Lean Implementations in the University Classroom and Curriculum Design

by
Dustin Diep

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Industrial Engineering
(Honors Scholar)

Honors Baccalaureate of Science in Manufacturing Engineering
(Honors Scholar)

Presented October 16, 2018
Commencement June 2020

AN ABSTRACT OF THE THESIS OF

Dustin Sydney Diep for the degree of Honors Baccalaureate of Science in Industrial Engineering and Honors Baccalaureate of Science in Manufacturing Engineering presented on _______. Title: Lean Implementations in the University Classroom and Curriculum Design

Abstract approved:____

Chinweike I. Eseonu

The Lean process improvement philosophy has been proposed as a method for improving components of Higher Education (HE), including student learning outcomes; however, current lean applications to HE have focused on administrative operations or general institutional interventions. Comparatively little work has been performed applying lean to teaching. This work fills this gap in the lean HE literature. Specifically, we investigate how lean can be applied to the university classroom. We used a literature review to assess lean in HE and conducted a case study in an introductory engineering computing class. The current state of the class is measured in terms of course grade performance, curriculum materials, and student feedback and analyzed for root causes of potential problems; improvements are then recommended for implementation based on lean principles and practices. The methodology presented in this work may be applied for educators looking to implement continuous improvement practices in their teaching.

Key Words: Lean, Higher Education

Corresponding e-mail address: diepd@oregonstate.edu

Lean Implementations in the University Classroom and Curriculum Design

by
Dustin Diep

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Industrial Engineering
(Honors Scholar)

Honors Baccalaureate of Science in Manufacturing Engineering
(Honors Scholar)

Presented October 16, 2018
Commencement June 2020

Honors Baccalaureate of Science in Industrial Engineering and Honors Baccalaureate of Science in Manufacturing Engineering project of Dustin Diep presented on October 16, 2018.

APPROVED:

____

Chinweike I. Eseonu, Mentor, representing School of Mechanical, Industrial, and Manufacturing Engineering

____

Cindy Grimm, Committee Member, representing School of Mechanical, Industrial, and Manufacturing Engineering

____

Michael O’Malley, Committee Member, representing College of Education

____

Toni Doolen, Dean, Oregon State University Honors College

____

Dustin Sydney Diep, Author

Table of Contents

Introduction

Higher Education's (HE) one hundred and fifty year old design is struggling to face the growing difficulties and challenges of the modern era (Davidson, 2017). Growing costs, increasing competition, and the need to prepare students for a globalized, fast-changing society will push universities to adopt new educational models (Davidson, 2017; Waterbury, 2011). As a solution to this issue, some universities and researchers have sought to apply Lean to the university system (Balzer, 2010; Balzer, Brodke, & Kizhakethalackal, 2015; Balzer, Francis, Krehbiel, & Shea, 2016; Vukadinovic, Djapan, & Macuzic, 2017).

Lean is a process improvement philosophy that originated from the Toyota Production System (Womack, Jones, & Roos, 1990) that is focused on systematically and continuously eliminating waste (any activity which does not add value to the product or service provided by a company). Higher Educational Institutions (HEIs) have been conceptualized as riddled with waste, such as excess movement of material or staff, unnecessary approvals, queuing for anything, etc. (Douglas, Antony, & Douglas, 2015), and as such, there is great potential for Lean to provide significant improvements and cost savings (Krehbiel, Ryan Jr, & Miller, 2015).

Existing literature is largely focused on applying lean to administrative operations or general institutional interventions (Balzer et al., 2016). Balzer et al. (2016) describe this existing literature as conceptual or case studies. Comparatively little work has been performed to apply lean to teaching—one of the core functions of the university. This is problematic because existing research shows that the passive learning techniques of the previous century are decreasingly effective at preparing students for this rapidly changing, globalized society (Davidson, 2017).

The following hypothetical comparison illustrates an additional argument for applying lean to teaching. A traditional manufacturing process (Womack et al., 1990) appears similar to that described in Figure 1. Operations are arranged linearly and raw material move through the system in large batches, incurring a large inventory build-up between operating stations and production delays due to the unexpected variability associated with large batches. Errors during the production process may propagate throughout an entire process and may not be caught until the end-of-line inspection, leading to the need for rework or scrap.

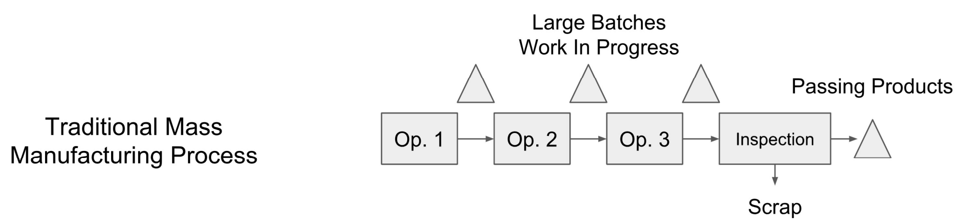


Figure 1: A traditional mass manufacturing process. Rectangles represent operations while triangles represent materials.

Now consider the hypothetical traditional passive lecture classroom shown in Figure 2 below. Students pass from lecture to lecture, accruing large batches of information. Errors in student understanding (knowledge quality) may compound over time and are not caught until a large summative assessment (end-of-line inspection) at the end of the term, at which point failing students must retake the course (rework) or drop the course altogether (scrap).

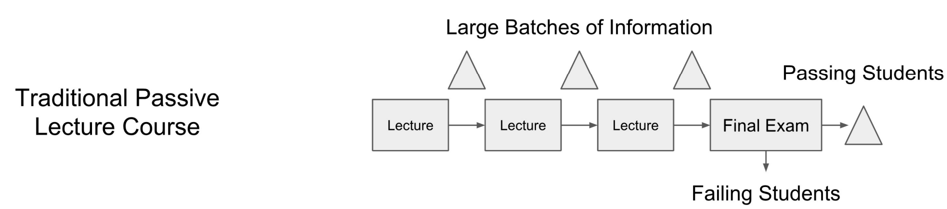


Figure 2: A hypothetical passive lecture course. Note the parallel to Figure 1.

This is a simplified example, but the parallels between teaching and manufacturing suggest there is potential for the application of lean process improvement principles and techniques to teaching.

This thesis demonstrates the potential effects of applying lean principles and practices to higher educational teaching and course design. This work has two primary components phases. The first phase is a literature review to summarize the current state of lean applications in teaching and curriculum design. The outputs from this first stage are conceptual comparisons between Lean teaching and current educational theory and practice. The second phase is a case study in which lean teaching is applied to an introductory engineering programming course. Höök & Eckerdal (2015) argue that introductory programming courses traditionally have low student pass rates. If effective, this lean approach should positively impact student pass rates. This work is significant among other works in that it is one of the few to draw specifically on the educational literature in arguing the validity of potential lean pedagogical techniques; it additionally explicitly exemplifies lean process improvement as relating to course improvement as opposed to the few pre-existing works.

Literature Review

Lean Production

Lean is a process improvement philosophy that is used to reduce waste (*muda* in Japanese) and to do more with less—less time, less resources, etc. This is accomplished through five Lean Thinking principles: Specify Value, Identify the Value Stream, Flow, Pull, and Pursue Perfection (Womack & Jones, 2003). The lean literature, however, has demonstrated conceptual ambiguity between principles and practices; Mirdad & Eseonu (2015) created a detailed conceptual map and matrix based on extensive literature review and surveys of Lean experts; this allowed for a summarization of lean practices under six principles (Specify Value, Flow, Pull, Zero Defects, Continuous Improvement, and Respect for Humanity). A selection of principles and practices relevant to this study are described below.

The concept of Lean Production was first introduced in the landmark publication *The Machine that Changed the World* (Womack et al., 1990). This book reported the results of a five-year Massachusetts Institute of Technology study on the automotive industry, with focus on the transition from mass production methods to lean production. The authors outlined lean concepts and principles in a follow-up book, *Lean Thinking* (Womack & Jones, 2003).

Specify Value

Specifying value involves defining the ultimate customer and determining how the customer defines value. “Lean thinking… must start with a conscious attempt to precisely define value in terms of specific capabilities offered at specific prices through a dialog with specific customers” (Womack & Jones, 2003, p. 19).

Practices related to this principle include Quality Function Deployment (QFD), Value Stream Mapping (VSM), and waste identification/elimination. Quality Function Deployment is a methodology used to convert the voice of the customer to product or service requirements using a series of matrices (Pyzdek, 2003). A value stream map is a diagram that illustrates the material and information flows through a system; it may be used to show a system's current state, identify improvement opportunities, or design a system's future state (Dennis, 2016).

Discussions of "value" and "Lean" are incomplete without simultaneously explaining waste (*muda* in Japanese). Waste may be described as "…anything other than the minimum amount of materials, equipment, parts, space, or time that are essential to add value to the product" (Nicholas, 2011, p. 60), where value is anything that contributes to goal achievement. The seven types of wastes are: defects, transportation, inventory, overproduction, waiting time, overprocessing, and motion. Additional wastes have been listed as knowledge disconnection (where there are disconnects in knowledge within a company or between a company and its customers and suppliers) (Dennis, 2016) and unused employee creativity (where employees are not engaged to utilize their ideas, improvements, skills, learning opportunities, etc.) (Liker, 2004). While seeing waste is essential to Lean Thinking, being able to identify waste alone does not lead to a lean transformation (Dennis, 2016).

Flow and Pull

Flow describes the movement of work through the factory floor; in lean, the way material flows is antithetical to the traditional batch-and-queue methods of mass production (Womack & Jones, 2003). This involves practices including line balancing, cellular manufacturing, and load leveling (*heijunka* in Japanese) (Mirdad & Eseonu, 2015). Line balancing involves adjusting and distributing work center times in order to match, as closely as possible, the required cycle time of a process (Nicholas, 2011). The required cycle time is determined by the takt time, which is the rate at which finished products should be delivered to the end of the line (e.g. if 60 units of a product A are required in 60 minutes, then the takt time is 60 minutes/60 units = 1 minute per unit) (Dennis, 2016). Cellular manufacturing describes a type of factory layout in which workstations are arranged in cells dedicated to a specific function or producing a product family (Tompkins, White, Bozer, & Tanchoco, 2010).

A key characteristic of flow in lean production systems is the material movement in small lot sizes and ideally single-piece flow (Womack & Jones, 2003). While counterintuitive to traditional batch-and-queue thinking of mass manufacturing, single-piece flow proves to be more efficient due to the benefits of its flexibility and reduction of delays and work-in-progress (WIP) In order to accomplish such a system, significant set-up time reduction is required for changeover between operations; this can be accomplished using the technique of Single Minute Exchange of Die or SMED (Nicholas, 2011).

In a lean production system, flow works in synchrony with the *pull* principle (Waterbury, 2011). The pull principle, in essence, states that nothing will be produced unless requested by a downstream (internal or external) customer (the full list of rules for the pull principle may be found in Nicholas (2011) p. 211). A practice characteristic of pull is Kanban, which describes a set of techniques for signaling when to begin production of a component (Nicholas, 2011). Flow and pull working together characterizes a Just-In-Time (JIT) system, where materials arrive just in time for usage in the correct quantity at the correct location (Karlsson & Ahlström, 1996).

Zero Defects

The zero defects principle describes lean's disposition towards quality: have no defects in the production process (Karlsson & Ahlström, 1996). Under this principle exist many techniques including 5S, Andon Boards, Root Cause Analysis, Visual Management, Poka-yoke, Statistical Process Control, Total Productive Maintenance, and Jidoka (or Autonomation) (Mirdad & Eseonu, 2015).

5S is an organizational methodology used to organize a workspace; it consists of 5 steps: Sort, Set in Order, Shine/Sweep, Standardize, and Sustain (Dennis, 2016). Andon boards are visual boards with lights that are used to display the status in the production process; at Toyota, where the tool was developed, when workers on the production line observed an issue, they would pull on a cord (named an Andon cord) to display a signal on an Andon board. This could allow a foreman to come to the area and solve the problem, in some cases stopping the line so all workers may gather together to determine the root cause (Nicholas, 2011; Womack et al., 1990).

The Andon board demonstrates two additional practices under the zero defects principle: Root Cause Analysis and Visual Management. Root Cause Analysis is a formal process used in finding the root cause of a problem and eliminating it such that the problem never occurs again ; tools used in this include the 5 Why's method and Cause-Effect diagrams (Brassard & Ritter, 2010). Visual Management is a broad term describing how to make information easily visible to workers; this includes creating visual controls which can convey all relevant information to a worker at a glance(B. Emiliani, 2015).

Additional Zero Defects practices include Poka Yoke, Japanese for "mistake-proofing." This technique is implemented in processes to prevent mistakes from ever being made; e.g. designing systems to be intuitively assembled or used only in the correct way (Dennis, 2016).

Statistical Process Control is used for monitoring processes to maintain performance within certain process standards (Montgomery, 2013). Total productive maintenance is a methodology for maintaining equipment with the final goal of having zero breakdowns in machine performance; all members are responsible for the tasks of maintaining their own equipment (Dennis, 2016). Finally, Jidoka, or autonomation, is the design of processes such that they automatically stop when a problem is detected so that the problem may be resolved (Abdulmalek, Rajgopal, & Needy, 2006; Dennis, 2016). The process by which problem solving occurs uses RCA and Poka-Yoke as described above and the improvement techniques listed in the following section, Continuous Improvement.

Continuous Improvement

The final principle of Womack and Jones, Pursue Perfection, is equated to the principle of continuous improvement (Karlsson & Ahlström, 1996; Mirdad & Eseonu, 2015). Lean has been described as a journey with no end destination; there is always room for improvement. This principle primarily manifests in techniques include A3s and PDCA as well as Kaizen events (Dennis, 2016). PDCA is short for the Plan-Do-Check-Act cycle of improvement (Montgomery, 2013). The Planning phase consists of defining the problem, determining the current state of the system through methods such as value stream mapping and Genchi Genbutsu (Liker, 2004), analyzing problems using root cause analysis and quality tools, then determining improvements to make for the future state of the process. The Do phase moves to implementing the improvements, and the Check phase checks whether the interventions resulted in actual improvement. The final Act phase is based on the results of check; if the intervention was successful, standardize the new process and disseminate through the organization accordingly; if unsuccessful, repeat the cycle.

The phases of PDCA are typically summarized in an A3, a one-page document describing the problem and actions being taken in a clear manner such that a reader can understand what is happening within a few minutes (Dennis, 2016).

Kaizen in Japanese literally means “Change for the Better” (B. Emiliani, 2015). Kaizen events are common vehicles for implementing rapid improvements in a short time (a few hours to a 4 or 5 days depending on the situation), and involve solving a specific problem or set of problems through the usage of previously described techniques (M. L. Emiliani, 2005).

Respect for Humanity

The Respect for Humanity principle is considered the primary cause factor in lean implementation (Mirdad & Eseonu, 2017). It consists of practices including employee involvement, employee training and growth initiatives, and humanizing the work environment (Mirdad & Eseonu, 2017). In general, this principle involves involving employees in the improvement process and trusting them to standardize and improve their work and environment, as Toyota did through the 20th century (Womack et al., 1990).

Lean in Higher Education

Lean Teaching

M. L. Emiliani (2004) outlines a strategy for applying lean principles and practices to business school courses. These principles and practices include 5S, standard work, visual control, load smoothing, just-in-time, voice of the customer, Muda (waste), Mura (unevenness), Muri (unreasonableness), and formal root cause analysis (M. L. Emiliani, 2004). He released a book that later built on the subject called *Lean Teaching* (B. Emiliani, 2015)[1], which detailed further how practices were applied to different components of his courses and the effects of such practices.

Emiliani applied lean to address issues like basic student stumbles, framework for inquiry, the syllabus, required readings, homework assignments, examinations, course evaluations, and visual controls (as course summary reference material). He attempted to address basic student stumbles using mistake-proofing (poka-yoke) techniques by providing students with lists of common errors to avoid and some assignment specific errors. A framework for inquiry is addressed by providing students with a code of ethics from an appropriate professional society, which could guide student thinking throughout the course; while not necessarily a lean practice, the need to increase competency in ethics in engineering education is also well established (Flumerfelt, Kahlen, Alves, & Siriban-Manalang, 2015). The syllabus is reduced to a simple one- to two-page document to improve the course focus, reducing the excessive documentation that typically comes with course syllabi. Emiliani focuses course material on short, focused readings that highlight the most

---

[1] M. L. Emiliani and B. Emiliani are the same person. In his 2004 paper, Emiliani uses his real initials, while in his book, he uses his nickname, Bob.

important concepts, arguing that students miss important concepts in longer reading requirements. This approach is akin to waste reduction, in which non-value adding reading material is eliminated. Homework assignments are usually due every few weeks, and the batching of information may lead to uneven workloads; smaller, focused weekly assignments smooth the workload (heijunka), and standardized formats assist grading speed and accuracy. Examinations are distributed from large batch-and-queue quantities of information into weekly assignments for regular assessment with Just-In-Time feedback. A course evaluation is performed in the middle of the course to allow for immediate improvement to the course, reflecting the respect for humanity principle of lean. Finally, visual controls are implemented through providing a one-page summary sheet on the entire course to the students as a reference document to take with them after the course. Table 1 below summarizes his applications.

Table 1: Summary of Emiliani's Lean Teaching applications

Course Component	**Solution**
Basic Student Stumbles	Poka-Yoke increase student awareness of errors
Framework for Inquiry	Code of ethics from professional society
Syllabus	Simplification to short, 1-2 page document
Required Reading	Reduce non-value-added material (overproduction/overprocessing) and focus readings to shorter pieces.
Homework Assignments	Standard formats for assignments, load smoothing to weekly assignments
Examinations	Weekly graded assignments for JIT feedback
Course Evaluation	Student feedback used in middle of course
Visual Control	Summarize course on one sheet of paper

While comprehensive in the material, *Lean Teaching* as presented by Emiliani has several limitations. The methods described are based almost entirely on his anecdotal experiences, and there are no connections between each practice and the educational literature on effective teaching practices. The techniques are primarily applied as practices without explicit connection to lean principles or the rest of the existing literature on the nomenclature and classification schemes. The primary principles described, continuous improvement and respect for humanity, are indeed central principles to lean (Mirdad & Eseonu, 2017), but do not address the other four principles described earlier. Most importantly, while there is positive information on student evaluations of the course, there is no actual information on whether the described techniques improved student learning.

Thus, while Emiliani's techniques hold some promise, more work is required establish whether these techniques improve learning outcomes and to develop formal parallels between lean principles and practices and the classroom, in curriculum design and pedagogy. The next section explores the literature to identify additional work related to the application of lean to the classroom.

Additional Literature

In the past two decades, there has been a growing body of literature around applying lean to Education. Balzer, Francis, Krehbiel, & Shea (2016) reviewed 64 publications, including journal articles (41), technical reports (6), trade publications (5), magazine and periodicals (7), and books (4 and one book chapter) on the current state of Lean Higher Education (LHE). They highlight significant potential for lean to improve higher educational institutions (HEIs), but more work is needed to formalize LHE definitions and create frameworks, expand measures for LHE's impact, and develop more empirically grounded studies on lean applications to Higher Education. Within their results, it appears that most current research on lean involves applications of lean to administrative functions or the aspects of the application of lean to the institution.

The literature identifies lean as beneficial to HE, but there are few to no applications within the context of actual teaching, curriculum, and assessment. Using reviews by Balzer et al. (2016), Cudney, Venuthurumilli, Materla, & Antony (2018), and Vukadinovic et al. (2017) and searches on the Google Scholar database using keywords listed in Table 2, a total of 15 publications were identified relating to Lean applied to teaching and curriculum. A list of these articles is provided in Table 2, with the quantity of publications over time summarized in Figure 3. The remainder of this section will discuss these articles along with other key publications related to applications of lean to teaching, curriculum, and assessment. Figure 4 summarizes the usage of lean principles and practices in various articles in a matrix format.

Table 2: Keywords used in identifying additional applications of lean to teaching and curriculum and articles identified through the literature review

Keywords	**Articles**
• Lean implementations in higher education • Lean Higher Education • Lean Institutions of Higher Education • Lean Higher Education Institutions • Lean Curriculum Design • Lean Applications to Curriculum Design • Lean Course Design • Lean Applications to Course Design • Lean Instructional Design • Lean Applications to Instructional Design • Lean Teaching	1. Ahlstrom (2004) 2. M. L. Emiliani, (2004) 3. M. L. Emiliani (2005) 4. Dey (2007) 5. Tatikonda (2007) 6. Alagaraja (2010) 7. Waterbury (2011) 8. Pavlović, Todorović, Mladenović, & Milosavljević (2014) 9. B. Emiliani (2015) 10. Suárez-Barraza & Rodríguez-González (2015) 11. Brouwer-Hadzialic & Wiegel (2016) 12. El-Sayed, El-Sayed, Morgan, & Cameron (2011) 13. M. L. Emiliani (n.d.) 14. Mansur, Leite, & Bastos (2017) 15. Thomas, Antony, Haven-Tang, Francis, & Fisher (2017)

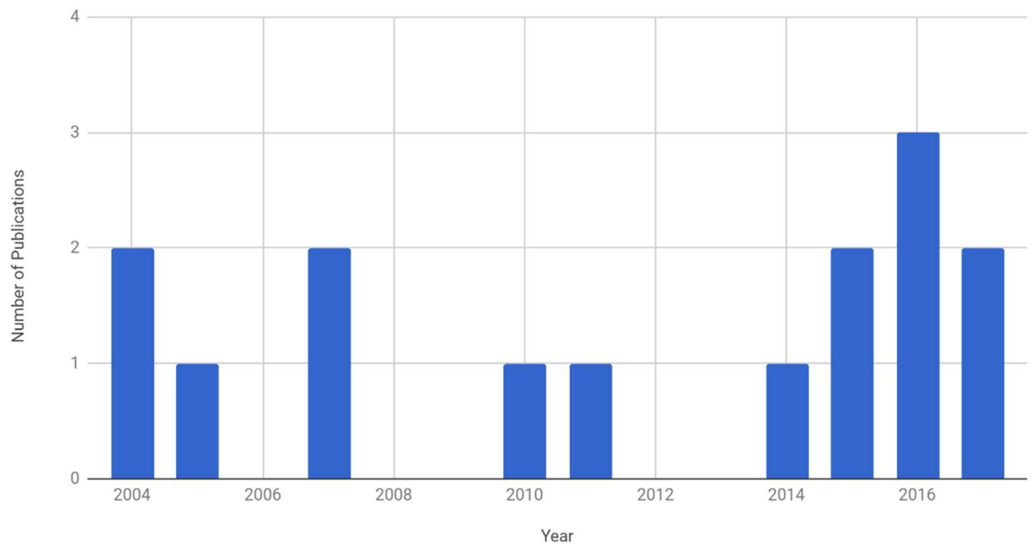


Figure 3: Publications related to Lean Teaching since 2004

		Lean Principles/Practices																																			
Author	Document Type	Specify Value	Quality Function Deployment	Value Stream Mapping	Waste identification/elimination	Shainin KVST	Flow	Cellular manufacturing	JIT	Load Smoothing	Small batch sizes	Takt Time	Pull	Kanban	Zero defects	Perfection	Root Cause Analysis	Poka-yoke	Quality at the source	Standardization/Standard Work	5S	Statistical Process Control	Total Productive Maintenance	Visual Management/Control	Kaizen	PDCA	Respect for People	Decentralized Responsibilities	Multifunctional teams	Other	Supplier integration	Balanced Scorecard	Theory of Constraints	Throughput	Hoshin Kanri	Vertical Information Systems	Mura	Muri	Total
Ahlstrom (2004)	Journal Article				x										x			x							x			x	x							x			7
Emiliani (2004)	Journal Article								x	x							x			x	x			x	x		x												8
Emiliani (2005)	Journal Article																								x														1
Dey (2007)	Journal Article				x						x								x	x							x		x		x								7
Tatikonda (2007)	Journal Article		x					x																								x							2
Alagaraja (2010)	Journal Article		x	x	x		x	x	x					x				x		x				x	x								x	x	x				13
Waterbury (2011)	Book	x		x	x		x					x	x			x				x	x					x													9
Pavlovic et al. (2014)	Journal Article				x												x					x			x														4
Emiliani (2015)	Book				x				x	x	x							x		x	x			x	x		x										x	x	12
Suarez-Barraza & Rodriguez-Gonzalez (2015)	Journal Article																									x													1
Brouwer-Hadzialic & Wiegel (2016)	Journal Article			x																																			1
El-Sayed et al. (2016)	Journal Article	x		x			x						x			x																							4
Emiliani (2016)	Journal Article (in review)												x						x					x															3
Mansur et al. (2017)	Book Chapter	x		x	x		x				x		x			x																							6
Thomas et al. (2017)	Journal Article					x																																	1
Total		3	2	5	7	1	4	2	3	2	3	1	4	1	1	3	2	3	2	5	3	1	0	4	6	2	3	1	2		1	1	1	1	1	1	1	1	

Figure 4: Matrix of lean principles and practices applied to teaching as found in the literature

Lean Principles and Tools in the Classroom

This section describes how the literature has translated lean principles and tools to the classroom, the limitations of these translations, additional potential for transfer, and alignment with current educational literature on learning. This section will be structured like the previous section describing lean so that techniques are organized based on lean principles.

