrations to illustrate concepts.</td>
<td></td>
<td><strong>X</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Instructor asks appropriate levels of questions that students handle with a high rate of success.</td>
<td></td>
<td></td>
<td><strong>X</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Instructor conducts the lesson or instructional activity at an appropriate pace, slowing presentations when necessary for student understanding but avoiding unnecessary slowdowns.</td>
<td></td>
<td></td>
<td></td>
<td><strong>X</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Instructor summarizes key points.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>X</strong></td>
<td></td>
</tr>
<tr>
<td>7. Instructor presents content in a logical manner using smooth transitions from one topic to another.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>X</strong></td>
</tr>
<tr>
<td>8. Instructor encourages creativity and critical thinking in problem-solving.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>X</strong></td>
</tr>
</tbody>
</table>

**Comments:**

The lecture is extensive on math. The instructor has done a good job on explaining the math and providing examples. The setting of mixed lectures and in-class projects seems to be very suitable for this class and attracts the interests of students.

4. **Instructional Feedback**

<table>
<thead>
<tr>
<th>1. Instructor provides sustaining feedback after an incorrect response by probing, rephrasing the question, giving a clue, or allowing more time.</th>
<th><strong>Often</strong></th>
<th><strong>Some</strong></th>
<th><strong>Rarely</strong></th>
<th><strong>NA</strong></th>
<th><strong>Yes</strong></th>
<th><strong>No</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Instructor treats all students in a fair and equitable manner.</td>
<td></td>
<td><strong>X</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Comments:**
### Post-Observation Interview with Students

<table>
<thead>
<tr>
<th></th>
<th>Often</th>
<th>Some</th>
<th>Rarely</th>
<th>N/A</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Instructor has established a set of procedures that govern the handling of routine administrative matters.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2. Instructor returns graded material in a timely manner.</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Instructor regularly provides useful feedback on out-of-class work.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>4. Instructor understands overall concepts.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5. Instructor creates learning activities that make subject matter understandable.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>6. Class assignments are reasonable and encourage learning.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>7. Out-of-class assignments are clearly set forth.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>8. Students know what is expected of them.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>9. The instructor uses strategies that encourage critical thinking and problem solving.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>10. Instructor uses in-class activities during the semester.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>11. Test questions are appropriate for the material covered in class.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>12. Instructor is readily available to students outside of class.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>13. The class environment encourages students to ask questions.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Overall Performance (check one):**

- Superior [ ]
- Above Average [x]
- Average [ ]
- Below Average [ ]
- Unsatisfactory [ ]

**Evaluator's Summary Comments:**

This is an attractive class with in-class project settings. The instructor is able to explain extensive math and lead students through hands-on programming projects. The class is generally well handled with positive students' feedbacks. We suggest the instructor to consider more vivid examples such as videos in the lectures.

**Instructor's Reactions to Evaluation:**


**Evaluator's Signature:** [Sign]

**Date:** Apr. 13, 2016

**Instructor's Signature:** [Sign]

**Date:** 04/13/2016

*Instructor's signature indicates only that the written evaluation has been discussed.*
What is Data Center Virtualization:

Data center virtualization, from a technical perspective, is the conversion of hardware devices into software resources. There are cases where you might have many different types of servers because each server has its own operating system, applications, and data. They all have something in common. They all consume power. Physical devices that use power also give off heat. We need to get rid of this heat by cooling the data center. Heat takes power and power costs money. Virtualization fixes this issue.

We can separate a server’s operating system and applications from the underlying hardware by presenting virtual hardware that does the same job. Virtual hardware is actually just software running within a hypervisor. A hypervisor provides each virtual machine with access to underlying resources. This enables virtual machines to share access to the same hardware without even being aware of each other. Through virtualization, we can consolidate our servers into fewer pieces of hardware. By reducing the number of physical devices in the data center, we can save money on power and cooling, but we can also save money on capital expenses and management costs. Some advantages of data center virtualization are fewer servers are needed, there is less power consumption, and it saves money on coding.