Specifying Value in the Classroom

Central to lean is specifying value in terms of the ultimate customer (Womack & Jones, 2003). There are various stakeholders in the HE system, including industry and future employers, faculty, the institution and administration, and, of course, students themselves (Thomas et al., 2017; Waterbury, 2011). What then, is value to these stakeholders? The answer to such a question is beyond the scope of this thesis. However, within the scope of a HE course—in curriculum and instructional design—one can identify two streams of value. This division is arbitrary but helps guide the discussion on value adding components of a course. One point of value is *what* is taught in a course, that is, the content of a course expressed as learning outcomes in a course syllabus. It can be argued that such learning objectives are difficult for students to define (they do not know what they have not learned about), and thus their definition is primarily the responsibility of the teacher, institution, and, depending on the course, the professional society related to a student's major.

Content describes *what* is taught in a course. However, a class as whole is an experience, meaning that *how* the course is taught—the pedagogical practices used—are of equal importance. A base definition for value in pedagogical practices may be taken as follows: "Value-adding pedagogical practices are those that help students learn and achieve the course learning outcomes and align with research-based instructional practices and recommendations as well as current learning theories." These pedagogical practices may be shaped by lean principles and will be discussed below in terms of how lean principles and practices may transfer to the learning environment.

Specifying value as discussed within the literature is primarily theoretical: Waterbury (2011) argues that customers in education are students and potential employers, and that the concept of value must be trained into employees. El-Sayed et al. (2011) describe customers in educational provider circles and education receiver circles, with educational providers being the instructor, department, and university and education receivers including the student, industry (potential employers and the profession), and society at large. Mansur et al. (2017) strongly align with student needs as a source of value specification and urge the application of this principle through involving students in pedagogical planning meetings.

On the pedagogical side, specifying value from the view of the customer translates to the idea of student-centered learning. It is well established that students learn through building upon pre-existing knowledge (Bransford, Brown, & Cocking, 2000); as such, it is sensible to design courses starting with what students bring to the classroom and accommodating their learning needs. Specifying value from the viewpoint of the students would allow for this to occur. Table 3 and Table 4 below summarize the application of certain practices related to specifying value, with the former discussing all tools except the seven wastes and the latter detailing specific applications of the seven wastes.

Table 3: Applications of practices related to Specifying Value to teaching

Practice	Application in Teaching	Additional Notes
Quality Function Deployment	Used to define course content in an accounting course (Tatikonda, 2007). Conceptually discussed as allowing for incorporating learner needs, instructor input, and support staff input to achieve cross-functional input from all parties in course design (Alagaraja, 2010).	A related application, the Shainin Key Variables Search Technique, is used to develop undergraduate engineering curriculum with input from multiple stakeholders: full-time students, part-time students, employers, and staff (Thomas et al., 2017).
Value Stream Mapping (VSM)	Suggests using VSM on course syllabi to visually represent material and information flow (Alagaraja, 2010). Envisions using VSM to categorize value-adding and non-value adding processes in education, with academic value streams including curriculum, delivery of credit courses, and delivery of non-credit courses (Waterbury, 2011) Discusses the potential of mapping value streams in identifying value-adding and non-value adding activities while maintaining value definitions from students and employers (Mansur et al., 2017) Used to map program outcomes between program educational objectives (El-Sayed et al., 2011) Applies a related technique, service blueprinting, to capture roles of individuals within the learning process and the nature of information flows in the classroom	The general ideas behind value stream mapping—making the actual current state of a system known and easily identifying its components and material/information flows—aligns with educational techniques including curriculum mapping, allowing one to see issues within a process including assignment duplication, missing learning objectives (value-adding activities), or uneven assignment flows (Jacobs, 1997; Waterbury, 2011), and constructive alignment, wherein all activities in a course are brought into alignment with learning objectives defined through a constructivist[2] lens (Biggs, 1996).
Waste	Generally mentioned by Ahlstrom (2004), Alagaraja (2010), Dey (2007), and Waterbury (2011). See Table 4 for a detailed list of wastes as applied to the teaching environment.	

[2] Constructivism is a learning theory based on the idea that knowledge is constructed based on the experiences of learners. See (Ertmer & Newby, 2013; Philips & Soltis, 2009)

Table 4: The 7 wastes of lean in teaching processes as documented by the literature

Waste	From the literature
Transportation	"Students passing from year to year, information transfer from the student service to the notice board regarding exams and mid-term tests" (Pavlović et al., 2014) Waste related to the knowledge exchange process between students and teachers including "…bad teaching classroom methodologies, non-adoption of multimedianess… lack of accessibility to knowledge for students with special learning needs, non-flexible course structure and schedules." (Mansur et al., 2017) Missing framework for inquiry; excessive required readings with little direction on important information; excessive, large batches of homework (B. Emiliani, 2015)
Inventory	"Students who are left back, great variation in the number of passed exams in different exam periods, taking the same courses again for not receiving the signature at the end of the previous term, students who failed or withdrew from taking the exam, unnecessary and redundant parts of the curriculum" (Pavlović et al., 2014) Consequence of overprocessing: too much stocked knowledge decreasing student learning efficiency; obsolete knowledge (Mansur et al., 2017) Large, infrequent examinations testing large batches of knowledge; students leaving the course with large quantities of instructional material (B. Emiliani, 2015)
Motion	"Students moving from classroom to classroom, not getting the information on: taking exams and mid-term tests, exam periods and registration, mistakes upon registration, at the right place at the right time." (Pavlović et al., 2014) Activities performed by people, e.g. students, teachers, and staff. Includes "transit among university facilities…, bad accessibility for students with special locomotion needs… [no] means to record and/or promote asynchronous class meetings… synchronous-centred [learning] approach." (Mansur et al., 2017) Excess required reading; excess homework assignments. (B. Emiliani, 2015)
Waiting	"Information (in exams, mid-term tests, exam periods), waiting to enroll in the next year, students and professors being late for lectures and practice classes, waiting for and replacing equipment necessary for teaching." (Pavlović et al., 2014) Waiting in lines for administrative processes or teachers late to class. (Mansur et al., 2017) Lengthy syllabi; feedback delays for examinations; delay in when course evaluation is administered and when changes are implemented (if at all) (B. Emiliani, 2015)
Overprocessing	"Re-taking exams, re-enrolment, re-taking classes." (Pavlović et al., 2014) "Tasks, assessments, activities that are not really useful to the learning process." Typically occur with teacher-centred learning approaches. (Mansur et al., 2017) No information on basic student stumbles; lengthy syllabi; excess required reading; excessive homework assignments (B. Emiliani, 2015)
Overproduction	"Excessive number of students who did not find a job within six months of graduation." (Pavlović et al., 2014) "…any knowledge acquired and assessed more than what is needed to the learning process." (Mansur et al., 2017) No prevention of basic student stumbles; length syllabi; excessive required reading; excessive homework assignments. (B. Emiliani, 2015)
Defects	"Wrong teaching plan and classroom schedule, mistakes upon registering for exams, failed exams, withdrawing from exams, incomplete or incorrect information." (Pavlović et al., 2014) "Wrong, inappropriate and obsolete knowledge" (Mansur et al., 2017) No prevention of basic student stumbles; length of syllabi; examination queuing and feedback delays; delay in course evaluation; leaving the course with excess instructional material (B. Emiliani, 2015)

Flow and Pull in the Classroom

In manufacturing, the Flow and Pull principles of lean primarily describe the material flows within a production system. Thus, in order to apply these principles to the classroom, one must identify *what* flows in the classroom are; in other words, what material is being worked on in the classroom (and the educational process in general)? Flow as applied to the educational process is a fairly new concept (Waterbury, 2011), and as such, there are different positions one may take in viewing material flows in education. Some take the position of students being the materials in the educational process, with teachers as operators (Pavlović et al., 2014). Others view the content knowledge that is being delivered to students as the material (Alagaraja, 2010; Mansur et al., 2017). Physical materials such as assignments may also be considered part of material flows (Alagaraja, 2010; B. Emiliani, 2015). While it is beyond the scope of this paper to address what are the "true" material flows in this classroom, it becomes evident that it is important to at least identify what one considers the material flows of the classroom when attempting to apply lean principles and practices to their teaching.

Pull as applied to the classroom also has varying interpretations. El-Sayed et al. (2011) apply pull through having assessments of courses pulled while they are needed; this application is tangential to the current discussion as they primarily discuss assessment of course curriculum, not teaching and learning itself. Mansur et al. (2017) argue that students motivate their own learning and thus pull knowledge as it is desired (taking the view of knowledge as the material flow). They go further to draw a parallel to the constructivist, active-learning methodologies of project-based learning and problem-solving learning, wherein students must pull knowledge in order to address the problems/projects given on hand. Emiliani (2018) applied pull in a specific example where he presents all information available in the course to students through the learning management system. Students are tasked with scanning through all the materials to determine what is of interest and develop questions they desire to be answered for the entire course. From this, students must pull answers from the professor, which in turn shapes how the professor provides instruction. This method is similar to other pedagogical practices of allowing students to shape their own curriculum, with some teachers going so far as allowing students to design their own course syllabus (Davidson, 2017).

Table 5 summarizes the applications of techniques related to the flow and pull principles as related to the classroom.

Zero Defects in the Classroom

The discussion of the zero defects principle by itself in the literature is marginal (Ahlstrom, 2004); however, the application of techniques classified under the principle are more numerous. Mentioned techniques within the literature include RCA, Poka-Yoke, Quality at the Source, Standard work and Standardization, 5S, and Visual Controls. Applications of these techniques are summarized in Table 6. In general, the zero defects principle can be thought to align with the assessment component of curriculum. How do you assure that students have correctly learned what they are supposed to? The ideal form of zero defects in the classroom could manifest as content being learned by all students correctly the first time through a lesson or course (whether this can be realistically accomplished, of course, is another story).

Table 5: Applications of practices related to Flow and Pull to teaching

Practice	Application in Teaching	Additional Notes
Just-in-Time Manufacturing	Suggested as producing or delivering items exactly when needed in the correct quantity to students (Alagaraja, 2010) Applied through returning graded assignments immediately when required, typically the day before the assignment is due (B. Emiliani, 2015)	The application of Just-In-Time to teaching has seen other pedagogical applications in the form of "Just-In-Time Teaching" (or JITT); see (Simkins & Maier, 2010).
Load Smoothing	Applied to assignment flows in balancing homework loads and using small, weekly assignments and quizzes as opposed to larger, infrequent assignments and large examinations, which batch larger quantities of information (B. Emiliani, 2015; M. L. Emiliani, 2004)	
Cellular Manufacturing	Suggested for application in the logical organization of course content, allowing for the development of sub units which may stand alone or work as an integrated whole (Alagaraja, 2010) Applied in grouping accounting topics in a more sensible flow beyond the separation of topics presented in textbook organization (Tatikonda, 2007)	
Kanban	Mentioned by Alagaraja (2010), but appears to align more closely with the technique of Poka-Yoke (mistake-proofing).	
Takt Time	Mentioned briefly by Waterbury (2011) in noting that all students differ and thus have differing time requirements for all subjects; therefore, there is no singular "takt time" that may be applied to students, at least in the learning process.	

Table 6: Applications of practices related to Zero Defects to teaching

Practice	Application in Teaching	Additional Notes
Root Cause Analysis (RCA)	Applied to every problem that arises in his courses (B. Emiliani, 2015; M. L. Emiliani, 2004). Used to determine causes to problems within the curriculum flow of a first-year mechanical engineering department's coursework (Pavlović et al., 2014)	RCA may have potential to be used in pedagogical fashions to help diagnose issues within student misunderstandings
Poka-Yoke	The whole teaching process is conceptualized in *Lean Teaching* as a "mistake-proofing activity." Includes efforts to show and prevent basic student stumbles (B. Emiliani, 2015). Suggests providing students with examples of common learner mistakes and providing feedback on activities (Alagaraja, 2010)	A potential extension of Poka-Yoke to the classroom is having all students learn all material correctly with the first delivery of material. How feasible this may be is another question, and it is arguable that some of the best lessons can come through students making mistakes and failures.
Quality at the Source	Applied within the online learning management system to automatically grade some assignments (M. L. Emiliani, n.d.). Mentioned by Dey (2007), but appears to be more of an application of benchmarking course outlines.	May be generally applied as automatically grading some assignments or having assessment occur quickly within lessons, such as through formative assessment techniques, where evaluations of students are performed during activities to provide students with feedback (Sadler, 1989).
Standardization	Various authors mentions standardization through simplifying and standardizing syllabi, assignments, course schedules, examinations, problem sets, etc. (Alagaraja, 2010; Dey, 2007; B. Emiliani, 2015; M. L. Emiliani, 2004; Waterbury, 2011).	Standardization is already found in education, with standardized assessments and curriculums required by various accreditation programs. Standardizing and simplifying syllabi may assist in course/curriculum organization, but the extent to the benefit to student learning appears yet to be seen from the literature examined.
5S	Applied as a method for organizing and standardizing materials, going along with the standardization practice above (M. L. Emiliani, 2004; Waterbury, 2011)	May be better suited for administrative organizational purposes compared to teaching students, although 5Smay be applied in streamlining course materials and removing extraneous/outdated materials.
Visual Controls	Applied in summarizing and entire course on one sheet of paper as a means of summative assessment and with the intention of providing students with a quick material reference in the future (B. Emiliani, 2015; M. L. Emiliani, 2004, n.d.) Recommends visually summarizing course content and creating a dashboard for course performance using diagrams and concept maps (Alagaraja, 2010)	

Continuous Improvement and Respect for Humanity in the Classroom

The principles of continuous improvement and respect for humanity do not appear to manifest in one particular component of curriculum development, rather, as overarching principles that influence the overall approach towards the process. Continuous improvement, translated to the classroom environment, primarily means continuously improving how one teaches and how one's courses are run (Ahlstrom, 2004; B. Emiliani, 2015) and being responsive to student feedback and eliciting improvement from students (M. L. Emiliani, 2004).

Table 7: Applications of practices related to Continuous Improvement to teaching

Practice	**Application in Teaching**
PDCA/PDSA	PDCA is used as a model for shaping the course syllabus (Suárez-Barraza & Rodríguez-González, 2015). PDSA is used as the basis for he Educational Lean model (Waterbury, 2011).
Kaizen	Implemented Kaizen events with other professors to improve business curriculum (M. L. Emiliani, 2005).

Respect for humanity is straightforward in application: involve, respect, and grow people within the context. In the literature, this manifests through organizing multifunctional teams of teachers for teaching and designing curriculum (Ahlstrom, 2004). Emiliani (2015) applies respect for humanity through respecting student time and needs and through eliciting student feedback for course improvement (thus operating in tandem with the continuous improvement principle).

It should be noted that a key difference between the traditional manufacturing process and the teaching environment is that students are not inanimate objects; they are people. Students may be simultaneously conceptualized as raw material (to be converted to the final, degree-holding product for future employers), operators (on themselves and surrounding students), suppliers (of their background knowledge), and customers (receivers of knowledge, the ones paying for the service). This makes the case for the respect for humanity principle applications within the classroom.

Table 8: Summary of Lean Principles/Techniques with associated applications from the literature

Principle/Technique	**Key Points Regarding Application in Teaching**
Specifying Value	Define value from the viewpoint of the ultimate customer (students), and other stakeholders (faculty, industry, society, etc.) Value is defined in the learning objectives of the course. Activities contributing to the (more) successful accomplishment of learning objectives are value adding; those that do not contribute are non-value adding. Pedagogically, specifying value can translate to student-centered learning.
Quality Function Deployment	May be used in planning the content and pedagogy in a curriculum; may incorporate learners' (students') needs.
Value Stream Mapping	May be applied to map course information and material flow and map program objectives. Such mapping of the current state allows for developing the future state of the course. Reminiscent of constructive alignment.
Waste Identification and Elimination	See **Error! Reference source not found.** for the specific applications of each waste
Flow	Consider what material you are examining as flowing through the system: students through the four-year curriculum? Physical assignment materials? Information and content knowledge? Aim for steady, continuous flow of these materials, with a reduction from large batches of material to smaller quantities at a time.
Pull	Students motivate their individual learning and pull knowledge as needed. Consider using pedagogies that reflect pull such as project-based learning and problem-solving learning
Just-In-Time	Provide information and feedback (e.g. graded assignments) exactly when students require it.
Load Smoothing	Smooth the quantity of work and information students must work with so that it is roughly equivalent across given time intervals (e.g. a week); antithetical to the batching method of large, infrequent examinations.
Cellular Manufacturing	Group topics into modules to process in logical units. May be extended to a higher level in the progression of courses in a curriculum
Takt Time	The rate at which topics or information *must* be understood or learning outcomes achieved as a course progresses. All students have a different takt time for learning.
Kanban	No current applications. May concern any signal representing a demand for more information by the student (e.g. questions).

Zero Defects	Aligns with the assessment component of curriculum. Have zero defects in the accomplishment of learning outcomes, that is, no misunderstandings in what is learned.
Root Cause Analysis	Perform anywhere problems occur in the teaching process, such as in course material, delivery of lessons, and in student misunderstandings. Root cause analysis provides some of the means for investigating and preventing future student mistakes from occurring in the future.
Poka-Yoke	Design course material and delivery such that student misunderstanding never occurs in the first place; learn material correctly the first time.
Quality at the Source	Check for understanding and achievement of learning outcomes at every step along the process. Some methods include formative assessment, using grading results from assignments to provide feedback to students, and even automating some assessment systems.
Standard Work/Standardization	Standardize course materials, syllabi, etc. Many curriculums and assessments are standardized by various accreditation societies.
5S	Use in the process of organizing and standardizing course materials.
Visual Control	Providing information in an easy, visual manner for students to view and see progress through the course; summarize a course on one sheet of paper.
Continuous Improvement	The instructor constantly makes efforts to solve issues in the course and responds to the needs and feedback of students
PDCA	Can be used to shape the design of a course syllabus: Plan a lesson, perform the lesson (Do), assess the results (Check), Act upon the results.
Kaizen	Use as a method to bring together multiple members involved in the delivery of a course to improve it.
Respect for Humanity	Recognize the value of student time and involve student feedback in course improvement.
Multifunctional Teams	Develop teams of instructors and administrative staff to work on and improve the overall curriculum.

Methodology

The following section describes the case study in which lean principles and practices are applied in an attempt to provide potential improvements for a course in higher education (HE). First, an overall description of the course is provided, followed by the methodology, then results and discussion.

Course Description

Oregon State University (OSU) is a public university with approximately 28,000 enrolled students as of the Spring term of 2017 ("Enrollment Summary - Spring Term 2017," 2017). The university operates on a quarter schedule with four academic terms (Fall, Winter, Spring, and Summer). The Introduction to Engineering Computing course (code: ENGR 112) is offered every term through Oregon State University's College of Engineering. It is designed to introduce computer science and programming through learning the engineering programming language MATLAB, a language used in industry, research, and several upper division engineering courses. It is typically recommended by various engineering majors to take during the Winter or Spring term of their freshman year ("Pre-engineering course requirements by major," n.d.). The class has experienced a 3-year-average Drop-Withdrawal-Failure-Unsatisfactory rate of approximately 28% since Spring of 2017, indicating over one out of every four students does not pass the class. Considering that the class is such an integral part of the pre-engineering curriculum and a pre-requisite to admission into the pro-engineering program, a case can be made for improving the course.

The academic terms used for analysis in this thesis range from Fall 2015 to Spring 2017. The number of students taking this course during this time period range from 100 to 250, and three instructors taught this course during this time (one per term); the specific characteristics are described in Table 9. Terms before this time period are excluded due to changes in the grading system, course materials, and learning management system that make comparison between the courses too difficult. Summer terms are omitted due to the fact that they typically have much fewer students and are too different from the other terms to provide a useful comparison.

Table 9: Students per term (based on gradebook data) and associated instructor

Term	Number of Students	Instructor
Fall 2015	101	A
Winter 2016	136	B
Spring 2016	204	C
Fall 2016	108	B
Winter 2017	152	A
Spring 2017	246	B

The methodology of this thesis consists of a three-phase process based on lean principles to characterize issues within the course and provide improvements based on the analysis of said issues. Phase 1 aligns with the value specification principle of lean. This phase validates the course learning objectives against requirements prescribed by instructors in upper level courses. Phase 2 concerns identifying the current state of the course using a two-pronged approach. Part A analyzes the course material available from the most recent term of the course relative to the undertaking of this study, while Part B analyzes student responses on electronic Student Evaluations of Teaching (eSETs). This phase aligns with the value stream mapping practice found within lean and sees some application of the Flow and Respect for Humanity principles as well. Phase 3 applies root cause analysis in alignment with the Zero Defects principle to identify root causes to potential issues identified in the previous phases. Improvements to validated root causes are suggested and discussed in the Discussion section. Figure 5 provides a high-level outline of the methodology. Each phase is now discussed in detail below.

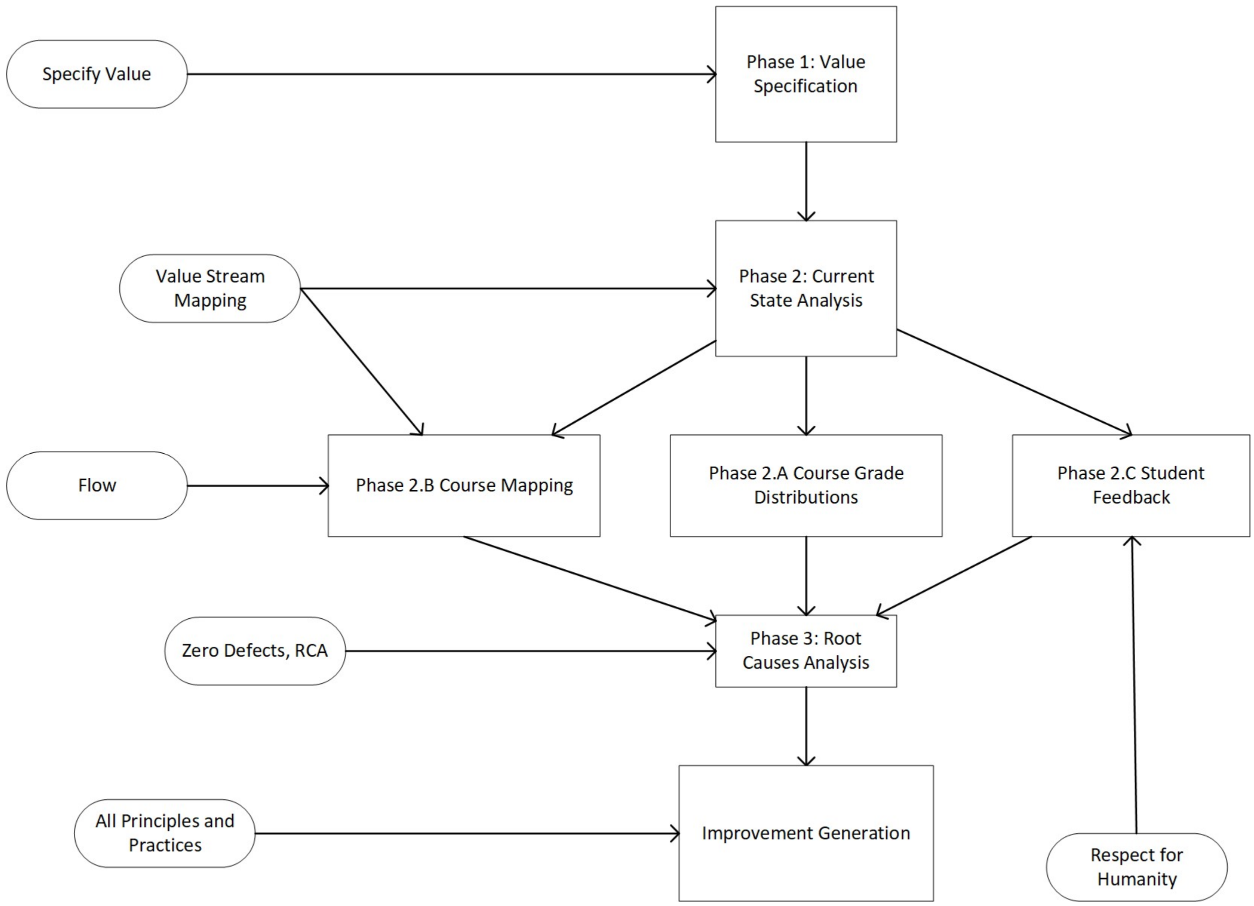


Figure 5: Methodology outline - rectangles are methodology phases would rounded shapes are the applied Lean principle/tool.

Phase 1: Value Specification

As discussed within the literature review, the Specify Value principle of lean relates to the content that is being delivered by a production system, that is, what the customer actually wants from the product being produced. To transfer this principle to the educational level, one must identify the content being delivered in that course. Within a course, this content is described at a high level by the course learning objectives and course syllabus.

Value-adding content in a course may be considered knowledge and/or skills which are deemed important for students to know. The question of *what* is important for students to know, however, can be difficult and subjective to determine. Multiple stakeholders and factors may be involved in determining what is important knowledge for students to know, including the instructor, the department and university, industry, the professional requirements of the program of study, available curriculum materials such as textbooks, and even the students themselves.

To completely validate the learning outcomes of ENGR 112 may fall beyond the scope of this thesis; however, a survey of instructors of courses dependent on ENGR 112 was performed in June of 2017 by the instructor of ENGR 112, providing information as to what are important topics to cover in the course with respect to the engineering curriculum within OSU (details of this survey are located in Appendix A). This data may be used to provide a demonstrative validation of the course learning objectives and topics to exemplify the application of the Specify Value principle to course design.

To perform this analysis, each topic from the survey is compared against the weekly learning outcomes of ENGR 112 in a matrix format (a full list of weekly learning outcomes for ENGR 112 is provided in Appendix B). For each topic and weekly learning outcome that coincide with each other, a mark is made within the matrix. The frequency of intersections may then be aggregated in the end rows and columns of the matrix to determine where topics are or are not being covered within the course and which topics within the course align or do not align with any of the requirements specified by the instructors. Figure 6 provides a high-level overview of this method. For those that desire a more explicit linkage to lean practices, this method may be considered a simplified version of Quality Function Deployment.

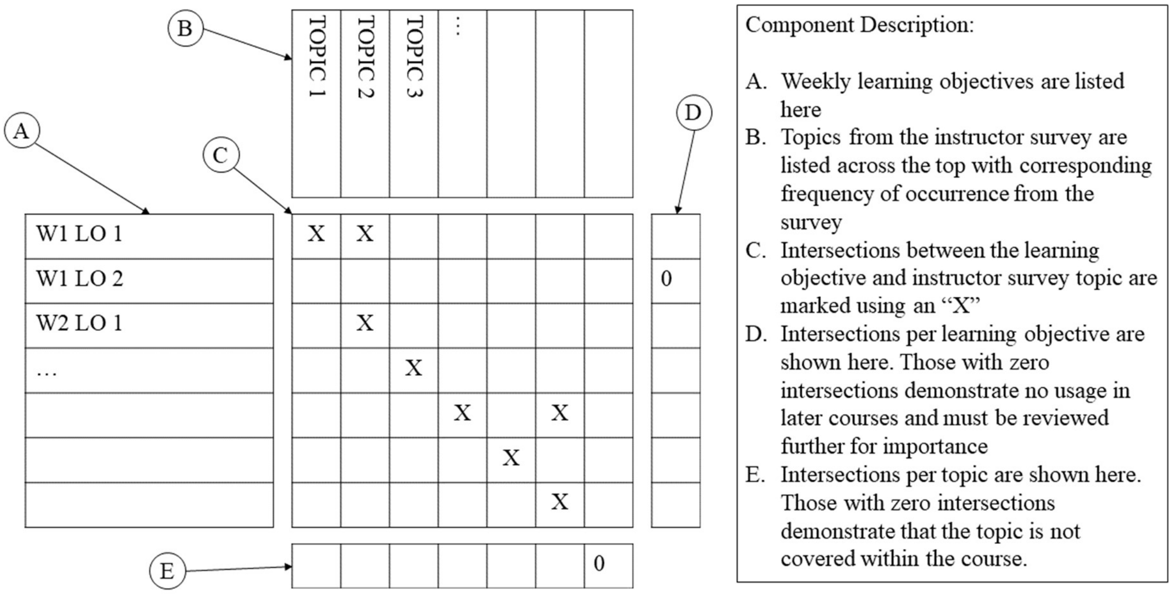


Figure 6: An overview description of the method used for validating course learning objectives.

Phase 2: Current State Analysis

The purpose of Phase 2 is to understand the current state of the course and identify potential issues associated with the course. This is performed using a three-pronged approach. Phase 2.A analyzes grade information available from the six available terms to identify possible behaviors in the grades and set a baseline for course performance. Phase 2.B maps the state of the course using available course materials provided by the instructor from the learning management system. Phase 2.C undertakes the quantitative and qualitative analysis of electronic Student Evaluations of Teaching. The sections below describe this in further detail.

Phase 2.A: Course Grade Distributions

This component of phase 2 undertakes the analysis of student scores in the categories listed in Table 10. For each category, the percentage of students with a given grade (A-F) is calculated and the distribution of the grade across all terms for a given category is placed within a chart, herein called a grade distribution chart. This chart provides information on the overall shape of a category's grade distribution across all six terms as well as in individual terms. The score cut-off for each letter grade is provided in Table 11, although for this analysis, pluses and minuses will be aggregated into one letter as the resolution of either does not provide significantly useful information.

The behavior of grade frequency over time is also summarized in another set of charts for each category. One chart allows one to identify trends over the six terms in the frequency of grades for a given category.

Table 10: Category descriptions for the grade components of interest

Category	**Description**
Lab Assignments	Assignments performed during the lab section of each week. These assignments are due before the following lab.
Homework (HW) Assignments	Assignments assigned at the end of each week for students to perform on their own time. They are due during the following week.
Midterm 1	The first written summative assessment taken during week 4 of the course.
Midterm 2	A secondary written summative assessment taken during week 7 of the course.
Final Exam	The final written summative assessment taken at the end of the course during Finals week.
Final Score	The final student score of the course used to determine the final letter grade (see Table __ for Grade Cut-offs).

Table 11: Cut-off for each letter grade

Score Range	Grade	Simplified Grade
$\text{Score} < 60$	F	F
$60 \leq \text{Score} < 63$	D-	D
$63 \leq \text{Score} < 67$	D	
$67 \leq \text{Score} < 70$	D+	
$70 \leq \text{Score} < 73$	C-	C
$73 \leq \text{Score} < 77$	C	
$77 \leq \text{Score} < 80$	C+	
$80 \leq \text{Score} < 83$	B-	B
$83 \leq \text{Score} < 87$	B	
$87 \leq \text{Score} < 90$	B+	
$90 \leq \text{Score} < 93$	A-	A
$93 \leq \text{Score}$	A	

Phase 2.B: Course Mapping

Phase 2.B aims to map the course materials to determine how information and materials flow in the course as the term progresses. This practice may be thought of as an application of the idea behind value stream mapping. This phase begins by applying a technique similar to those described in curriculum mapping techniques in (Jacobs, 1997). This table is a summary of weekly course material, including relevant syllabus sections, weekly learning outcomes, detailed schedule, lectures, textbook readings, lab assignments, homework assignments, and exams (if any). This overview provides the researcher with a better understanding of the progression of materials in the course. Notes are kept in cells next to each category to comment on potential discrepancies in material, such as misalignment of descriptions in the detailed course schedule and course materials, or requirements in course content that go beyond the pre-requisite/co-requisites of the course. Figure 7 demonstrates the general format of this course summary table.

Course Component	**Week 1**	**Notes**	**…**
Syllabus Description	*Verbatim description*		
Weekly Learning Objectives	*Verbatim description*		
Textbook Readings	*Titles of each chapter section*		
Lecture Description from Detailed Schedule	*Verbatim description*		
Actual Lecture Material	*Summarized based on lecture agenda and titles within slides*		
Lab Description from Detailed Schedule	*Verbatim description*		
Actual Lab Material	*Learning objectives and one-line summary of each problem*		
HW Description from Detailed Schedule	*Verbatim description*		
Actual HW Material	*Learning objectives and one-line summary of each problem*		

Figure 7: General format of the course summary table with content

The course summary table is used primarily as an intermediate step towards developing an understanding of the course materials. This summary table is then used along with the course materials to generate a matrix demonstrating the flow of content. Weekly learning objectives are compared against course materials and an "X" is placed where the course material covers content that aligns with the learning objective. It is possible to mark intersections of course materials from before and after a course material's given week (e.g. marking where week 1 or week 3 learning objectives intersect with a week 2 homework assignment), however, this would begin to develop into a large task with decreasing returns on usefulness. To narrow the scope, only assignments and examinations are examined in terms of where they intersect with previous and future weeks' materials. The rationale for this, besides reducing the workload for analysis, is that assignments and exams are the primary avenue for where students are required to apply and practice material; lectures focus on the introduction of new material. Figure 8 demonstrates the format of the content flow matrix in further detail.

This course flow matrix allows one to receive a high-level overview of the progression of content within a course. An expected diagonal of intersections emerges between each week's learning objectives and material. This matrix also allows for the observation of which topics are built upon within assignment materials and where there may be gaps in the reinforcement of certain materials. Finally, this matrix allows one to see where course materials extend beyond material required for a given week.

It should be noted that a primary limitation of this technique is that one cannot see where the course material covers topics beyond those specified in weekly learning outcomes. Thus, a cell for additional notes is added for each assignment where notes may be made for later investigation. The output of this course flow matrix is a visual representation of the general flow of course materials and where issues may be in material coverage.

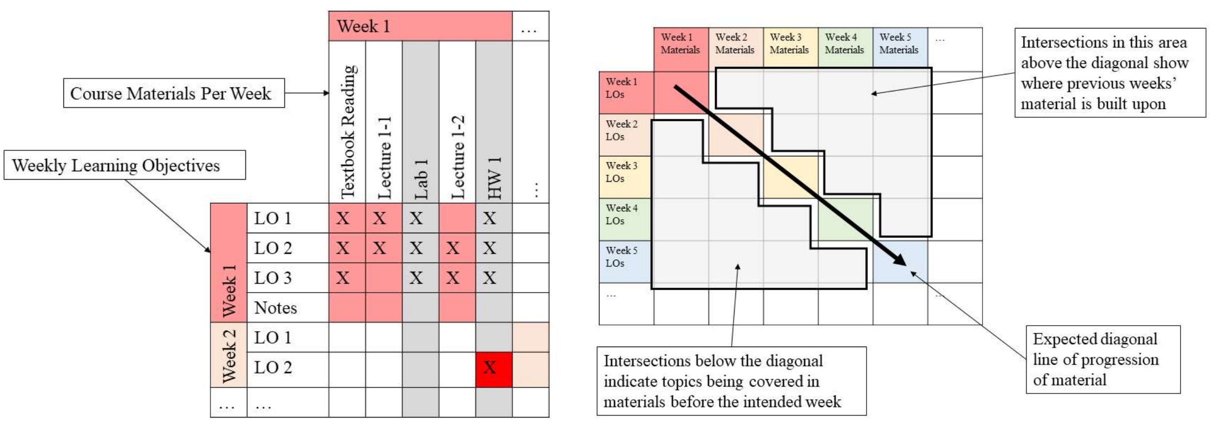


Figure 8: Course flow matrix format – left side shows detailed format; right side shows overview of completed matrix.

Another limitation of this technique is that temporal aspects of the course are not easily visible in terms of when course material occurs. To mitigate this, a timeline is created of the following course components: textbook readings, lectures, labs, homework assignments, and examinations. Time periods to include are: time to work on assignments, lecture/examination days, lab days, due dates, and grading periods. This timeline includes estimated grading times and can show potential issues in the timing of grading feedback.

Finally, for lab and HW assignments, a chart demonstrating assignment grade frequency behavior across individual assignments within the term is generated to highlight trends in assignment performance as the term progresses. This may provide insight as to which assignments cause more difficulty for students and hint at where there may overloading in course content or difficulty.

Phase 2.C: Student Feedback

This third component of Phase 2 aims to analyze data available through the student evaluations of teaching (eSETs) in order to incorporate the student feedback into course improvements. This aligns with both the Specify Value principle in incorporating requirements from the ultimate customer and the Respect for Humanity principle in involving students in improvements. eSETs are administered at the end of each term for all courses at OSU to elicit student feedback on course quality and areas for improvement. This evaluation consists of two standard components. Quantitative data is collected via 12 questions on a six-point Likert scale. The questions and rating conversions are available in Table 12 and Table 13**Error! Reference source not found.** respectively. Additionally, qualitative data is collected using the standard prompt “Please comment about ways to improve instruction” for both lecture and lab sections of the course. Instructors are allowed to generate their own questions on separate pages, but these will not be examined within this thesis.

Table 12: Quantitative eSET prompts with indices and shorthand titles (for reference in the rest of the thesis).

Index	eSET Prompt	Shorthand
1	The course as a whole was	As a whole
2	The instructor's contribution to the course was	Instructor's contribution
3	Clarity of course objectives or outcomes was	Course Objectives
4	Clarity of student responsibilities and requirements was	Clarity of student responsibilities
5	Course organization was	Course organization
6	Availability of extra help when needed was	Availability
7	Instructor's use of various instructional techniques to accommodate differences in learning styles among student was	Accommodate differences
8	Instructor's interest in my learning was	Instructor's interest
9	Instructor's ability to stimulate my thinking more deeply about the subject was	Stimulate thinking
10	Instructor's timely feedback to tests and other work was	Feedback
11	Instructor's ability to develop a welcoming classroom environment for all participants was	Classroom environment
12	Instructor's evaluation of student performance in accordance with course objectives was	Evaluation of Student Performance

Table 13: Quantitative eSET point scale

Rating	Numerical Value
Very Poor	1.0
Poor	2.0
Fair	3.0
Good	4.0
Very Good	5.0
Excellent	6.0
Unable to Rate	N/A

The quantitative data is analyzed by plotting the averaged median quantitative scores over time to observe trends in these scores. The categories may then be prioritized for further analysis in Phase 3 (described in that section).

Qualitative comments are coded by the author using the following method, based on grounded theory (Locke, 2002) and techniques described in (Miles, Huberman, & Saldana, 2014).

First, comments are read and preliminary codes are generated and assigned during this reading. Codes are continuously reviewed, changed, joined, and separated as the reading process continues. The coding process is oriented towards identifying potential issues within the course that may be addressed through improvement. Codes are then combined into larger categories for summative purposes. The codes are validated through spot checks performed by the author's mentor.

For the coding process, it is important to describe the author's background as is relevant to this work so as to provide insight towards their potential biases in the coding process. The author is an undergraduate student at OSU studying industrial and manufacturing engineering with a minor in education. They have both been a student in the ENGR 112 course during Winter 2016 and a teaching assistant in Winter 2017. During this experience with the course, they enjoyed learning the course material but noted that several components of the course could be improved; this experience served as part of the reason for undertaking this project. Bias may thus exist in what the author identifies or highlights as issues during the coding process. The spot checks are performed by an unbiased professor with no ties to the course, although the spot checks are limited in nature. This is why additional analysis of the current state available from course materials, grade distributions, and process knowledge is included throughout this methodology for the purposes of validating evidence.

Once the coding process is complete, the frequencies of codes both within and between categories may be discussed and used for identification of areas for improvement within the course.

Phase 3: Root Cause Analysis

This section attempts to synthesize the results of Phase 2 in order to identify actionable improvements to root causes of issues. To focus the scope, the three lowest performing categories from the quantitative section of the eSETs will be used as the starting point for individual analyses. A combination of the 5 Why's method and Cause and Effect diagrams (Brassard & Ritter, 2010) will be used to structure this analysis, and potential root causes suggested using the evidence from the previous sections of this methodology and the author's process knowledge in the course through a retroactive application of the participant observer methodology (Glesne & Peshkin, 1992), given that the author was once a participant in the course as both a student and TA.. Improvements will then be suggested based on the application of lean principles and practices in tandem with each set of results.

Results

Phase 1: Value Specification

Five weekly learning objectives were identified to have zero intersections with the topics listed in the instructor survey. These five are listed in Table 14 below. Learning objectives one and two (as indexed in Table 14) may be considered necessary components of interacting the MATLAB programming language and basic skills of programming. Learning objectives three and four may also be considered essential skills to developing computer programming abilities. Learning objective five may be considered an important concept in computer science, but, as results from Phase 2.B will show, does not show any actual usage in course assignments. The lack of intersection for these learning objectives may be attributed to their lack of inclusion in the survey options.

Table 14: List of weekly learning objectives with no intersections with topics in the instructor survey.

Index	**Week Number**	**Weekly Learning Objective**	**Notes**
1	1	Find and describe [components of the MATLAB environment]	This is a necessary component of learning the MATLAB programming language as users must learn to navigate the MATLAB development environment to use the program.
2	1	Get input from the user and display the results of calculations in the command window	This is a necessary component of operating the MATLAB programming language as many problems and programs require the user to input values to programs.
3	2	Use the debugger to examine the value of a variable while executing	Learning to use the debugger is necessary for fixing inevitable errors in the coding process.
4	3	Use the debugger to fix if statements and loops	Learning to use the debugger is necessary for fixing inevitable errors in the coding process.
5	4	Local versus global variables	This is an important concept in computer science, but not included in the list of survey topics.

Fifteen topics from the survey were found to have zero intersections with weekly learning objectives. These topics are listed in Table 15. Of these, five topics had no instructors indicate they were important to learn. This indicates that the topics are being properly excluded from the course. Four course topics did not explicitly align with any course objectives; however, results from Phase 2.B indicate that these topics are addressed within certain course assignments. Six course topics, indicated by instructors to have importance within their courses, are not covered within the weekly learning objectives. These topics may have the opportunity to then be added to the course if the instructor considers them important enough to be added and enough capacity is available within the course to do so.

Table 15: Survey topics with no intersections with weekly learning objectives.

Index	Topic	Survey Count
1	Trees	0
2	Hash Tables	0
3	Priority Queues	0
4	Lists	3
5	Cells/Structures	3
6	Volumetric	3
7	Grid Structures	3
8	Writing their own optimization programs	0
9	Data reduction – e.g. PCA, machine learning	0
10	Symbolic equation solving	4
11	Writing their own code to perform simulations	4
12	Filtering/signal processing	4
13	User interface – menus and buttons	4
14	Simulation (set up and run)	5
15	Data fitting (2D and higher)	7

One topic, “Lambda/anonymous functions”, had no instructors indicate it to be a desired topic for their courses. This topic, however, has two intersections with weekly learning objectives. This result could be indicative of a non-value adding component within the course; however, anonymous functions are a necessary and useful component of working with the concept of functions. It is possible that this topic was not marked by instructors due to a lack of description on the component instead.

The results of this analysis provide preliminary insight on value-adding topics that may be added and removed from the course. This analysis is limited, however, in at least two ways. First, it only looks at topics from the voice of the instructors in downstream courses, without regard for information from other stakeholders. Research has also shown that such downstream experts of content knowledge may be unaware of some of the pedagogical needs of students, and their selection of topics, while in alignment with the organization of the content domain, may be in conflict with the developmental needs and learning capabilities of students (Nathan, Koedinger, & Alibali, 2001; Nathan & Petrosino, 2003). Second, the number of instructors indicating the importance of a topic is not examined, only whether or not a topic was marked at least once in terms of importance. This analysis could be extended using the survey counts for each topic to prioritize topics for teaching. In the event that the class is teaching too much material (the lean waste of overproduction), this prioritization could provide a guide as to what topics may be unimportant enough to be removed. This work would be ultimately left to the course owner's discretion.

Phase 2: Current State Analysis

Phase 2.A: Course Grade Distributions

Results for Phase 2.A are available in Appendix C. Grade distribution charts for the lab and homework assignments display relative stability over time, with the vast majority of the class receiving A’s for both types of assignments. Grade frequency over time for these assignments has remained relatively stable over time as well.

Grade distributions and frequencies over time for exams show much greater variation in results term to term, with larger percentages of the class receiving D’s and F’s compared to assignments grades. The distributions appear to vary in shape from term to term, with some unimodal and others more u-shaped bimodal. Of particular note are final exam results which recently have seen an increase in the percentage of F grades in the pass two terms and a sharp decline in A’s, indicating poorer performance on final exams.