We need data center virtualization today more than ever because IT departments around the world are being pressured to reduce their costs while at the same time improving system availability and agility. This can create some unique challenges for an organization because cost reductions are usually at odds with these system-wide improvements. These challenges can be broken down into four categories. The four categories are availability, scalability, optimization, and management. An example of availability is when you need to minimize the downtime caused by hardware failures. Another example would be when you need the ability to perform hardware maintenance during normal business hours, and you do not want to impact your applications. An example of scalability is when you are interested in virtualization, but do not have a shared storage infrastructure. Another example would be occasionally needing the ability to add more CPU’s and memory to your servers without running into downtime. An example of optimization is when you have several servers that have very large disk drives, but most of the space is empty. Optimization is when you want to make use of this space more efficiently. Another example would be when you need detailed performance reports and proactive notifications to help you identify trending and immediate performance problems. An example of management would be if you were a one person IT team and you need a centralized and simple way to manage your environment. Another example would be if your server environment is complex and decentralized. You need a central management solution that will help you keep track of everything worldwide. These are just some of the variety of challenges in IT world. Data center virtualization helps in overcoming these challenges.
The core of virtualization is the virtual machine. From the outside view virtual machines look and behave just like physical servers. Even though it may be hard to distinguish a virtual machine from a server, there are some differences that make them unique and very useful in the data center. First, the hardware of a virtual machine really is just software. This gives you a number of advantages. It gives you the ability to replace and upgrade components of the virtual hardware on the fly. It allows you to add new hardware devices such as network cards and processors without rebooting the virtual machine. It can also help reduce your downtime because you will not need to reboot your virtual machine every time you wish to upgrade its capabilities.

Just like a physical server, each virtual machine has its own operating system and dedicated hardware. It is virtual hardware though, so you will not be using up any additional space in your data center. Virtual machines are just as stable and dependable as their physical counterparts. Because the virtual hardware is designed to behave in exactly the same way as physical hardware, there is no need to modify the operating system to support virtualization. This is important because you will not be introducing any special challenges to the operating system that may affect the virtual machines stability. The administrator is in complete control of the virtual machine and its hardware. You install the operating system and applications on virtual machines almost the same way that you install these programs on a server. You can use CDROMs or even ISO images. Through the vSphere Web Client, you can remotely control a virtual machine, install software, and even perform more advanced operations such as cloning or snapshots.

Virtual machines can be used to host any application from file servers, database servers, email servers, and even high performance application servers. Organizations choose to use virtual machines for a number of reasons. Some companies are interested in consolidating lightly used servers to conserve space and power in their data center. These workloads are ideal for virtualization because you can often place many virtual machines on a single physical host. Other times organizations look at virtualization as a way to increase their availability, whether as protection against common hardware failures or complete site level disasters. Virtual machines are easy to move, copy, and restore. This makes disaster recovery very simple. Other organizations look to virtualization as a way to provision new servers quickly because it is possible to create and deploy new virtual machines in just minutes. Virtualization does not hinder your performance. It enhances your capabilities.

Your virtual machine can do things that physical machines cannot accomplish. If you try and apply a patch to a server and it causes data loss or some other system instability then you will have one upset manager and have lost evenings and weekends while recovering the server from a backup. This dilemma can be avoided in the virtual world with just a few clicks of a mouse. A feature called snapshots allows you to capture the entire running state of a virtual machine. Snapshots are fast and easy to use. If a patch or update does something that is unexpected, you can easily revert back to that point in time without the massive investment of
a full system restore. Snapshots can help you get out of tough situation but they are not designed to be used as daily backups. Using a bare-metal backup, you can take a complete imaged based backup of a virtual machine while it is running. It allows you to restore the entire server without the need to load a recovery operating system or perform file by file restores. All virtual machines can be protected in this way. This saves time and money during restores and you can restore individual files. The reason that virtual machines are easy to backup is because they are built on virtual hardware. Virtual machines have little to no dependency on physical hardware, so you are free to restore them to a device of your choosing. You do not need to find the exact hardware match. This makes testing your disaster recovery plan much easier. If you are backing up or backing out of a problem, virtual machines can assist you. Virtual machines allow for the hot add of virtual hardware. This means that if your virtual machines operating system supports this you can increase the capabilities of a virtual machine without restarting it. These features are not just technical advantages. They directly support your ability to help a business to remain more agile. By simplifying the backup and recovery processes, you can assure management that your disaster recovery plan is not just for show. You can actually prove that it works without affecting your production systems.