Final course grade frequencies display no obvious trends besides slight increases in the frequency of A’s and F’s and slight decreases in all other grades’ frequencies. This indicates an increasing accentuation of a bimodal u-shape in the grade distributions. The average percentage of F’s across the six terms is 14.41%, approximately half of the 28% three-year average reported earlier. This may indicate an improvement in course performance across the three years, although it should be noted that the three-year average is a Drop-Withdrawal-Failure-Unsatisfactory rate. The grade data provided does not capture students who dropped or withdrew from the course, meaning the average DWFU rate is likely higher than the average failure rate reported here. It should also be noted that the College of Engineering requires a grade of C (not C-) or higher in all core classes in order to be admitted to the pro-engineering school. This means that students with a grade of C- or worse end up having to retake the course, leaving them with a de facto failure. Taking this into account, the de facto failure rate for the course based on grades alone is likely closer to 20%.

Reasons for variation in course grades and exams may be attributed to a number of factors, including variation in student characteristics, instructor differences, class size differences, variation in examinations, and small assignment variations. ENGR 112 is typically recommended for students to enroll during Winter and Spring terms. Fall terms, as noted by the course owner, typically consists more heavily of students retaking the course. Three different instructors taught the course over the six-term period, and class sizes have been steadily increasing each year, although Fall term tends to see much fewer students than Winter and Spring. Finally, examinations, while supposedly held to approximately equivalent difficulty across terms, may see changes in difficulty due to the question differences. These points of variation are speculative at this point, however, and there appear to be too many sources of variation at this point to make any definitive claims with respect to the course so far.

Phase 2.B: Course Mapping

The entire course summary table is attached as a fold-out in Appendix F. An example excerpt from the matrix is shown in Figure 9: Example of summary matrix. This figure shows the written summary of the material in each component described in the leftmost column with preliminary notes in the rightmost column. Notes as to some of the findings in this matrix as recorded in the notes section are as follows. Many topics covered in the course and assignments problems require conceptual knowledge that is not part of the course prerequisites. These include physics concepts in one of the term-long problems, differential notation in another term-long problem, calculus concepts such as differentiation and integration, series notation and calculation, matrix algebra, Gaussian curves, miscellaneous engineering concepts such as stress or Kirchhoff's Voltage Law and structs in MATLAB. While it may not be necessary to fully understand a concept or to have completed an entire university course on the subject to apply it coding practice, the acquisition of an external, non-programming concept on top of learning programming skills and problem-solving may strain a learner's cognitive abilities. This may present itself as an example of the overproduction of information, a lean waste.

Other issues include gaps in the application of term-long problems, such as the Teacup problem, which has one required problem and HW 2 and the next required problem in HW 7. This discontinuity in the problem progression may present itself as an interruption to course flow. Lecture slides appear to be dense and examples of problems and their solutions include walls of code, hinting at the overproduction of material within lecture slides. However, it is difficult to determine how slides were actually used during lecture given the retroactive nature of this study. There also appear to be inconsistencies in the naming of lectures in the detailed schedule compared to what is present in the actual course files, but this is of minor consequence. Finally, examinations require the reading and description of written scripts by students and the handwriting of code on paper. There is little opportunity for the practice of this skill throughout other course activities, however, besides on small lab quizzes. This again hints at possible misalignment between assessment techniques and the practice of programming; however, this is again outside the bounds of this thesis to address.

Moving to the course flow matrix, shown in Figure 10, one can observe that the majority of assignments are in alignment with weekly learning objectives. The overflow of topics in assignments is easily visible in weeks 1, 4, 6, and 7. Additionally, it can be noted that topics from weeks one through four are consistently covered in all weeks. Topics in week five appear to be reinforced through week seven, and the remaining weeks show little reinforcement of topics following week four. This makes subjective sense given that the first four weeks cover foundational knowledge in computer programming (knowledge of variables, control structures, functions, etc.) which are necessary for more advanced programming work, while the following weeks cover more specific areas of functionality. This matrix thus provides a useful at-a-glance look at where material is being reinforced throughout the term and which material may stand more individually. This appears to be in line with the practice of value stream mapping, which is to map the value stream in a way that is easy to see the flow of material through a process and easily identify problems, non-value adding processes, and opportunities for improvement.

The course timeline, shown in Figure 11, provides additional insight as to the nature of course flow. A grading period of one week is assumed for all assignments in this analysis. Observations are as follows. First, there may be up to three assignments for a student to work on at any given time. Second, given that there is a one week grading period, Lab 1 may be returned to students with feedback after Lab 2 is turned in. Similar issues are more likely for homework assignments. This length of grading periods could also result in delays in when feedback may be delivered to students during synchronous course time (lecture or lab) on assignment performance. For example, a grading period of this length would mean that Lab 1 results would not be able to be addressed until week 3 lab or week 3 Friday lecture. Feedback for HW 1 concepts would not be able to be addressed until week 4 Monday lecture. These delays are all an example of the lean waste of waiting. Given the author's process knowledge of the course, it is known that it may take even longer for the grading of certain assignments to be completed. The learning management system's ability to grade materials requires all assignments to be graded before grades and feedback may be released. This means that delays by one grader can prevent all assignment feedback for all students from being delivered in time.

Finally, for grade results of assignments, Figure 12 and Figure 13 show the respective lab and homework grade frequencies of various assignments over the Spring 2017 term. Both show the behavior of having the majority of assignments with a grade of A, but the frequency of F's for both assignments appears to increase as the term continues, indicating an accentuation of the U-shaped distribution of grades within the assignments. There are many factors which may contribute to this behavior, however, such as delays in feedback, a large quantity of assignment materials, or the coinciding of the increase in F's in Lab 4 and HW 3 with the midterm in week 4.

These results provide information related to the course material flow which will inform improvement development efforts in Phase 3.

	Week 1	Notes
Syllabus	Introduction & historical overview Introduction to numeric, character, and Boolean data types Arithmetic precedence rules & MATLAB intrinsic functions MATLAB script files Input/output operations (DISP, FPRINTF, INPUT)	LO 1, 2
Weekly Learning Objectives	Find and describe the following in the MATLAB environment: The command window, where you can type commands directly into MATLAB The script window, where you can edit and save scripts The variable window(s) for displaying values of variables The history window and how to save and repeat commands Create, and run, a script that calculates a given equation Document the script using comments Understand (and use) the following commands: clear, clc, sin, cos, log, exp, ^, pow, sqrt Use sin and sind appropriately Get input from the user and display the results of calculations in the command window Use the following commands: disp, fprintf Print out the value of a variable using %0.0f Explain what the ; does at the end of a line	
Textbook Reading (zyBooks)	1.1 Solving engineering problems with MATLAB 1.2 MATLAB background 1.3 MATLAB and the interpreter 1.4 Computer basics 1.5 A brief tour of a computer 1.6 Basic input: The input() function 1.7 Basic output II: fprintf() 1.8 Basic output 1: disp()	Primarily corresponds to lecture 3
Lecture (Mon) - Detailed Schedule	1-1 Intro 1-2 Variables Equations	
Lecture (Mon) - Actual	1-1 Intro-Sami Introduction to course, what is learned, format and flow, expectations and keys to success, assignment format, To-Do before Lab, Classes where MATLAB is used, About your instructor. 1-2 Variables Equations Variables What they are Syntax rules for names Declaring and assigning a value to a variable Arithmetic Operators Precedence (when to use parentheses) 3 examples.	Can't tell what was actually used to teach. Lecture states that weeks 1-2 are difficult, weeks 3-5 are very hard, and then it gets more manageable from there. Already this hints at unevenness, or mura.
Lecture (Fri) - Detailed Schedule	1-3 Functions IO Input, sin/cos, sqrt, exp, nthroot 1-4 Pseudo Code	

Lecture (Fri) - Actual	1-3 Functions IO Matlab built-in equations and variables: sqrt, nthroot, exp, abs, log, log10, factorial, round, floor, ceil, mod, pi, inf, NaN, all trig functions. Input from user Printing output to the command window Omit ;, disp(), fprintf Script files 3 examples 1-4 Pseudo code Comments and pseudo code Steps to go from a word problem to Matlab code Explicit Problem Solving Strategy explanation 2 examples	
Lab - Detailed Schedule	MATLAB windows, create a script, create variables, plot, write equation	
Lab - Actual	LO: What is a script file? What is the difference between a script file and the command window? How do you create variables? How can you examine the values of variables? How do you use the debugger? 1. Basic variable manipulation in command window and script. 2. Use linspace to create an array of variables and use various display commands for arrays 3. Plot a circle using cos and sin 4. Solve 2 equations 5. Demonstrate usage of the debugger	Linspace and plotting is in week 2. Students have not learned anything about arrays or plotting yet. The due date of 4/10 confirms this as there would be no time to receive additional lecture.
HW - Detailed Schedule	HW1: Write equations with scalar values	
HW - Actual	Goals: What is a script file? What is the difference between a script file and the command window? How do you create variables? How can you examine the values of variables? How do you create equations, pass them variables, and print out the results? How do you display the values of variables? 1. Solve two equations. 2. Solve a projectile motion problem: find landing location (x, y) and time of hitting the ground 3. Find the volume of a hollow sphere. 4. Teacup 1 - Translate and rotate points about the origin 5. Epidemic 1 - Model two steps of an epidemic 6. Euler leaf 1 - Model taking time steps of a leaf in the wind	Some redundant learning objectives from lab 1. Assume that most learning objectives are integrated into the problem solving system. Problem 2 uses projectile motion concepts which may not have been learned yet. Teacup problem is hard to visualize and the term canonical position can confuse people (as it did Winter 2017). Epidemic problem uses differential notation which students may not have learned yet Euler problem uses physics concepts which may not have been learned well yet Teacup and Euler problems are over a page long, and term long problems have a long introduction.

Figure 9: Example of summary matrix. This figure shows the written summary of the material in each component described in the leftmost column with preliminary notes in the rightmost column

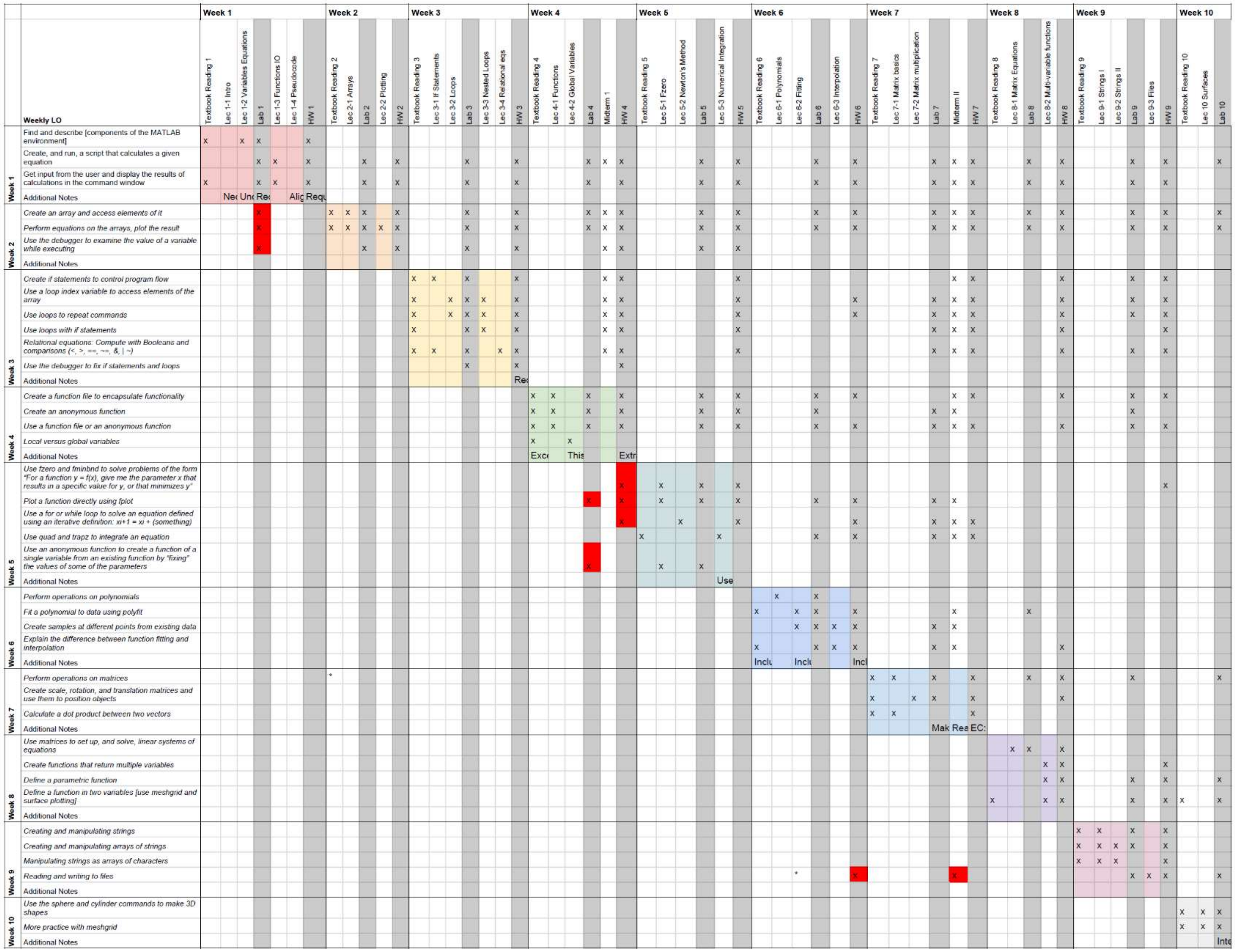


Figure 10: The content flow matrix for ENGR 112 as of Spring 2017.

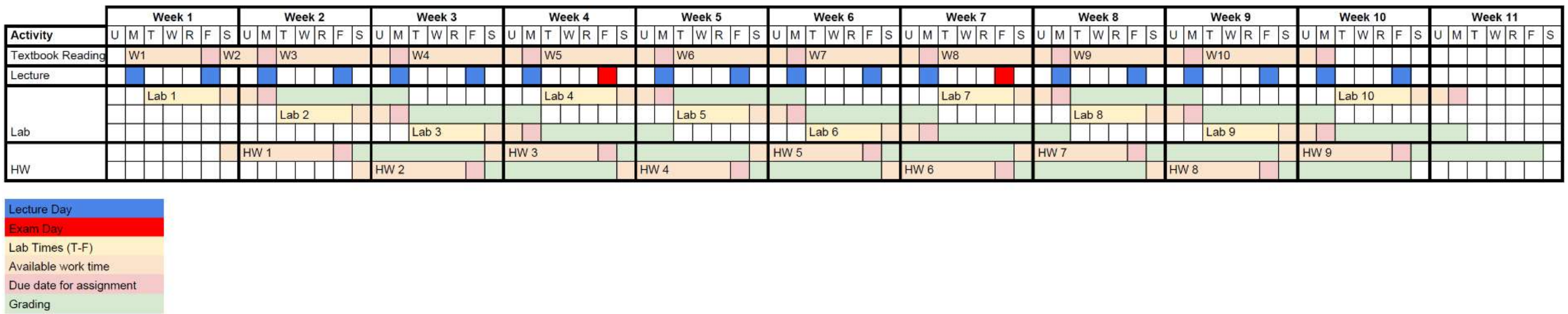


Figure 11: Course Timeline

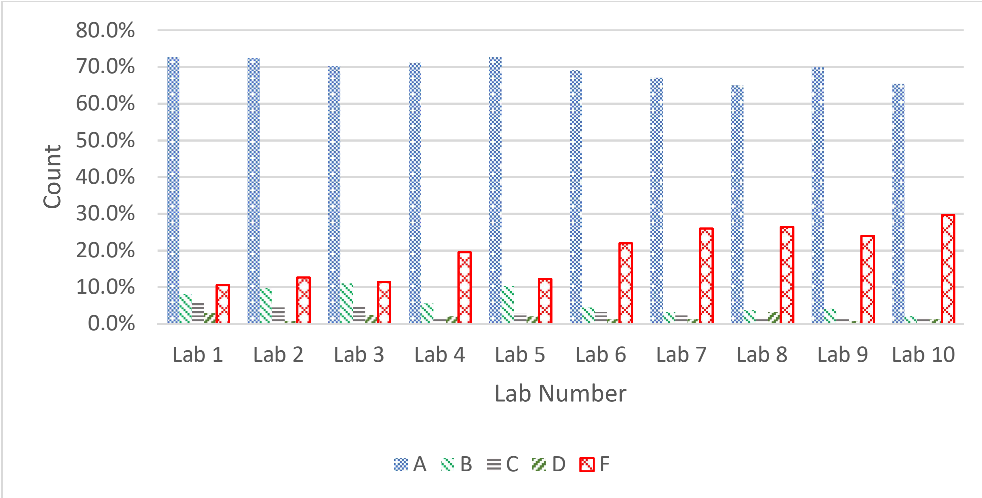


Figure 12: Lab Grade Frequencies for Spring 2017

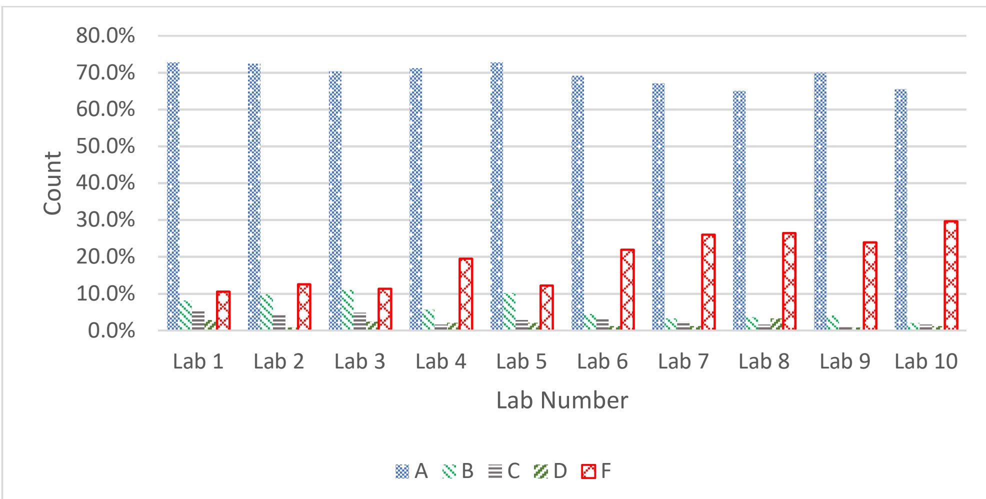


Figure 13: HW Grade Frequencies for Spring 2017

Phase 2.C: Student Feedback

Table 16: Average median eSET scores of ENGR 112 over the six terms reviewed in this study, arranged in order of lowest to highest score.

Prompt	Average Median Score over 6 terms
Feedback	3.92
Accommodate differences	4.03
Course organization	4.15
Stimulate thinking	4.18
As a whole	4.22
Evaluation of student performance	4.23
Instructor's interest	4.23
Instructor's contribution	4.27
Course objectives	4.28
Classroom environment	4.32
Clarity of student responsibilities	4.33
Availability	4.48

Table 16 above shows the data for average median scores over the six terms reviewed in this study in order of lowest to highest, and Figure 14 demonstrates how median scores have fluctuated across the six terms using the top 3 and bottom 3 eSET categories' scores. While there can be fluctuation and cross-over between scores over the six terms, the bottom three generally stay at the bottom of the graph, with timely feedback consistently at the bottom of every term. The three worst performing prompt categories will be focused on to help narrow the scope of analysis and improvement. The prompt regarding timely feedback to student work is found to have the lowest average median score (3.92). The second lowest concerns the instructor's ability to accommodate differences in student learning styles (4.03). Literature on accommodating learning styles in teaching, however, has been found to have little impact on student learning outcomes (Willingham, Hughes, & Dobolyi, 2015), however, and so this category among the eSETs will not be addressed. The next lowest categories are the quality of course organization and the instructor's ability to stimulate thinking on the subject. These categories will provide the basis for the root cause analysis to come in phase 3.

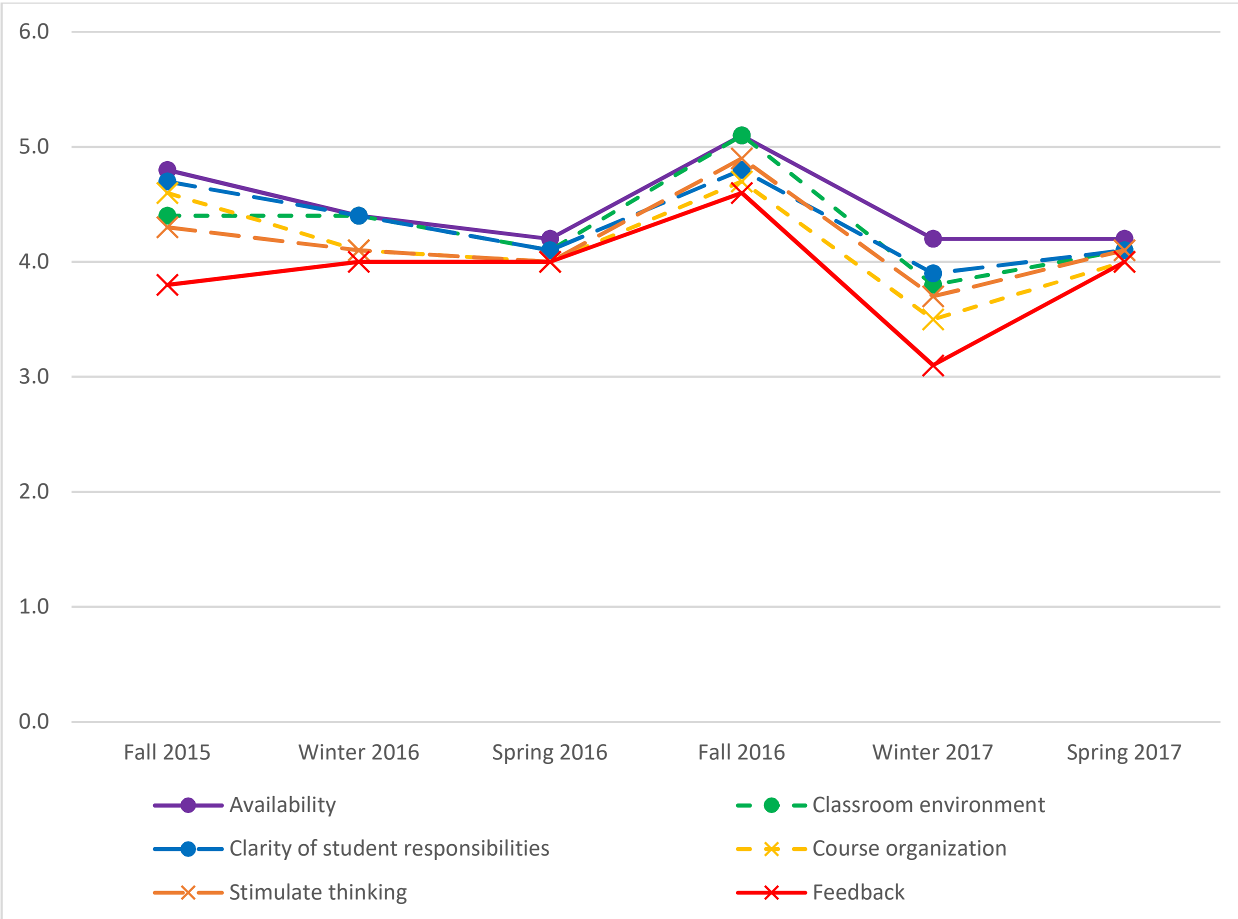


Figure 14: Median eSET scores across the six terms for the top 3 and bottom 3 categories.

Qualitative data analysis of the codes yielded 38 preliminary codes and 430 coded responses. These codes were organized into ten categories and are described in Appendix E. Note that some of these codes may be a variant of a certain topic, such as Lecturer Quality being "Good", "Bad", and "Acceptable". Table 17 lists the 10 categories with their quantities of coded responses and the corresponding percentage of all coded responses, with Figure 15 demonstrating these percentages graphically. Over 75% of the coded responses lie within the first five categories, warranting some basic frequency analysis of within category coded responses. Figure 16 to Figure

20 display the percentages of coded responses within a given category. Appendix Figure E.1 displays all codes and their corresponding percentage of all coded responses in order of largest to smallest.

Table 17: Quantity of coded responses per category and corresponding percentage of all code responses.

Category	Quantity of Codes	Percentage of All Coded Responses
Lecture Characteristics	78	18%
Course Flow	69	16%
Opinions on Overall Course	61	14%
TA Characteristics	61	14%
Feedback & Availability	61	14%
Assignment Material Issues	33	8%
Instructional Components	25	6%
Lab Characteristics	23	5%
Voice of the Student	11	3%
Course Policies	8	2%

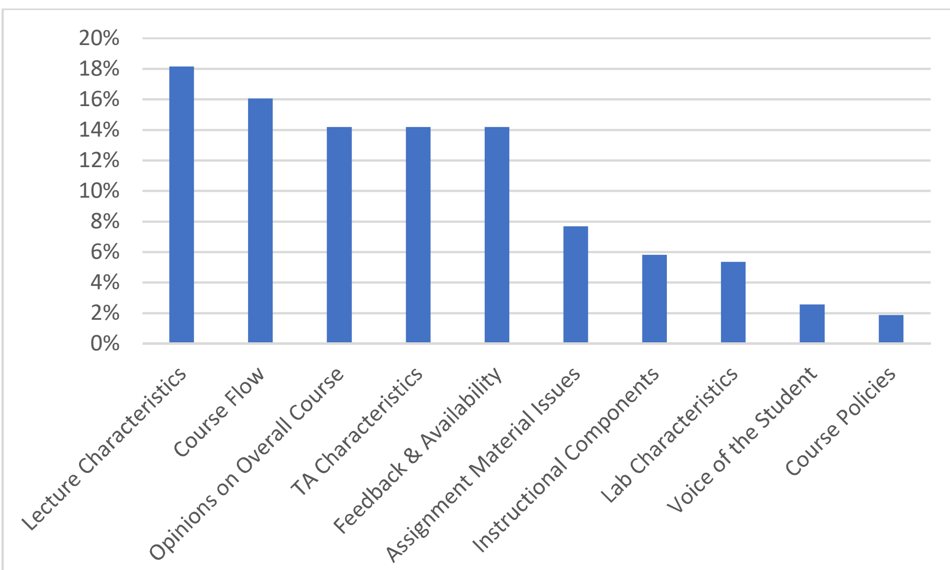


Figure 15: Percentage of coded responses within the ten categories

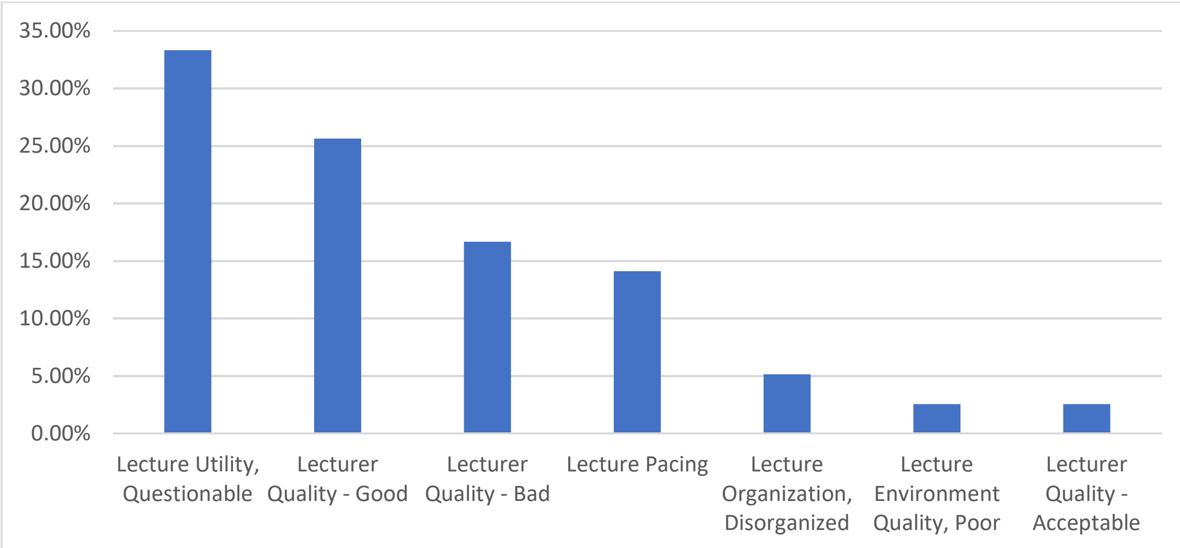


Figure 16: Within-category code frequencies of the “Lecture Characteristics” category, which accounts for 18% of all coded responses.

Figure 16 displays the code frequencies related to “Lecture Characteristics.” Of these codes, the code “Lecture Utility, Questionable” has the highest percentage of approximately 33%. This code occurs across all six terms This code relates to students noting that going to lecture does not appear to benefit their learning, with material being unhelpful or confusing. Other codes relate to lecturer quality, with more positive reviews than negative, the pacing of lecture, usually inconsistent or too fast, and lecture organization or environment quality.

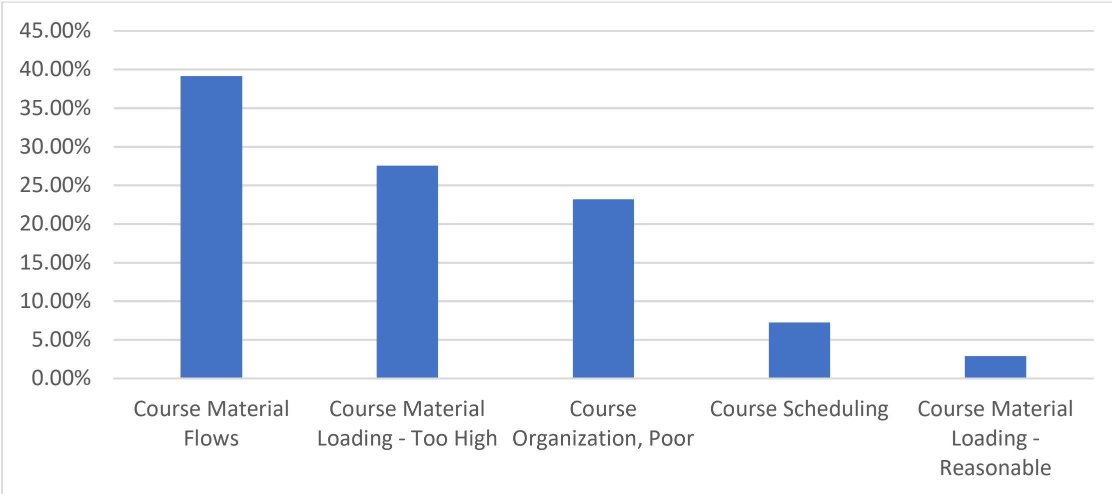


Figure 17: Within-category code frequencies of the "Course Flow" category, which accounts for 16% of all coded responses.

Figure 17 displays the code frequencies for the "Course Flow" category. The highest percentage of coded responses fell under the "Course Material Flows" code, which relates to the progression of content and how components of the course connect to each other (e.g. lecture preparing students for lab). Other codes within this category relate to students noting overall poor course organization and large course material loading (e.g. assignments requiring too much student time).

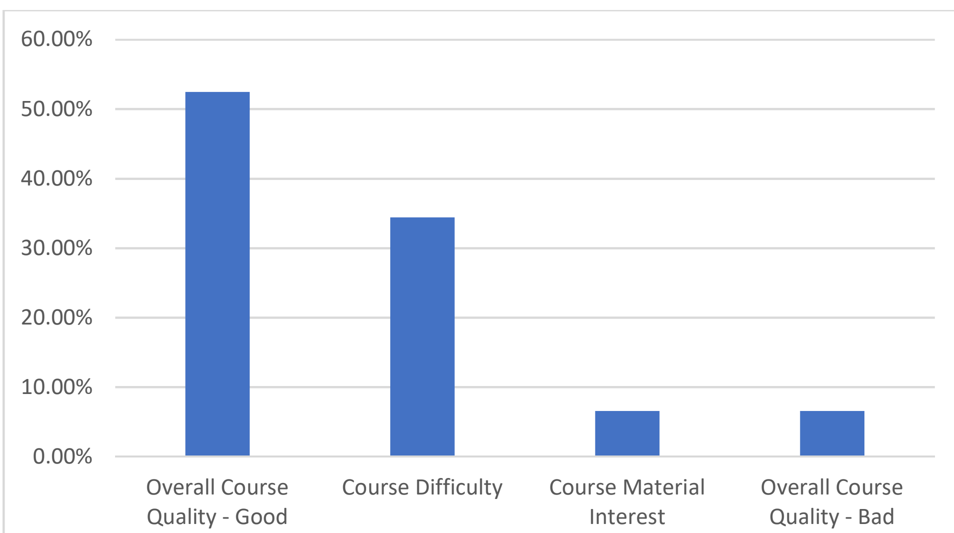


Figure 18: Within-category code frequencies of the "Opinions on Overall Course" category, which accounts for 14% of all coded responses.

Figure 18 displays the code frequencies for the "Opinions on Overall Course" category. This category contains codes which relate to general student views on the overall course. The majority of codes within this category denote that the overall course quality is good. This demonstrates that not all students view the course negatively, although as other categories show, there are components of the course which have room for improvement.

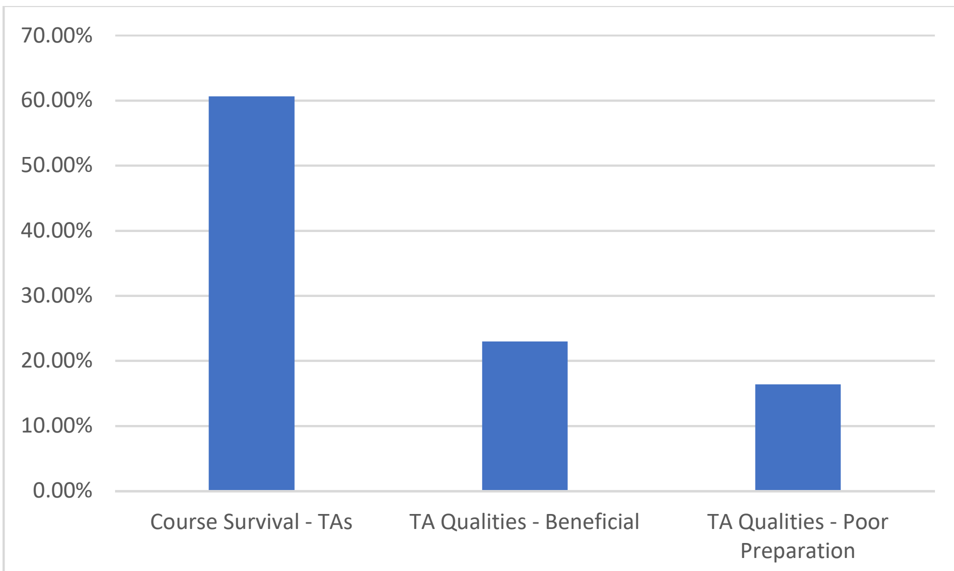


Figure 19: Within-category code frequencies of the "TA Characteristics" category, which accounts for 14% of all coded responses.

Figure 19 illustrates the code frequencies within the TA Characteristics category. The most frequent code is "Course Survival – TAs", wherein students note that they would not have been able to make it through the course without support from the TAs. This code consistently appears across all six terms within the lab section comments, making it a consistent component of the course, and is also the most frequent code of all the coded responses, as shown in Figure __. The second most common code is "TA Qualities – Beneficial", where students generally remark positively on TAs. The third code, "TA Qualities – Poor Preparation", denotes instances of TAs being unprepared for lecture or unprofessional.

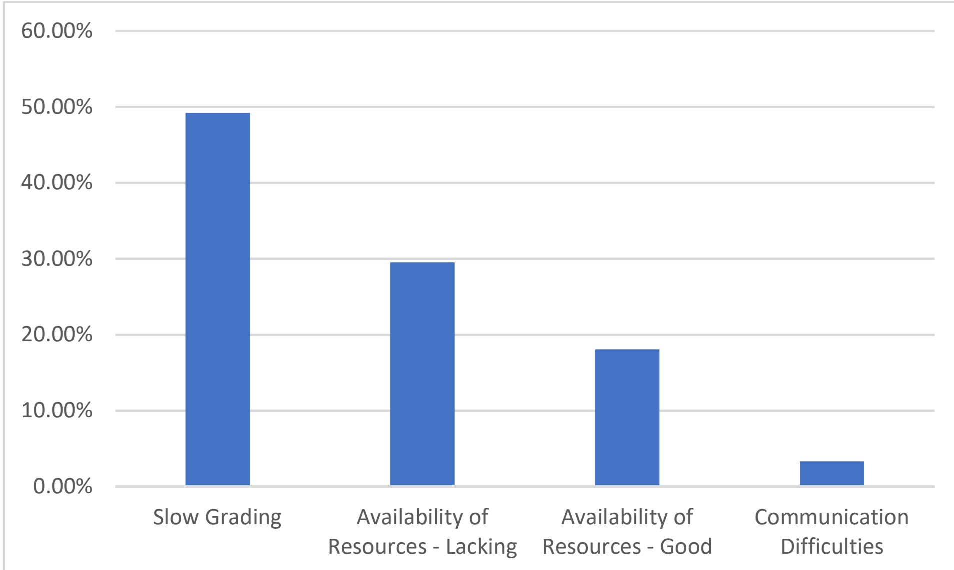


Figure 20: Within-category code frequencies of the "Feedback and Availability" category, which accounts for 14% of all coded responses.

Finally, Figure 20 illustrates the code frequencies within the Feedback and Availability category. The most frequent code is "Slow Grading", where students remark on the slow turnaround for grades, with some homework assignments being weeks behind in grading. The code "Availability of Resources – Lacking" primarily relates to a lack of availability of TAs during office hours, or sudden cancellations in office hours. The code "Availability of Resources – Good" describes students noting that help for assignments or online materials were readily available as well.

Phase 3: Root Cause Analysis

The following section details a preliminary root cause analysis of the three worst performing quantitative eSET categories discussed in the previous section, along with potential improvements based on lean practices.

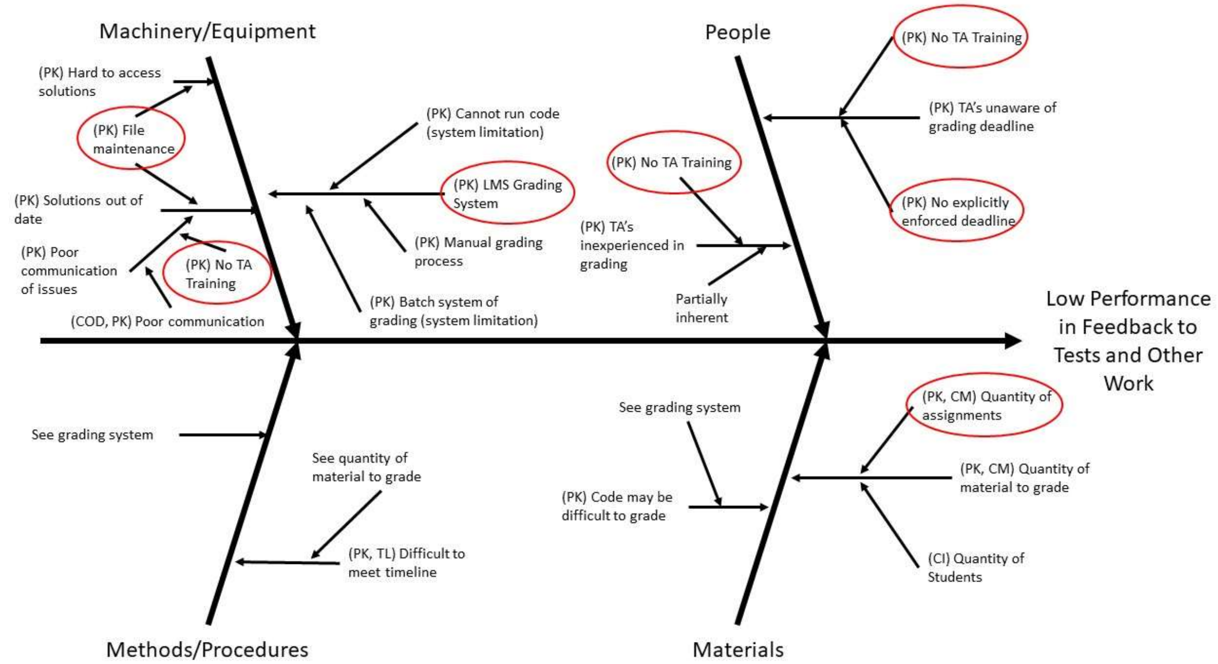


Figure 21: Figure __: Cause and effect diagram of low performance to feedback to tests and other work. Key for evidence of cause: PK – Process Knowledge; CM – Course Mapping; MTX – Content Matrix; TL – Timeline; COD – Codes; CI – Course Information

Figure 21 is a cause-and-effect diagram for the first quantitative category concerning instructor feedback to tests and other work. The source of a cause is shown in parentheses, the key to which is provided within the captions. Circled diagrams represent actionable root causes wherein improvements may be suggested. Potential causes are a lack of TA training, limitations in the grading system, lack of explicit enforcement in grading deadlines, a lack of proper file maintenance, and the sheer quantity of assignments.

A lack of TA training can slow a TA's ability to grade as they are unfamiliar with the procedure and are unaware of how to handle issues that arise through the grading process (e.g. missing rubrics, items not covered within the rubric, etc.). The grading system only allows for viewing pdfs of student submissions, meaning that it is difficult to run a student's code on one's computer without downloading all script files and running them separately. Likewise, a grader must manually read through a script to identify errors, which may be cumbersome given larger assignments. While not explicitly mentioned as an issue by students, it is also known that the grading system makes it difficult to capture specific errors made by students in assignments; only quantitative scores are available for the assignment as a whole. This can make it difficult for an instructor to diagnose issues in students' learning and develop appropriate feedback. A lack of explicit enforcement of grading deadlines may lower the priority of grading, and a lack of file maintenance can result in misplaced or un-updated rubrics which lead to inconsistencies or delays in grading. Finally, the quantity of assignments to grade obviously results in a need for more grading time; the ability to address or reduce this quantity, however, must be left to the instructor's discretion.

Improvements that can be made are as follows: some basic training for TAs that demonstrates how to grade example assignments, sets a standard process for handling abnormalities in grading (e.g. missing assignment rubrics, student errors not addressed in grading rubrics, etc.), and sets explicit deadlines for grading and processes to mitigate TAs falling behind in grading can support TAs in their work. This aligns with the respect for humanity principle of lean, investing in employees—your best resources who understand the process best.

An improvement to the grading system may also be recommended. Ideally, the grading system would allow a grader to run and debug student and capture statistics as to what specific issues are being made by students. Automatic grading systems may also be investigated, although the complexity of assignment problems may make automation difficult. Feedback would ideally be delivered immediately to students, or at least before the next assignment is due. In alignment with lean principles and practices, an improved grading system would attempt to improve the flow of the grading process to allow feedback to reach students just in time for their next assignment. Automatic grading aligns with quality at the source, allowing assessment to occur within the process. Having statistics available as to student errors may allow the instructor access to data which they may use to continuously improve student learning.

Disorganized or unmaintained assignment files may be addressed using practices found under the zero defects principle, wherein practices such as 5S may be used to organize and remove unnecessary/outdated files, and total productive maintenance plans may be used to keep assignments current.

Finally, a reduction and focusing of assignments may reduce the total processing time for grading in general. The reduction of material is akin to the reduction of inventory in lean production systems. To do so would require instructor discretion and further research into what assignments help achieve the most learning (i.e. are the most value-adding to students).

The overall grading process, viewed through the lens of lean, aims to provide feedback to students Just-in-Time for when they need it. This is how Emiliani (2015) applies the JIT practice to his work in teaching as well. The act of providing feedback just-in-time for students has been implemented as a pedagogical technique as well in Just-in-Time Teaching, or JITT (Simkins & Maier, 2010). Providing feedback in this way has been shown to improve student learning, even within the computer science context (Bailey & Forbes, 2005). Even more radical feedback systems may include in-person grading, where the grading process occurs right when the student turns in the assignment; however, in a study of the usage of in-person grading in computer science classes, while students generally view this system more favorably in comparison to traditional grading systems, there did not appear to be any gains in actual accomplishment of learning objectives (East & Schafer, 2005). The extent of improvement of the feedback system will rely on the instructor's resources and it will be limited by logistics and technology; alignment with Flow and the Just-in-Time practice of lean, however, can prove to be a worthwhile goal given the successes afforded to both manufacturing systems (with applications of lean manufacturing) and the classroom (as with JITT).

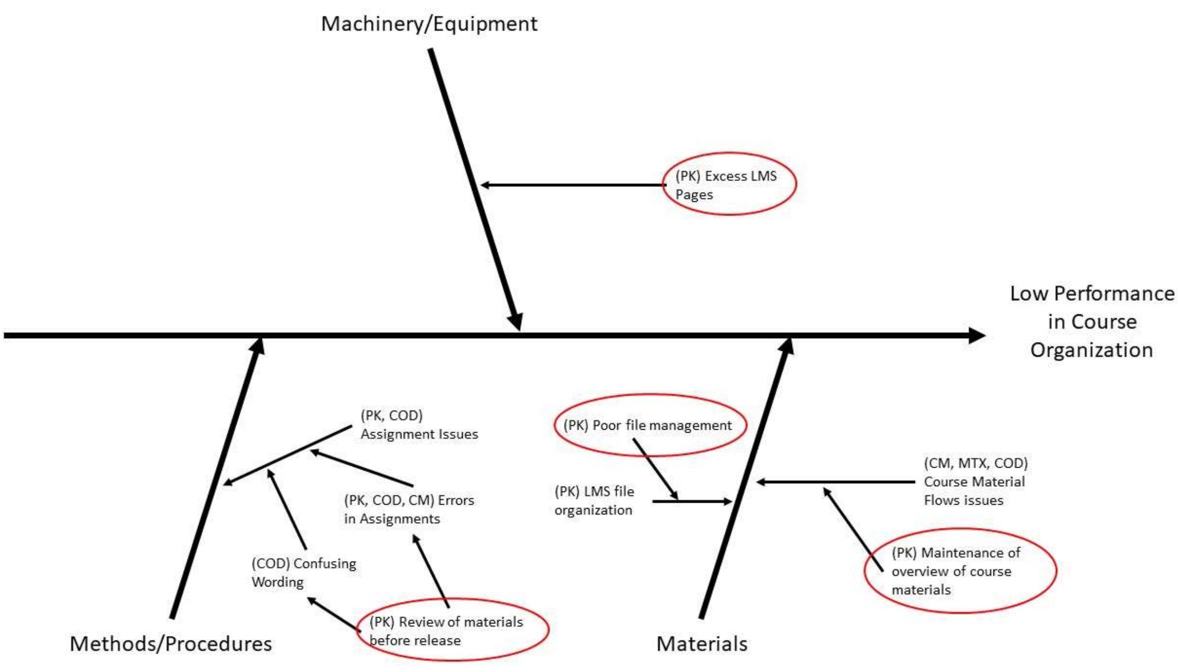


Figure 22: Cause and effect diagram of low performance in course organization. Cause and effect diagram of low performance in course organization. Key for evidence of cause: PK – Process Knowledge; CM – Course Mapping; MTX – Content Matrix; TL – Timeline; COD – Codes; CI – Course Information

Figure 22 displays a cause-and-effect diagram for the second quantitative category of interest: course organization as a whole. This category is more difficult to analyze given the subjective nature of what may be considered "organized." Student responses in the eSETs noted general disorganization, but less detail was provided as to what specific components were considered disorganized. What is displayed here are potential conjectured causes, but further investigation would be required in order to validate this.