The hypervisor job is to provide virtual machines with their virtual hardware as well as provide the appropriate share of physical resources to the virtual machine as defined by you, the virtual machines administrator. You control how each virtual machine will perform by defining how much of a given resource will be allocated to the virtual machine. The hypervisor executes based on your designed instructions. There are two types of hypervisors. The type 1 hypervisors are called bare metal because they are installed on a physical device without an operating system. They perform the functions of the operating system as well as the resource management features. VMware’s ESXi is an example of a Type 1 hypervisor. Type 2 hypervisors are called hosted because they operate as an application on top of a pre-existing operating system. VMware Workstation is an example of a Type 2 hypervisor. Data Center Virtualization is typically performed using Type 1 hypervisors because there is less dependency on another operating system. There is also less resource overhead because the Type 1 hypervisor can perform both roles in a single piece of software.

Virtual machines take advantage of virtual networking. Your ESXi host can leverage storage technologies such as NFS, iSCSI, and FCoE. These all rely on traditional Ethernet networking. ESXi hosts are managed by the vCenter server and leverage the network for features such as vMotion, HA, and DRS. Your virtual machines use the network to communicate with each other and to the physical world. All of these require the use of virtual networking.

Virtual networking is very similar to physical networking. Each virtual machine and ESXi host on the network has an address and a virtual network card. These virtual network cards are connected to virtual Ethernet switches. VMware took the programming and logic of a physical Ethernet switch and turned it into a virtual device. It is very similar to how a virtual machine is
built out of virtual motherboards, network cards, and storage controllers. It is just software that behaves like hardware. The rules of networking are still enforced. Virtual machines must have the proper addresses and protocols, but it can be handled with a lot less cables. You can use these virtual switches to attach your virtual machines to the physical network. You can also create isolated networks to be used during testing and development. With virtual networking you have the same flexibility that server virtualization offers. There are different types of virtual switched as you can choose the one that is best for your business needs.

Just like physical switches, virtual switches come in different forms, and they each have a different feature set. vSphere supports two main categories of virtual switches. It supports the standard switch and the distributed switch. Both switches help you reduce network clutter by reducing the number of physical network cables plugged into your ESXi hosts. Each ESXi host comes with a standard switch that provides basic connectivity and management features. The distributed switch expands upon that model by providing a central interface to manage the different connections and features found in the virtual switches. The distributed switch is able to provide more features as a result of this centralized management approach.

Virtual networking can be as simple or as complex as you need it to be. VLANs provide logical separation of your network traffic, and are often used to isolate different sub networks, such as a test or restore network. Traffic shaping is a feature that allows you to restrict the inbound and outbound network bandwidth of a group of virtual machines. This helps reduce congestion in your virtual network. Port mirroring gives you the ability to monitor a virtual machine’s traffic for troubleshooting or intrusion prevention. It allows you to capture all the traffic sent to or from a virtual machine for later inspection. Quality of Service (QoS) and DSCP are networking standards that allow network switched to prioritize certain network traffic over others. An example would be if you wanted to prioritize the voice traffic from a call manager server to improve performance. NetFlow is a network monitoring tool that allows you to determine your top talkers on the network as well as other metadata about the communication that occur on your network. CDP and LLDP are discovery protocols used to identify neighboring physical network switches. They can be used to help discover and help troubleshoot errors.