A primary issue noted by students in the qualitative data were course material flow issues; specifically, lab and homework assignments were covering topics that had not been covered yet in lecture. An examination of the content flow matrix can confirm this for a few assignments such as Lab 1 and Lab 4. Reasons for these inconsistencies may be due to an overall lack of maintenance over course materials or recording an overview of the course materials in a curriculum map. Issues in assignments in terms of wording and errors may be attributed to a potential lack of review in the release of new assignment materials.

Addressing these issues through lean would require maintenance of course materials through a combination of TPM and retaining updated value stream maps (curriculum maps). Total productive maintenance and the implementation of review process for assignments may help in reducing and maintaining organization of assignment files.

Maintaining current versions of value stream maps of the course allows the instructor to retain an overview of the course and identify issues in course flow. This curriculum map may then also serve as a foundation for which improvements to the course may be made. Recording and maintaining updated curriculum maps aligns with the practices described in Jacobs (1997), whereby maintaining a picture of the actual state of the course as the school year progresses (not simply what is planned or written in the syllabus) allows an instructor or curriculum designer to develop a feasible future state. This is akin to Womack & Jones' (2010) discussion of value stream mapping, where they state that there are three states of your system: what you think is happening, what is actually happening, and what your future state will be. This parallels with the classroom system: what you think is happening is planned in your syllabus, what is actually happening occurs every class over the term, and what the future state will be is the next improved iteration of the class following an analysis of the current state of the system.

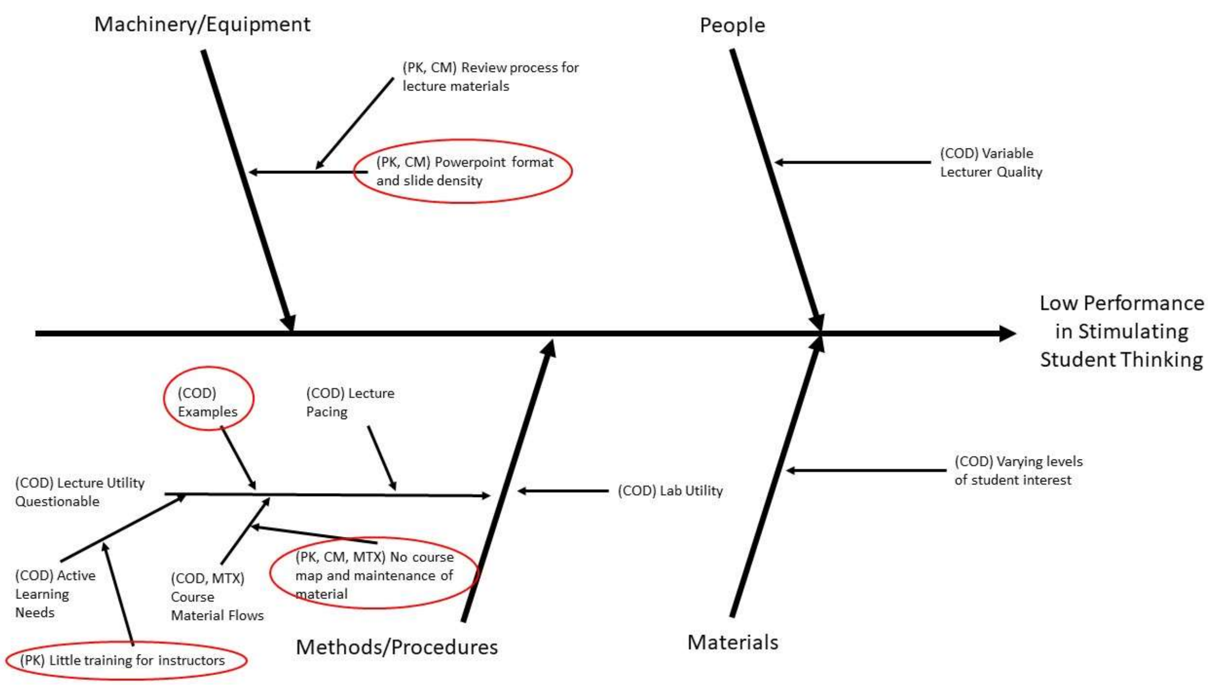


Figure 23: Cause and effect diagram of low performance in stimulating student thinking.
Key for evidence of cause: PK – Process Knowledge; CM – Course Mapping; MTX – Content Matrix; TL – Timeline; COD – Codes; CI – Course Information

Figure 23 illustrates the cause-and-effect diagram for the third category of interest: stimulating student thinking. Causes appear to point towards course material flows (discussed previously) and issues in lectures. It is difficult to fully review and analyze the effectiveness of lectures given the retroactive nature of this study; however, student feedback from the qualitative data denote a need for increasing the usage of active learning techniques and examples throughout during lecture, as well as somewhat difficult to read slides. It may be conjectured that the lack of active learning techniques is attributed to a lack of formal pedagogical training for instructors, which is not a fault of instructors themselves given the lack of mandatory training at the university.

Given the lack of resolution towards the nature of lectures, it is difficult to suggest any improvements. To fully develop a list of active learning pedagogical techniques is beyond this study’s capabilities; however, work in Ott, Robins, & Shephard (2016) and Vihavainen, Airaksinen, & Watson (2014) may provide a starting point for potential practices to implement. It can be recommended to develop and mandate instructional training programs for instructors at the university. Such an action aligns again with lean in Respect for Humanity, investing the growth and training of employees. The ability to implement such a program, however, may be far beyond the author’s scope of influence.

Table 18 summarizes the list of issues, potential improvements, and their alignment with lean principles and practices.

Table 18: Summary of root causes of issues, potential improvements, and alignment with lean principles and practices.

Root Cause	Potential Improvement	Alignment with Principles/Practices
No TA Training	TA training for grading, orientation	Respect for Humanity, Employee Training
LMS Grading System	Find grading system that can run code, grade automatically, and collect data	Zero Defects, Quality at the Source, Flow, JIT, Continuous Improvement
File maintenance/management	Organize files and institute procedures for maintenance	Zero defects, 5S, Standardization, TPM
No explicitly enforced grading deadlines	Explicitly enforce grading deadlines, have trainings	Zero Defects, Standardization
Quantity of assignments	Consider focusing and reducing assignment quantity	Flow, Small batch sizes
Excess LMS pages	Consolidate/Remove unneeded pages	Zero Defects, 5S
Maintenance of course material overview	Maintain a course map of materials	Value Stream Mapping
Not reviewing materials before release	Have at least one person review documents before release	Zero defects
Instructor training, esp. wrt active learning	Implement program at university to formally train instructors in pedagogy and teaching	Respect for Humanity, Employee Training
Example usage in lecture	Increase usage of examples, consider usage of active learning techniques such as JITT or Peer Instruction	Specify Value, Pull
PowerPoint format and density	Consider reducing PowerPoint density	Flow, Small batch sizes

Conclusion

This study undertook the effort to analyze an introductory engineering course at an institution of higher education to implement lean principles towards the goal of course improvement; the output was a list of recommendations for lean improvements, with some focused on improving the standardization of course materials and organization and improving data collection systems and others on tracking the flow of course content and materials and improving the turnaround of feedback.

Limitations

This study was not without limitations. First, though this thesis began with a review of the literature that provided results relating to the implementation of lean in curricular and pedagogical techniques, the extent to which those results could be implemented in this work proved to be limited. This was due in part to resource limitations and the retroactive nature of this work. Resource limitations were primarily in personnel available to undertake this task and the time frame within which the study was undertaken. The retroactive nature of the work obscured the actual activities which took place during class time, leaving only secondary sources (student feedback, physical course materials) for analysis.

The value specification process was limited through the usage of informal survey information from instructors in upper level courses. This provided only one source of input towards shaping course content, with no input from other stakeholders. Furthermore, the research has shown that the usage of downstream experts of content knowledge may enact curricular decisions based on the structure of the content domain that conflict with the developmental and learning needs of students (Nathan, Koedinger, & Alibali, 2001; Nathan & Petrosino, 2003).

Current state analysis, as mentioned previously, used retroactive, secondary sources for analysis, thus introducing a gap between the data analyzed and the actual state of the system. Student feedback was provided through electronic student evaluations of teaching, which may be biased unto themselves (Shevlin, Banyard, Davies, & Griffiths, 2000). The coding process was performed by one researcher with spot checks by another, which may thus lead to the injection of bias in the analysis. The root cause analysis was primarily performed by one researcher and could thus be subject to bias in the outputs as well.

Guidelines for Future Implementations and Future Work

Despite these limitations, this study does discover potentially useful initial results. It shows that there is already potential for the implementation of lean in stabilizing course operations and introducing standardization into course operations, falling in line with the Zero Defects principle. We find that before lean may be more thoroughly implemented in a course's structure, especially in material flows (i.e. pedagogical techniques) and value specification (eliciting stakeholder input for determining curricular content), work must first be taken towards the stabilization and standardization of course operations, including a measurement and analysis of the current state of a course. Such a conclusion lends itself to the House of Lean model proposed in Dennis (2016). Before the pillars of Jidoka and Just-in-Time may be built, a foundation of stability and standardization must first be established.

This study provides the following guidelines for future implementation:

1. Initial implementations of Lean in the classroom require a documentation of the current state of the system and a standardization of current processes to provide the groundwork for improvement to begin.
2. Assessing the alignment of course materials with course outcomes and an evaluation of the course outcomes and experience against the needs of the learners in the class is key to implementation as well.
3. Once the foundation has been laid, improvements may be implemented following the PDCA cycle, examining quality and assessment techniques in the course and the flow of information and material.
4. Throughout this, TAs and new instructors will need to be involved in trainings not only in the standard processes of the course but in continuous improvement methodologies as well, such that they may be prepared to recognize and act upon issues as they occur through the course.

This work only scratches the surface of potentials for lean in the classroom. Future work includes a systematic study of existing pedagogies and evidence-based instructional practices and potentially aligning them under Lean principles. Lean principles and tools may be developed into curricular tools in their own right, such as the usage of value specification techniques for developing curricular content or applications of value stream mapping to the course level, overall curriculum level (i.e. four-year undergraduate programs), and even individual lesson level. Root cause analysis may provide a method for analyzing student errors or misunderstandings to provide educational remedies, and the cycle of PDCA may provide a systematic method for monitoring and improving courses. These tools may then be studied and validated in terms of effectiveness for course improvement and student learning. Once a “Lean Pedagogy” is developed, additional work may also be undertaken to study the process of implementing continuous improvement within curriculum development and the classroom and methods for improving success and mitigating challenges.

This future work ultimately aims to provide a model or roadmap for implementations of lean in classrooms across higher education, complete with established parallels between lean principles and tools and educational theory and evidence-based instructional practices. Establishing solid connections between a powerful continuous improvement philosophy and the work of teachers may set the stage for an impactful pedagogy of continuous improvement.

Appendix

Appendix A: Instructor Survey Results

The instructor survey was administered during June of 2017 to determine which topics related to ENGR 112 are used within downstream courses depending on ENGR 112. The relevant sections of the survey—basic data types, programming structures, and skills—are presented here in the following tables. Instructors are presented with a checklist of options and select those which are relevant to their courses. The frequency is then summarized within the count column of each table.

Basic Data Types	
Category	**Count**
Trees	0
Hash tables	0
Priority Queues	0
Other	1
Lists	3
Cells/Structures	3
Volumetric (quads/tets)	3
Grid Structures	3
Strings	4
Meshes/surfaces	4
Booleans	7
Graphs	8
Arrays	11
Matrices	15

Appendix Table A.1: Instructor survey results for "Basic Data Types"

Programming Structures	
Category	**Count**
Lambda/anonymous functions	0
Procedural/Scripts	4
For/While Loops	15
Functions	15
If Statements/Logic	16

Appendix Table A.2: Instructor survey results for "Programming Structures"

Skills	
Category	**Count**
Writing their own optimization programs	0
Data reduction - e.g. PCA, machine learning	0
Read unstructured data	1
Write unstructured data	1
Optimization (using)	3
String manipulation	4
Symbolic equation solving	4
Writing their own code to perform simulations	4
Filtering/signal processing	4
User interface - menus and buttons	4
Simulation (set up and run)	5
Iterative functions (writing their own)	5
Polynomial equations	5
Plot/display 3D data	7
Data fitting (2D and higher)	7
Data interpolation/extrapolation	7
Calculus functions	8
Write structured data	9
Fit a curve/function to data	9
Matrix transformations	9
Plot/display 1D and 2D data	10
Solving linear systems of equations	10
Read structured data	12

Appendix Table A.3: Instructor survey results for "Skills"

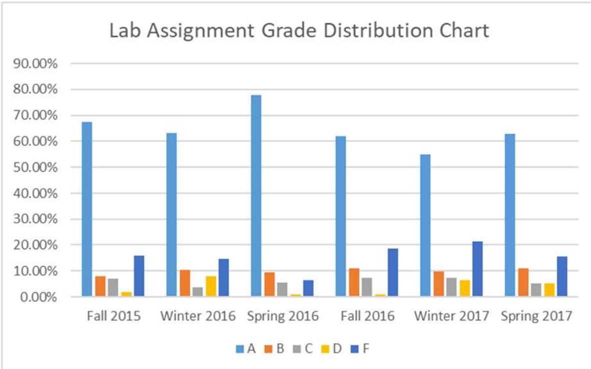
Appendix B: List of ENGR 112 Weekly Learning Outcomes

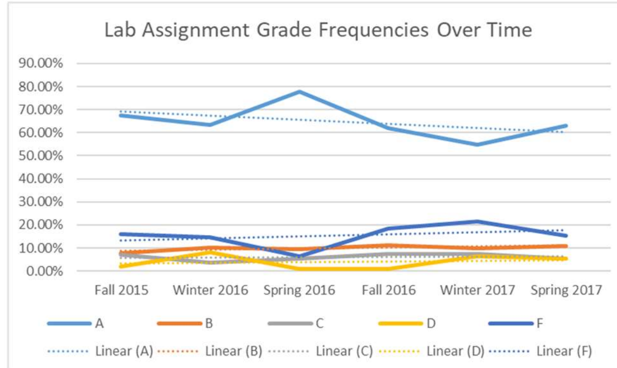
A table of the weekly learning objectives for ENGR 112 is shown below. It should be noted that there are sub-learning objectives for each shown here that are not described or used for this analysis. It is assumed that this high-level list provides an adequate level of resolution for the content validation. It is possible to validate every individual sub-learning objective, but that would go beyond the demonstrative usefulness of this analysis.

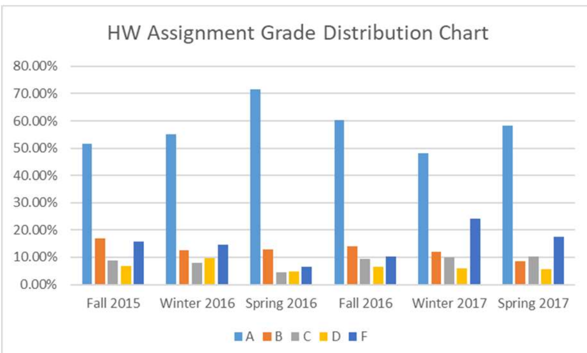
	Learning Objective
Week 1	Find and describe [components of the MATLAB environment]
	Create, and run, a script that calculates a given equation
	Get input from the user and display the results of calculations in the command window
Week 2	Create an array and access elements of it
	Perform equations on the arrays, plot the result
	Use the debugger to examine the value of a variable while executing
Week 3	Create if statements to control program flow
	Use a loop index variable to access elements of the array
	Use loops to repeat commands
	Use loops with if statements
	Relational equations: Compute with Booleans and comparisons (<, >, ==, ~=, &, | ~)
	Use the debugger to fix if statements and loops
Week 4	Create a function file to encapsulate functionality
	Create an anonymous function
	Use a function file or an anonymous function
	Local versus global variables
Week 5	Use fzero and fminbnd to solve problems of the form "For a function y = f(x), give me the parameter x that results in a specific value for y, or that minimizes y"
	Plot a function directly using fplot
	Use a for or while loop to solve an equation defined using an iterative definition: xi+1 = xi + (something)
	Use quad and trapz to integrate an equation
	Use an anonymous function to create a function of a single variable from an existing function by "fixing" the values of some of the parameters

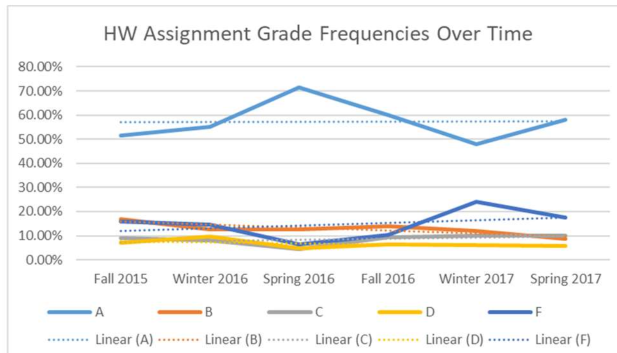
	Learning Objective
Week 6	Perform operations on polynomials
	Fit a polynomial to data using polyfit
	Create samples at different points from existing data
	Explain the difference between function fitting and interpolation
Week 7	Perform operations on matrices
	Create scale, rotation, and translation matrices and use them to position objects
	Calculate a dot product between two vectors
Week 8	Use matrices to set up, and solve, linear systems of equations
	Create functions that return multiple variables
	Define a parametric function
	Define a function in two variables [use meshgrid and surface plotting]
Week 9	Creating and manipulating strings
	Creating and manipulating arrays of strings
	Manipulating strings as arrays of characters
	Reading and writing to files
Week 10	Use the sphere and cylinder commands to make 3D shapes
	More practice with meshgrid

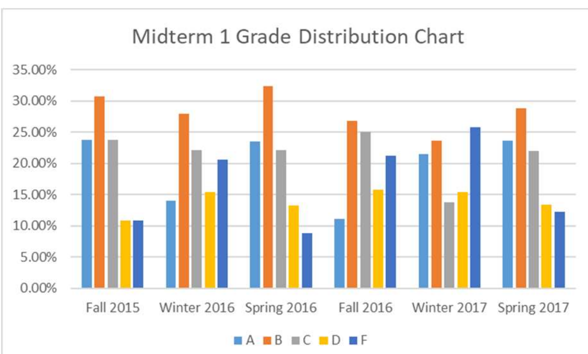
Appendix C: Course and Assignment Grade Distributions

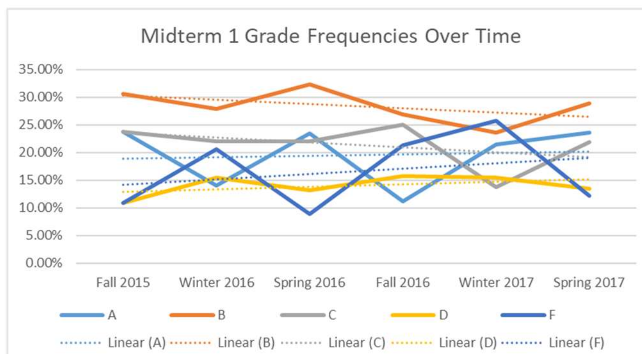


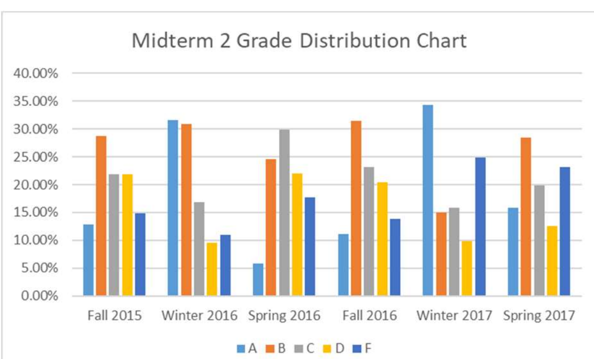


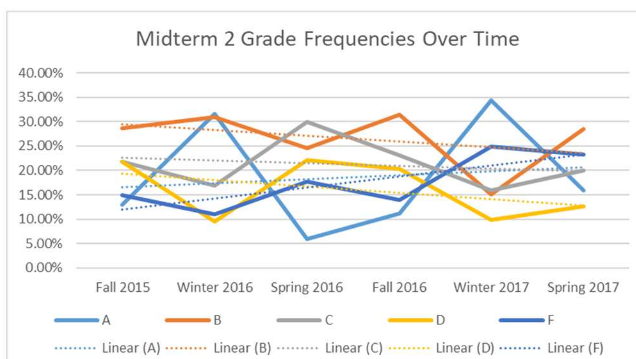


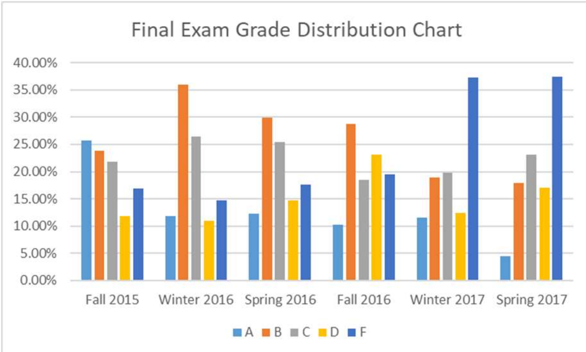


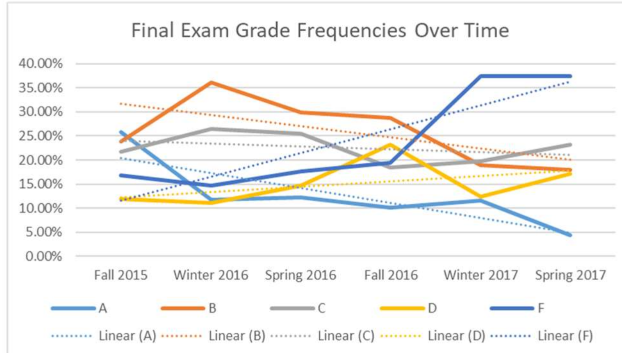


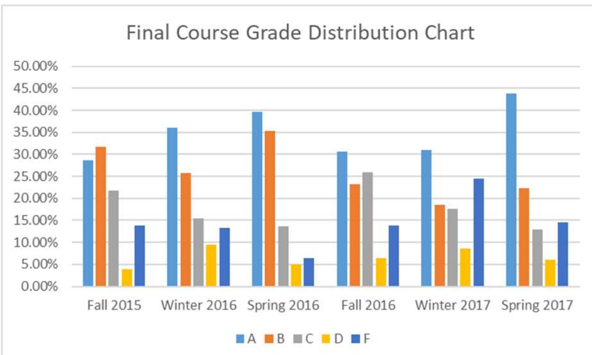


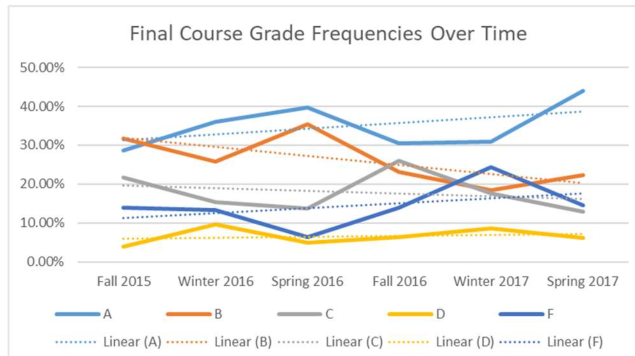





Final Exam Grade Distribution Chart
40.00%
35.00%
30.00%
25.00%
20.00%
15.00%
10.00%
5.00%
0.00%
Fall 2015
Winter 2016
Spring 2016
Fall 2016
Winter 2017
Spring 2017
A B C D F