**Applying Data Center Virtualization to the Real World:**

Veem Autolab was produced by Veem, a VM backup solution, in order to test products, and aid in upgrading business environments. The Lab is also used to aid students in the VCP5 certification course. In order to run this lab, software that was available from VMware and Microsoft were acquired including:

- Workstation 10
- ESXi 5.0
- vCenter 5.0
- VMware PowerCLI (Windows PowerShell Snapin)
- VMware Tools for Windows
- Windows Server 2008 R2 x64
- Windows Server 2003 x32 for nested ESXi Hosts

A computer with the minimum requirements detailed in the lab guide was also required to run the lab. The computer used to host, though outdated, was a prime candidate for the lab.

- Intel Xeon 5130 2 Core @ 2GHz
- 16GB Server Ram
- Windows 8 x64
- 500gb RAID 0 for performance.

The host was also required to have Virtualization Support imbedded in the CPU. This will allow the user to run x64 operating systems natively in Workstation.

With all the materials obtained, planning the environment was priority. The group needed to have a visual representation of how the environment would look during the process of installation. The figure below was drawn using Visio in order to do so.

![Figure 1](image)

**Figure 1**

Each host has a purpose in the environment:

- NAS: Networked Storage for All VMs using FreeNAS
  - 512 MB RAM
- DC: Domain Controller, utilizing Active Directory login, and
  - 1GB RAM
- VCenter: vCenter server, for management of ESXi hosts on the network
  - 3GB RAM
- Host 1, Host 2: Nested(within a virtual environment) ESXI Server for testing vCenter, and other VMs
  - 4GB RAM each

With the figure in play, the host up and running with Windows 8, and VMware Workstation installed, it was time to configure Workstation 10 to run our VMs. In order to do so we needed to reserve host ram for Workstation’s virtual machines, and configure networking. The steps below show how and what we used in our environment.

1. Set 8GB of RAM aside for VMs, and set to Fit All VMs into reserved host memory.

![Figure 2](image)

2. Set network to VMnet3, Host Only, Use Local DHCP Service, set subnet to 192.168.199.0, and mask to 255.255.255.0
3. Open required pre-built VMs in workstation, and power on NAS.

A NAS was used to have network storage for the multiple servers that needed to share data, scripts, and installers. With the NAS powered on, and the static IP set, we were able to access the windows share locally on the Windows 8 Host using \192.168.199.7\Build

In this folder, we were able to place the files required to build the DC, vCenter Server, and ESXi Hosts. It also came with multiple scripts provided by vBrownBag (VMw automatization specialists), that configured the hosts on power up. To tweak this entire package for our goal, we used the automate.ini script provided. The script allowed us to turn the VMware View installer off, which is a desktop virtualization solution, which was not needed for this project. The vCenter install was also set to 5, and it was time to build the DC.

The DC install started with mounting the Windows 2008 R2 ISO to the host, and powering on the machine. With the install started, this is where automation has proven its worth, 60 minutes later the installation was complete. The last step was to verify the server, with the
With green lights, we were ready to proceed with the vCenter Server. The vCenter server was set up the same way as the DC, with more commands for post install. The figure below shows...
the post install script running, and then verifying the installation of the server.
Figure 6
The next stage is to add the ESXi hosts to Workstation. ESXi is extremely easy to install as well even without automation. The hosts were then powered on, and the 5.0 install was selected. Once powered on, the ESXi host was verified by the static addresses 192.168.199.11, and 192.168.199.12. The figure below can be used as a reference for the static addresses that were assigned to the host.

![Image of ESXi host verification](image.png)

*Figure 7*
With the two hosts configured and verified, the vCenter server is really useless without something to manage. This is where the script in Figure 5 comes in. This script populates ESXi into vCenter for management, and installs Windows Server 2003 x32 for future management. After the script ran, and verified, the main test to see if we were complete was to verify the hosts were in vCenter.

Figure 8
With basic servers up and running, it was now time to shut down the hosts for another day. With the script shown in figure 5, the lab can be shut down using command 6. This command shuts down the entire system, except for the NAS. With this complete (shown in the figure below) the lab was ready for future testing.

Figure 9
Using the Lab in A Real World Scenario:

Cryptolocker is a type of Ransomware that uses a Trojan horse approach to attack its victims. Cryptolocker usually starts off as an email attached zip file that seems to be sent by a legitimate source. Once the zip file is downloaded and extracted there is an executable program that is disguised as another type of file, such as a Microsoft word or PDF file. Once the file is clicked it starts to run and installs itself in the “Documents and Settings” folder under an unassuming name. It also adds a registry key that causes the program to run on startup. It then sets launch points in the %appdata%, %localappdata%, and Recycle Bin locations. From there it connects to a remote server that sends back a commercial-grade 2048-bit RSA public key. The program then uses the public key to encrypt files across the computer and connected network drives. Cryptolocker only encrypts files with certain extensions such as but not limited to Microsoft office documents, CAD files, and pictures. Once Cryptolocker is done encrypting files it will display a message similar to the one below explaining the Ransomware and its intentions.

![Cryptolocker Message](image-url)
In order for the user to receive a matching RSA private key to decrypt their files they must pay a fee. If the user does not pay the fee within a certain amount of time then the private key will be destroyed and the encrypted files will never be restored. If the user decides to pay the fee Cryptolocker will ask the user to pay using the Bitcoin digital payment network. As a side note the fee has risen considerably since Cryptolocker’s inception earlier in 2013. The two roadblocks that make Cryptolocker very hard to deal with are the RSA Encryption and the use of Bitcoin as a form of payment. The 2048-bit RSA encryption that is used by Cryptolocker is the largest of the RSA numbers and it is estimated that it may not be factorizable for many years to come, unless substantial technological advances are made. Not only does Cryptolocker use an encryption that is near impossible to break, but it also uses a payment network that is pretty much impossible to trace. Since Bitcoin’s are not considered legal tender it does not fall under the same rules and laws as other forms of currency. In turn since there is no regulatory body for them it’s hard to trace them to any real life identity. This makes Bitcoin’s the perfect form of payment for Cryptolocker. Both the RSA encryption and Bitcoin payment system make Cryptolocker very dangerous because, you cannot get rid of the Malware and you cannot find the people responsible for deploying and profiting off of it.

Once Cryptolocker is deployed onto a computer and its shared storage network there is no way to stop it completely. The best way to combat Cryptolocker is to prevent it from happening in the first place. The best way to prevent Cryptolocker is to use a program called CryptoPrevent. CryptoPrevent implants group policy objects into the registry in order to block certain executables in certain locations from running. The number of rules created by CryptoPrevent is somewhere north of 150 and. It is very easy to use and allows a good amount of customization to ensure you’re protected in the way that suits you best. Below is a screen of CryptoPrevent’s user interface:
To apply the real world test we started by downloading, and running Cryptolocker securely in a Mac VMware Fusion Virtual Desktop running Windows 8, and having no network attached storage, or accessibility to networks (virtual or otherwise). The Cryptolocker did not successfully install the VM as it has, like other Malware, anti-VM and anti-debugging capabilities. We assume it has these capabilities enabled for a couple of reasons. First we believe that the makers of the virus know that infecting a controlled sandbox environment is useless and they want to make it as hard as possible to study the virus in a controlled environment. Secondly we believe that the authors of the Malware would want the user to open it inside a VM and it not execute so the user would think it would be okay and safe enough to run on their host and it execute there. This bodes well for the security of Virtual Machines as they are not targeted by viruses such as Cryptolocker. Future testing will be taken outside of this project, in order to analyze the virus.