Final Exam Grade Frequencies Over Time
40.00%
35.00%
30.00%
25.00%
20.00%
15.00%
10.00%
5.00%
0.00%
Fall 2015
Winter 2016
Spring 2016
Fall 2016
Winter 2017
Spring 2017
A B C D F
Linear (A) Linear (B) Linear (C) Linear (D) Linear (F)


Final Course Grade Distribution Chart
50.00%
45.00%
40.00%
35.00%
30.00%
25.00%
20.00%
15.00%
10.00%
5.00%
0.00%
Fall 2015
Winter 2016
Spring 2016
Fall 2016
Winter 2017
Spring 2017
A B C D F


Final Course Grade Frequencies Over Time
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
Fall 2015
Winter 2016
Spring 2016
Fall 2016
Winter 2017
Spring 2017
A B C D F
Linear (A) Linear (B) Linear (C) Linear (D) Linear (F)

Appendix D: eSET Quantitative Data Results

As a whole										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	0	1	8	17	12	12	1	4.4	1.1	52%
Winter 2016	4	1	22	31	22	13	1	4.1	1.2	68%
Spring 2016	5	7	13	25	11	12	38	4.0	1.4	54%
Fall 2016	0	1	5	13	17	18	1	5.0	1.0	56%
Winter 2017	7	12	17	28	12	11	1	3.8	1.4	58%
Spring 2017	2	2	11	39	11	10	44	4.1	1.1	54%
							Average	4.23	1.20	

Instructor's contribution										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	0	2	9	13	13	13	1	4.6	1.2	52%
Winter 2016	4	3	17	37	18	15	1	4.1	1.2	68%
Spring 2016	4	10	11	23	11	12	40	4.0	1.4	54%
Fall 2016	0	0	6	12	17	19	1	5.0	1.0	56%
Winter 2017	8	11	17	24	18	9	1	3.8	1.4	58%
Spring 2017	2	1	10	36	10	8	52	4.1	1.0	54%
							Average	4.27	1.22	

Course objectives										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	0	1	7	16	13	13	1	4.6	1.1	52%
Winter 2016	4	0	15	33	26	16	1	4.3	1.2	68%
Spring 2016	6	5	12	25	14	11	38	4.0	1.4	54%
Fall 2016	0	1	7	15	15	16	1	4.8	1.1	56%
Winter 2017	8	11	15	25	17	11	0	3.9	1.5	57%
Spring 2017	2	1	13	37	9	11	46	4.1	1.1	54%
							Average	4.28	1.22	

Clarity of student responsibilities										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	0	1	5	17	13	14	1	4.7	1.0	52%
Winter 2016	2	1	14	33	31	13	1	4.4	1.1	68%
Spring 2016	5	6	11	24	13	14	38	4.1	1.4	54%
Fall 2016	0	0	8	15	13	18	1	4.8	1.1	56%
Winter 2017	4	8	22	24	16	13	0	3.9	1.3	57%
Spring 2017	1	1	15	36	9	12	45	4.1	1.1	54%

Course organization										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	0	1	7	16	11	15	1	4.6	1.1	52%
Winter 2016	7	3	15	35	23	11	1	4.1	1.3	68%
Spring 2016	5	8	12	24	12	12	38	4.0	1.4	54%
Fall 2016	0	2	4	17	17	14	1	4.7	1.0	56%
Winter 2017	8	10	24	24	11	9	0	3.5	1.4	57%
Spring 2017	2	4	12	37	10	8	46	4.0	1.1	54%
							Average	4.15	1.23	

Availability										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	1	2	5	13	13	16	1	4.8	1.2	52%
Winter 2016	3	1	17	28	21	24	1	4.4	1.3	68%
Spring 2016	4	5	10	24	14	15	39	4.2	1.4	54%
Fall 2016	0	0	6	13	14	21	1	5.1	1.0	56%
Winter 2017	5	7	18	18	16	22	1	4.2	1.5	57%
Spring 2017	1	1	11	33	12	15	46	4.2	1.1	54%
							Average	4.48	1.25	

Accommodate differences										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	2	3	8	14	9	12	3	4.3	1.4	52%
Winter 2016	4	6	22	34	12	15	2	3.9	1.3	68%
Spring 2016	4	9	12	25	8	12	41	3.9	1.4	54%
Fall 2016	0	2	4	23	12	13	1	4.4	1.0	56%
Winter 2017	11	9	21	25	11	9	1	3.6	1.5	57%
Spring 2017	3	0	10	36	9	10	50	4.1	1.1	54%
							Average	4.03	4.03	

Instructor's interest										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	1	3	7	17	9	13	1	4.3	1.3	52%
Winter 2016	4	4	12	29	27	17	2	4.4	1.3	68%
Spring 2016	3	10	11	22	12	12	41	4.0	1.4	54%
Fall 2016	0	0	6	14	15	19	1	5.0	1.0	56%
Winter 2017	11	9	20	22	17	7	1	3.6	1.5	57%
Spring 2017	3	0	10	33	13	9	51	4.1	1.1	54%
							Average	4.23	1.25	

Stimulate thinking										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	1	6	4	18	9	12	1	4.3	1.3	52%
Winter 2016	5	3	20	33	20	13	1	4.1	1.3	68%
Spring 2016	5	7	14	19	11	14	41	4.0	1.5	54%
Fall 2016	0	1	6	15	13	19	1	4.9	1.1	56%
Winter 2017	8	10	19	25	13	11	1	3.7	1.4	57%
Spring 2017	3	0	11	36	8	11	50	4.1	1.1	54%
							Average	4.18	1.29	

Feedback										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	2	5	13	14	7	8	2	3.8	1.4	52%
Winter 2016	5	7	15	38	19	9	2	4.0	1.2	68%
Spring 2016	5	10	9	22	14	11	40	4.0	1.5	54%
Fall 2016	1	0	4	21	11	17	1	4.6	1.1	56%
Winter 2017	13	14	25	20	7	7	1	3.1	1.4	57%
Spring 2017	2	3	11	37	8	8	50	4.0	1.1	54%
							Average	3.92	1.28	

Classroom environment										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	0	0	8	19	10	13	1	4.4	1.0	52%
Winter 2016	2	3	10	37	25	17	1	4.4	1.1	68%
Spring 2016	3	7	12	24	13	12	40	4.1	1.3	54%
Fall 2016	0	1	3	15	13	22	1	5.1	1.0	56%
Winter 2017	7	9	19	27	15	9	1	3.8	1.4	57%
Spring 2017	2	1	12	32	11	10	51	4.1	1.1	54%
							Average	4.32	1.17	

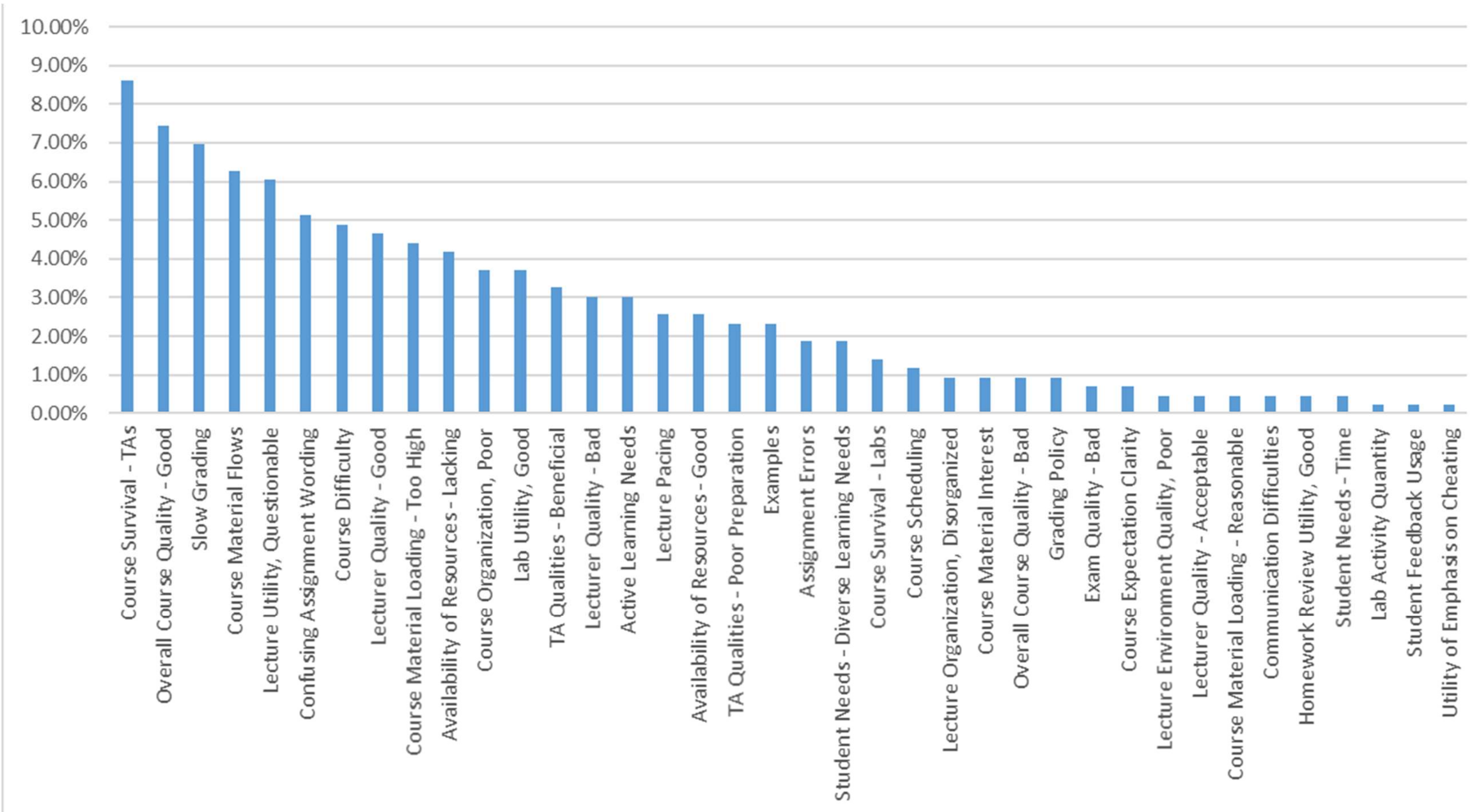
Evaluation of student performance										
Term	**Very Poor (1.0)**	**Poor (2.0)**	**Fair (3.0)**	**Good (4.0)**	**Very Good (5.0)**	**Excellent (6.0)**	**Unable to Rate**	**Median**	**Std Dev**	**Response Rate**
Fall 2015	0	1	6	19	12	11	2	4.4	1.0	52%
Winter 2016	3	4	13	33	23	16	2	4.3	1.2	68%
Spring 2016	5	7	10	24	10	15	40	4.1	1.5	54%
Fall 2016	0	1	7	14	17	13	1	4.7	1.0	54%
Winter 2017	6	9	18	31	11	11	0	3.8	1.4	57%
Spring 2017	2	1	11	34	12	8	51	4.1	1.1	54%
							Average	4.23	1.20	

Appendix E: Category and Code Descriptions

Categorization	Description	Code	Description
Opinions on Overall Course	This category encompasses codes which pertain to student opinions on the overall course. While useful in developing a high-level view of student views on the course quality, general opinions are not as useful in locating the specific issues with the course.	Course Difficulty	Any reference to the course's difficulty, especially with respect to components being challenging. Most responses note the course as being challenging or difficult.
		Course Material Interest	Describes students' interest in course content in positive or negative affect
		Overall Course Quality - Good, Bad	Concerns general judgements on the quality and experience of the course. Course quality may be judged as Bad or Good.
Lecture Characteristics	This category involves codes which pertain to different components of lecture such as the pacing, organization, and environment, as well as broader opinions such as the student view of the quality of lecture and its utility.	Lecturer Quality - Good, Acceptable, Bad	Concerns judgements on lecture quality. Judgments range from Good to Acceptable to Bad.
		Lecture Pacing	Concerns how students view the pacing and speed of lectures. Some students note inconsistencies in the speed while other note that lecture moves too quickly.
		Lecture Organization, Disorganized	Relates to comments on the general organization of lecture. Students primarily view the organization poorly.
		Lecture Utility, Questionable	Concerns judgements on the overall utility or usefulness of lecture. Students note that lectures are overall not very useful.
		Lecture Environment Quality, Poor	Relates to student views of the physical classroom space for lecture. Students primarily note that lecture is too small and difficult to see given the classroom layout.
Lab Characteristics	This category concerns codes which relate to characteristics of lab. These codes a primarily opinions related to lab, noting that lab is overall useful or necessary to course survival.	Lab Utility, Good	Relates to student views on lab utility. Most view lab with positive affect, noting that lab sessions are helpful.
		Lab Activity Quantity	Relates to the need for lab activities. Most comments express the need for increasing the number of lab activities.
		Course Survival - Labs	Relates to the idea that lab is necessary to survive through the course.

Course Flow	This category pertains to the overall flow of course material, including scheduling, material (content) progression, and overall organization. Within a Lean Thinking framework, this category relates to the Flow principle.	Course Material Flows	Relates to the general flow of course content and activities and any issues in the process, including preparation and movement from one activity to another.
		Course Scheduling	Relates to needs expressed by the students to change the scheduling of the course or scheduling changes that occur during the course.
		Course Organization, Poor	Concerns general student opinions on the overall organization of the course.
		Course Material Loading - Reasonable, Too High	Concerns the quantity of work students must perform and student opinions on the subject. Some comment the loading as being "Reasonable", while many others say it is too high.
Course Policies	This category relates to course policies such as expectations, grading, and cheating, as well as the clarity of their emphasis.	Course Expectation Clarity	Relates to the need to clarify course expectations and how well they are set.
		Utility of Emphasis on Cheating	Some students note the usefulness of the emphasis on cheating at the beginning of the course. This is primarily expressed with negative affect.
		Grading Policy	Relates to student comments on the grading policy, specifically its fairness.
TA Characteristics	This category concerns the characteristics of teaching assistants (TAs) in the course. These codes show that TAs are primarily helpful and possibly necessary to get through the course, although they may sometimes be poorly prepared.	Course Survival - Tas	Relates to students expressing that TAs are vital to getting through the course and their help is essential
		TA Qualities - Poor Preparation	Some students note Tas as being unprepared or unprofessional in behavior.
		TA Qualities - Beneficial	Some students note that Tas are beneficial in helping students and are overall "great".
Instructional Components	This category relates to specific instructional components of the course such as the usage of examples, the attendance to active learning needs (primarily in lecture), and the	Examples	Relates to student comments on the need or usefulness of examples used throughout lecture and other components of the course.
		Active Learning Needs	Anything that describes the need to promote interactivity or collaboration during class time.

	usefulness of homework review sessions.	Homework Review Utility, Good	Any comments that describe the utility or usefulness of homework review. Comments primarily express positive effect.
Assignment Material Issues	This category involves specific issues with the physical assignment materials, such as errors and wording as well as exam quality.	Exam Quality, Bad	Refers to student comments on the Exam Quality. Comments have been primarily of negative affect.
		Assignment Errors	Describes assignments as having errors in wording or self-check
		Confusing Assignment Wording	Wording of assignments causes confusion in students
Feedback & Availability	This category pertains to the ability of the course to provide feedback to students. Feedback and availability are grouped together as they both relate to the responsiveness of the course to relaying information back to students.	Slow Grading	Many students note that grading is slow or that there is a need to improve grading speed.
		Availability of Resources - Good, Lacking	Describes the availability of resources and help in the course, including digital course materials or people (TA's or Instructors). "-Good" that resources are available and seem abundant. "- Lacking" indicates that resources are not readily accessible.
		Communicatio n Difficulties	Refers to any difficulties in the communication process
Voice of the Student	This category concerns capturing the voice of the student; that is, how the course attends to and respects student needs and opinions. Within a Lean Thinking framework, this category relates to the "Respect for Humanity" principle.	Student Feedback Usage	Concerns the usage of student feedback during the course to improve the course.
		Student Needs - Diverse learning needs	Some students express that their classmates have diverse learning needs. This is mainly instantiated as differing quantities of background knowledge, with some students having significant coding experience while other have none.
		Student Needs - Time	Concerns student needs relating to their available time to complete course activities.



Appendix Figure E.1: Percentage of preliminary coded responses of all coded responses.

Appendix F: Course Summary Table

This appendix contains a fold-out of the entire Course Summary Table.

	Week 1	Notes	Week 2	Notes
Syllabus	Introduction & historical overview Introduction to numeric, character, and Boolean data types Arithmetic precedence rules & MATLAB intrinsic functions MATLAB script files Input/output operations (DISP, FPRINTF, INPUT)	LO 1, 2	Vectors and vector operations 2-D plotting Statistics of an array: (MIN, MAX, MEAN & STD)	LO 2, 3
Weekly Learning Objectives	Find and describe the following in the MATLAB environment: The command window, where you can type commands directly into MATLAB The script window, where you can edit and save scripts The variable window(s) for displaying values of variables The history window and how to save and repeat commands Create, and run, a script that calculates a given equation Document the script using comments Understand (and use) the following commands: clear, clc, sin, cos, log, exp, ^, pow, sqrt Use sin and sind appropriately Get input from the user and display the results of calculations in the command window Use the following commands: disp, fprintf Print out the value of a variable using %0.0f Explain what the ; does at the end of a line		Create an array and access elements of it Difference between using linspace and the :: operator to create arrays Create arrays by hand Access elements of an array using the : : operator Use end to get the last element Count backward Skip every other element Show which index gives which value in the array Perform equations on the arrays, plot the result Use the .*, .^, and ./ operator to create equations that operate on arrays Plot multiple data sets in a single window (plot, hold on) Create sub windows (subplot) Set the color, marker, and line type when plotting ('X-r') Annotate plots with a legend, title, and axes (title, xlabel, ylabel, legend) Use the debugger to examine the value of a variable while executin Know how to set (and clear) break points Know how to examine the values of variables while executing a script	Missing some of the work with the debugger. Although it is implied through the problems solving process, it does not appear to be explicitly reiterated in the material.
Textbook Reading (zyBooks)	1.1 Solving engineering problems with MATLAB 1.2 MATLAB background 1.3 MATLAB and the interpreter 1.4 Computer basics 1.5 A brief tour of a computer 1.6 Basic input: The input() function 1.7 Basic output II: fprintf() 1.8 Basic output 1: disp()	Primarily corresponds to lecture 3	2.1 Row arrays 2.2 Column arrays 2.3 Constructingrowarrays 2.4 Functions to create numericrow arrays 2.5 Multi-element row array indexing using logical arrays 2.6 Indexing rows and columns usinga single colon 2.7 Row array resizing 2.8 Concatenation 2.9 Dimensional properties of arrays 2.10 Reshaping arrays 2.11 1D element-wise arithmetic operators 2.12 2D Arrays: Arithmetic Operators 2.13 Functions and 1D arrays 2.14 Simple plotting 2.15 2D data plots I 2.16 2D data plots 2.17 Plots	Gets into 2 dimensional operators and matrix operations already, which aren't covered until week 7
Lecture (Mon) - Detailed Schedule	1-1 Intro 1-2 Variables Equations		2-1 Arrays min, max, mean, sum, length, :, linspace	
Lecture (Mon) - Actual	1-1 Intro-Sami Introduction to course, what is learned, format and flow, expectations and keys to success, assignment format, To-Do before Lab, Classes where MATLAB is used, About your instructor. 1-2 Variables Equations Variables What they are Syntax rules for names Declaring and assigning a value to a variable Arithmetic Operators Precedence (when to use parentheses) 3 examples.	Can't tell what was actually used to teach. Lecture states that weeks 1-2 are difficult, weeks 3-5 are very hard, and then it gets more managable from there. Already this hints at unevenness, or mura.	2-1 Arrays Arrays (Vectors) Creation, Modification, Basic Operations Creation: [], :;, linspace, zeros, ones Accessing and modifying Addition,subtraction, multiplication, dividision, exponents, element wise operations Vector functions: length, max, min, sum, mean, std 2 exercises: read code, predict output 1 example	Slides have walls of code. It is not apparent how much is actually done through coding in the MATLAB interface without further investigation. Example is dense and talks about tensile strength, covered in ENGR 213.
Lecture (Fri) - Detailed Schedule	1-3 Functions IO Input, sin/cos, sqrt, exp, nthroot 1-4 Pseudo Code		2-2 Plotting subplots, plot, polar, loglog, line styles	Not reflected in syllabus
Lecture (Fri) - Actual	1-3 Functions IO Matlab built-in equations and variables: sqrt, nthroot, exp, abs, log, log10, factorial, round, floor, ceil, mod, pi, inf, NaN, all trig functions. Input from user Printing output to the command window Omit ;, disp(), fprintf Script files 3 examples 1-4 Psuedo code Comments and pseudo code Steps to go from a word problem to matlab code Explicit Problem Solving Strategy explanation 2 examples		2-2 Plotting 2D Plotting Creation, Formatting, Examples Plot, annotations (xlabel, ylabel, legend, title, grid, hold on, axis equal, figure, clf) Exercise: Plot by hand Plot Types: polar, loglog, semilogx, semilogy, bar, pie Locators Subplot Exercise: Plot projectile motion	Example models a damped spring-mass system. Script text on slide is very small.
Lab - Detailed Schedule	MATLAB windows, create a script, create variables, plot, write equation		Create arrays (linspace, :, []), access arrays (:, ()), write an equation with an array, plot, swap variables	
Lab - Actual	LO: What is a script file? What is the difference between a script file and the command window? How do you create variables? How can you examine the values of variables? How do you use the debugger? 1. Basic variable manipulation in command window and script. 2. Use linspace to create an array of variables and use various display commands for arrays 3. Plot a circle using cos and sin 4. Solve 2 equations 5. Demonstrate usage of the debugger	Linspace and plotting is in week 2. Students have not learned anything about arrays or plotting yet. The due date of 4/10 confirms this as there would be no time to receive additional lecture.	LOs How do you make, edit, and use an array? How do you write an equation using variables that are arrays? How do you plot those variables? More practice on turning word problems into MATLAB code 1. Array creation and accessing different elements; printing sum, min, average 2. Editing an array, accessing multiple elements in ana array 3. Plotting values from Lab 1 4. Variable swap	Digging back through to Lab 1 for the equation is a waste of time Nothing on debugging
HW - Detailed Schedule	HW1: Write equations with scalar values		HW2: Write equations with arrays, plot the results, different types of plots	
HW - Actual	Goals: What is a script file? What is the difference between a script file and the command window? How do you create variables? How can you examine the values of variables? How do you create equations, pass them variables, and print out the results? How do you display the values of variables? 1. Solve two equations. 2. Solve a projectile motion problem: find landing location (x, y) and time of hitting the ground 3. Find the volume of a hollow sphere. 4. Teacup 1 - Translate and rotate points about the origin 5. Epidemic 1 - Model two steps of an epidemic 6. Euler leaf 1 - Model taking time steps of a leaf in the wind	Some redundant learning objectives from lab 1. Assume that most learning objectives are integrated into the problem solving system. Problem 2 uses projectile motion concepts which may not have been learned yet. Teacup problem is hard to visualize and the term canonical position can confuse people (as it did Winter 2017). Epidemic problem uses differential notation which students may not have learned yet Euler problem uses physics concepts which may not have been learned well yet Teacup and Euler problems are over a page long, and term long problems have a long introduction.	Goals Understand how to create arrays, calculate with them, and extract data from them. Plot the results of calculations. Be able to split up plots into multiple windows, plot multiple things in the same window with different colors and attributes. 1. Plot a function, find the max and min values and the t value where they occur 2. Plot changes in sound level 3. Print a polar rose in various subplots 4. Plot stresses based on changes in different variables 5. Teacup 2 - Sketch a teacup and apply all the transformations at once to an array of points, then plotting the initial and final result 6. Epidemic 2 - Calculate two time steps with an array of starting values	Problem 1: Does not appear that they have been explicitly taught how to retreive both the value and index of a max or min Problem 2: Students may not have learned about the concept of sound pressure and loudness, although this should not impact their ability to code. Problem 3: Students may not understand how polar plotting works Problem 4: Students may not understand equation for stress, although it should not affect how they code the problem Problem 5: Students may not understand how ginput works

	Week 3	Notes	Week 4	Notes
Syllabus	Program development: MATLAB relational operators/selection structures: IF statements MATLAB repetitive structures: FOR and WHILE loops	LO 1, 2	Program development: User-defined MATLAB functions & function files Anonymous functions	LO 2, 3
Weekly Learning Objectives	Create if statements to control program flow Create a conditional statement with a variable Use elseif to create multiple conditions Use else to catch remaining conditions Use a loop index variable to access elements of the array Use loops to repeat commands Use a for loop to plot multiple versions of a function on a graph Use a for loop to perform iterative calculations x(k+1) is a function of x(k), xk+1 = (something) of xk Use a while loop to perform iterative calculations until some criterion is met Use a while loop to control a simulation Use a for loop to perform the same calculation multiple times but with different input values Use loops with if statements Switch between two different equations based on the values of the input Perform iterative calculations where the calculation changes at each iteration based on the current values Relational equations: Compute with Booleans and comparisons (<, >, ==, ~=, &, | ~) Express a conditional problem statement (e.g., stop when x is less than a small number) as a relational equation Combine multiple such conditional statements into a single relational equation using &, | and ~ Use the debugger to fix if statements and loops		Create a function file to encapsulate functionality Evaluate an equation with one (or more) input variables Save the function file with the correct name (same as function name) Set the output variable(s) correctly Create an anonymous function Evaluate an equation with one (or more) input variables Use an anonymous function to "fix" the values of the inputs on a function file Know which values are set when the function is called and which are set when the function is created (ie, are not input variables) Know what happens to an anonymous function when clear is called (it disappears) Use a function file or an anonymous function What order to pass in the input values so they get assigned to the correct input variables in the function Know where MATLAB looks for function files Local versus global variables How to create a global variable How to use one in a function or different script Review material from weeks 1-3 for mid-term	
Textbook Reading (zyBooks)	3.1 Relational operators and row arrays 3.2 Logical operators and 1D arrays 3.3. Combining relational and logical operators 3.4 Find function 3.5 Relational operators 3.6 Logical operators 3.7 If-else statement 3.8 Multiple branches 3.9 Switch statement 3.10 While loops 3.11 More while examples 3.12 Counting 3.13 For loops 3.14 Loops and arrays 3.15 Nested loops	Switch statements seem to be optional in lecture	4.1 Custom functions 4.2 Counting function arguments 4.3 Scope of variables 4.4 Global variables 4.5 Function handles 4.6 Local functions 4.7 Nested functions	Nested functions do not appear to be covered in lecture.
Lecture (Mon) - Detailed Schedule	3-1 Ifs and Relational ops 3-2 Loops if/elseif/else, for, while		4-1 Functions function files, anonymous functions 4-2 Global variables	
Lecture (Mon) - Actual	3-1 If Statements Relational Operators (< > == ~= & | ~) Like +, .*, except true or false Selection statement (if elseif else end) Used to control program flow, i.e. do this or do that Program flow Exercise: Ands and Ors Control structures and if statements Exercise: Add flat part to a graph Exercise: Formal if statement 3-2 Loops Loops for loop (do something n times) while loop (repeat until done) Repetition Structures Some code reading exercises Exercise: For loop - predict output of a for loop Exercise: Plot multiple plots, observe solution in slides While loops	Slides get really dense at examples, code is hard to read, not sure if this isn't just run in the IDE and stepped through with the debugger	4-1 Functions User-Defined Functions What is a Function Creating a Function Using Function scripts Anonymous functions Two examples of functions Conceptually - Function on one side of a wall Weibull example Anonymous functions Concept Check (read code) 4-2 Global Variables Local Variables and Global Variables One example	Global variables aren't really used anywhere?
Lecture (Fri) - Detailed Schedule	3-3 Plotting in loops, nested loops 3-4 Relational statements & | ~ 3-5 Switches		Midterm I Equations, arrays, plots, simple if/while/for loops	
Lecture (Fri) - Actual	3-3 Nested loops Nested For Loops Simple example runthrough 3-4 Relational eqs If statement review More on Relational Operators How to combine them together How not to use for loops Info on order of precedence Exercise 1: Ands and Ors: Predict output Exercise 2: Precedence: Predice output Exercise 3: Operators on arrays Find() Example 3-5 Switches (Optional Slides)		Midterm I 1 - [Checklist] Reasons to comment 2 - [Checklist] When to use dot operator 3 - [Short Answer] Fix code (missing multiplication symbol) 4 - [Short Answer] What is the output of this statement (calling array components) 5 - [MC] Plotting with a line (syntax) 6 - [MC] Valid if statement conditions 7 - [Short Answer] Output of if statement 8 - [Short Answer] Fix code (plot xs instead of ts) 9 - [Short Answer] Output of code (array functions) 10 - [Short Answer] Reading a for loop 11 - [Write Script] Evaluate an equation with array input 12 - [Short Answer] Logical comparisons 13 - [Free Response] Write pseudocode to perform a problem (calculate budget) (loops, conditionals) 14 - [Write Script] for 13)	Writing scripts on paper tends to be tricky. Students are given very little experience in having to read and interpret code in assignments, yet the exam requires them to read code. Lab quizzes do require a little code reading, but inconsistently.
Lab - Detailed Schedule	Write for loops, iterate through elements in for loop to calc sum, if statements, if statement in for/while loop		Write a function file and an anonymous function. Use them. fplot versus plot. Local versus global variables.	
Lab - Actual	LOs: Write several simple for/while loops. Write several simple if statements to control program flow. Edit a variable inside of a for loop. Using the debugger to step through the for loops and if statements. 1. Count bottles up and down using a for loop 2. Write a for loop to calculate sum and average of any sized array 3. A while loop to check if a user enters a positive, negative, or zero number. 4. EC: Use a while loop to sum vector elements until the result is larger than 15	There is no problem to practice using if statements by themselves. Debugging is recommended within steps, but not explicitly required as an output like Lab 1.	LOs: Create (and call) a simple function file Create (and call) a simple anonymous function Use an anonymous function to create a function with one parameter from a function with multiple parameters 1. Create a function file and pass it variables and inputs. Use a breakpoint. Use fplot with the function. 2. Create an anonymous function and try various options. Use fplot 3. Create a function that returns two values, and plot concentric circles. EC requires for loops	There is no fplot coverage in lecture.
HW - Detailed Schedule	HW3: For loops, while loops, if statements		HW4: Create functions. Practice with iterative equations.	
HW - Actual	Goals Understand how to use for loops to avoid replicating code and for iterating over arrays. Know how to use relational operators to ask questions like greater than/less than/the same about data. Know how to use if statements to control program flow. 1. Increase the volume of spheres using a while loop until it reaches a certain volume. (while loop, conditionals) EC: Use if statements to print the middle volume 2. Create a Dragon Curve Fractal using a for loop and plot 3. Epidemic 3 - Perform 140 iterations of the epidemic using a for loop, and plot the results. (for loop) 4. Euler 2 - Move a particle multiple steps. Plot x, v, and F of the particle. Use a while loop and conditionals to find when the particle reaches it's peak. EC: Use arrays	Problem 2: Students may be confused by the fractal notation. Students are required to look up the randi function. Euler: Has been a week since the previous problem. There are a lot of implementation notes.	Goals: More practice with for loops and if statements How to create a function file How to create an anonymous function How to use functions as inputs to other functions 1. Redo a HW problem (or do an original problem) with an anonymous function 2. Redo a HW problem (or do an original problem) with a function file 3. Approximate Euler's number using a series (conditionals, while loops, count iterations, EC. requires counting operations) 4. Epidemic problem 4 [Using functions] Create functions DiseaseStep and DiseaseSimulate 5. Euler leaf problem 3 [Using functions] Create functions EulerIntegrate and EulerSimulate 6. EC: Teacup problem 3 [multiple teacups] Nested for loops to plot multiple teacups 7. EC: Swoosh problem, use fzero and for loop to find roots of function, use fplot, plot zeros	There are 7 problems here, although 2 are extra credit. 3. Requires conceptual knowledge of series. If students have not successfully completed continuous assignments, they will be having a lot of trouble with these problems.

	Week 5	Notes	Week 6
Syllabus	Finding the root and min of a function (FZERO, FMINBND) Numerical Integration & Differentiation (INTEGRAL, TRAPZ)	LO 2, 3, 4 Potential Issues: Some students have not taken calculus yet	Curve-fitting: Interpolation, optimization and cubic splines (INTERP1 , POLYFIT & SPLINE) MATLAB matrix operations
Weekly Learning Objectives	Use fzero and fminbnd to solve problems of the form "For a function y = f(x), give me the parameter x that results in a specific value for y, or that minimizes y" Plot a function directly using fplot Use a for or while loop to solve an equation defined using an iterative definition: xi+1 = xi + (something) Use quad and trapz to integrate an equation Solving for distance traveled Solving for area Use an anonymous function to create a function of a single variable from an existing function by "fixing" the values of some of the parameters		Perform operations on polynomials Create a polynomial as an array Use polyval to evaluate a polynomial Calculate the roots of a polynomial Calculate the derivative of a polynomial Perform arithmetic operations on polynomials Add (+), subtract (-), multiply (conv) and divide (deconv) Fit a polynomial to data using polyfit Adjust the degree of the polynomial to find the best fit to the data Apply transformations to the data in order to fit exponential, logarithmic, and inverse equations using polyfit with one degree Create samples at different points from existing data Use polyfit to fit a polynomial and then evaluate the polynomial Use interp1 to interpolate the samples Explain the difference between function fitting and interpolation How to create a global variable How to use one in a function or different script
Textbook Reading (zyBooks)	5.1 Numerical integration and sum function		6.1 Polynomial interpolation 6.2 Linear regression curve fitting: Polynomial least squares 6.3 Calculus
Lecture (Mon) - Detailed Schedule	5-1 FZero fzero, anonymous functions Lec 5-2 Newton's method		Lec 6-1 fitting and polynomials Lec 6-2 optimization
Lecture (Mon) - Actual	5-1 FZero More on anonymous functions Using functions in functions Finding roots of a function Min/max for functions "Fixing" parameters for anonymous functions fplot fminbnd (for min/max of function) fzero Example: velocity function Example: plotting, fzero 5-2 Newton's Method Newton's Method for Finding Roots (essentially fzero) Writing iterative code (x_i+1 = x_i) Iterative computation example (code) Checking difference between last two elements	Some of this material is relevant for HW 4 Newton's method requires understanding of calculus which may not be understood by students yet.	Lec 6-1 polynomials Polynomial review Converting polynomials to arrays Polyval() Example code for polyval and plotting roots(p) + example code Polynomial Addition and Subtraction with example for each Polynomial Multiplication and Division (convolution, deconvolution) (example) Polyder(p) (example) Polyint(p) (example) Exercise on Polynomial Operations (writing) Lec 6-2 fitting Curve fitting with polynomials (polyfit) Using linear polynomial fit with non-linear functions Using MATLABs fit function Exercise: Sketch a fitted curve for some points polyfit() (code example) Example code problem with polyfit Example for fitting non-polynomials in code fit() function, with example (involves struct)
Lecture (Fri) - Detailed Schedule	Lec 5-3 Numerical Integration integral, trapz		Lec 6-2 fitting (cont.) Lec 6-3 interpolation
Lecture (Fri) - Actual	Lec 5-3 Numerical Inteegration Definite Integration Review Trapezoidal Rule Simpson's Rule Example of trapz() (in code) Matlab Integral() function Example Integral() Example Integral() vs. trapz()	Integration is taught in math 252 and is not even a pre-requisite or co-requisite to the course	Lec 6-3 Interpolation Interpolation interp1() Example in code Linear vs. Spline
Lab - Detailed Schedule	Using anonymous functions. Finding zeros of functions.		Using integral and trapz. Creating and evaluating polynomials. Using MATLAB's fit function.
Lab - Actual	LOs: More practice creating, and using, functions How to use fzero 1. Create stress function on three different plots with different parameters. (Creating function files) 2. Repeat 1 with fplot and an anonymous function ("Fixing" parameters with anonymous functions) 3. Use fzero() (with function file/anonymous function, plotting, EC: for loop)		LOs Using integral and trapz to integrate functions Creating and editing polynomials Very simple polynomial fit to data Use dlmread to read in data 1. Calculate and plot polynomials. Perform simple arithmetic. (Declare polynomials, linspace, polyval, fplot) 2. Use a polynomial to create data and then practice fitting (polyval, polyfit) 3. Integrate a function three ways: eyeball it, use trapz, and use integral with a function (EC: polyint)
HW - Detailed Schedule	HW5: Fzero, iterative functions, trapz/integral		HW6: Function fitting, optimization
HW - Actual	Goals: More practice with for loops and if statements More practice with using functions as inputs to otherfunctions More practice writing functions 1. Cannon problem: use fzero to find the time and x location where a projectile hits the ground (like HW 1 problem) 2. Find all the intersections of two functions (plotting, print x values, use for/while loop, fzero, array) 3. Epidemic 5: Determining Epidemic Peak (create a function IsEpidemic() to return if an epidemic occurs, when the max is, and when it happens. Then plot it. (Functions, plotting, min, max, loop) 4. EC: Use a function file in a Teacup Problem		Goals: Fitting with polynomials (both fitting a pure polynomial and using polynomial fitting to, for example, fit a logarithm) Integration using trapz/integral (In expectations: More practice re-using code. More practice with functions. More practicewith making code "general" e.g. using dlmread) 1. Fit data on melting ice using polyfit (plotting, polyfit, for loop) 2. Data linearization - simulate a dataset, transform the dataset four ways, use polyfit to create a first order polynomial, reconstruct function, plot all 3. Fix an existing script with two gaussian peaks. 4. Euler leaf problem 4 [2D] 5. EC: Integrate using a function and **quad**

	Notes	Week 7	Notes	Week 8	Notes	Week 9	Notes	Week 10	Notes	Other Notes
Syllabus	LO 2, 3, 4 Potential Issues: Students may be unfamiliar with interpolation and optimization. Matrices are not covered in curriculum until MTH 306.	Matrix variables, operations, and systems of linear equations	LO 2, 3 Potential Issues: Students have not learned matrix operations yet	MATLAB matrix operations (cont.) Multivariable functions and data interpolation	LO 2, 3 Potential Issues: Students may not have learned how to work with matrix operations, multivariable functions, or interpolation	Character and string manipulations Numeric-to-character data conversions (NUM2STR)	LO 1, 2	Introduction to 3-D graphics Review	LO 2, 4	Certain LO's can be assumed to be integrated into all assignments. Developing computer programs (LO 2) is integrated into all assignments. LO 5 - essentially debugging and validating work, is also integrated into all assignments.
Weekly Learning Objectives		Perform operations on matrices Create matrices using zeros, ones, eye, and [;] Get the size of a matrix using size Iterate over all elements of the matrix Access single rows and columns of a matrix (:,-) and (-,:) Perform arithmetic operations on matrices Rules for when two matrices can be multiplied together Transpose a matrix Create scale, rotation, and translation matrices and use them to position objects Adjust the degree of the polynomial to find the best fit to the data Apply transformations to the data in order to fit exponential, logarithmic, and inverse equations using polyfit with one degree Calculate a dot product between two vectors Use polyfit to fit a polynomial and then evaluate the polynomial Use interp1 to interpolate the samples Review of material for Midterm II, focused on weeks 3-6		Use matrices to set up, and solve, linear systems of equations Turn a set of linear equations into Ax = b matrix form Determine variables Rearrange terms Find coefficients of variables (A matrix) Find constant terms (B matrix) Print out values of variables Create functions that return multiple variables Define a parametric function Create a function file that returns multiple parameters Use the function to create points to plot Define a function in two variables Use meshgrid to create values for the function Evaluate and draw the surface		Creating and manipulating strings Putting strings together using strcat Converting numbers to strings using num2str Creating strings with sprint (same as fprintf) Formatting strings with white space for printing out tables Creating titles based on parameter values Creating file names Creating and manipulating arrays of strings Creating legends based on parameter values Using arrays of strings to set line styles mod re-visited to cycle through a list Manipulating strings as arrays of characters Using array notation to build strings Converting to ASCII codes and back Make strings cycle through letters Get out a substring from a string Reading and writing to files Define absolute versus relative path names Use subdirectories to save files Open a file and write to it using fprintf		Use the sphere and cylinder commands to make 3D shapes Drawing with surf and mesh Changing drawing parameters (colors, shading styles, camera view points) Interacting with 3D renderings Rotate the camera More practice with meshgrid Review for final for, if, and while control structures Function files Surfaces and 3D curve plotting Solving systems of equations Reading and writing to files Matrix multiplication Translation, scaling, rotation		
Textbook Reading (zyBooks)		7.1 2D arrays: Introduction 7.2 Elementary 2D arrays 7.3 Indexing an element in a 2D array 7.4 Multi-element 2D array indexing using integer arrays 7.5 Manipulating 2D arrays using a single colon 7.6 Psuedo-random number generators 7.7 Functions and two-dimensional arrays 7.8 Linear algebra I 7.9 Operators	Some of the linear algebra components can be beyond what students have learned so far in pre-requisties, as well as matrix operations in general.	8.1 3D line plots	Does not align with the other content.	9.1 Strings 9.2 Strings as arrays 9.3 Constructing strings 9.4 Arrays of strings		10.1 Meshgrid 10.2 Surface and mesh plots		
Lecture (Mon) - Detailed Schedule		Lec 7-1 matrix basics Creation, editing, matrix mathematical operations Lec 7-2 matrix operations Rotate, Scale, translate		Lec 8-1 Systems of equations		Lec 9-1 Strings 1 Lec 9-2 Writing to files		Lec 10-1 Surfaces		
Lecture (Mon) - Actual	Calculus pre-requisites. Not sure about examples with doing polynomial arithmetic on paper. Lecture names really don't match with detailed schedule fit function involves structs, which is taught nowhere else in the course.	Lec 7-1 Matrix basics Matrices: Creating and accessing Matlab matrix functions Matrix multiplication zeros(), ones(), eye() size(), max min sum mean, transpose(A), A', diag(A), flipud(), fliplr() (wiht visual examples) Calculations: element-by-element, dot(), cross(), matrix (with visual examples) Code for cross multiplication Code example of matrix multiplication Example of matrix multiplication for failure modes Lec 7-2 Matrix multiplication Using matrix multiplication Specific types of matrices Rationale for matrix multiplication (animation and analysis) Scaling matrices Matrix rotation Matrix Translation Combining matrices	Matrices are covered in MTH 306. Not much practice of doing multiplication by hand.	Lec 8-1 Matrix equations Solving Simultaneous Systems of Equations (using matrix algebra) Exercise 1: Build polyfit (degree 1), shows code Exercise 2: Repeat for degree 2 (optional), shows code Present Graph	These are linear algebra concepts which are not taught until MTH 306	Lec 9-1 strings I Alphanumeric data What is alphanumeric data? chars, strings (text) Manipulating Strings strcat, fprintf, sprintf, cells {} Converting from numbers to strings and back again num2str, str2num Ways to manipulate alphanumeric data Concatenation: strcat, horizontal concatenation num2str, str2num, sprintf Exercise 1: Titles with variables in them, answer + code solution Arrays of strings + code example strcmp Lec 9-2 strings II Alphanumeric Data How does matlab store strings? arrays of characters Converting from numbers to strings and back again ASCII encoding Manipulating strings as arrays, code example String concatenation methods Exercise 1: Create a legend Show answer Exercise 2: Create numbered file names Writing to a folder Converting to ASCII numbers		Lec 10 Surfaces meshgrid review 3D surfaces Using matrices with surfaces (optional) Plotting 3D surfaces with meshgrid Plot a cylinder Plot a sphere in a grid Various examples of shapes		
Lecture (Fri) - Detailed Schedule		Midterm II Function fitting, integration, optimization, roots, fzero		Lec 8-2 Multi-variable functions		Lec 9-3 Strings II		Review		
Lecture (Fri) - Actual	Can't tell how much of Lec 6-2 is taught here.	Midterm II 1 - [Y/N] trapz & integral inputs 2 - [MC] Which is less accurate: interp, extrap, evaluate function directly 3 - [MC] Select appropriate procedure for estimating value from data (polyval, polyfit) 4 - [MC] Which has only one output? (anonymous function) 5 - [MC] Best strategy to integrate area of a golf course 6 - [Write script] Read data, fit quadratic polynomial, print result in quadratic equation form 7 - [Write script] Ask for "agent" ID and passcode and compare against matrix of codes 8 - [Write script] Create a MATLAB function and plot the function over a graph 9 - [Write script] Calculate the area under a polynomial with two different methods 10 - [Write script] Create a function that finds the max of an array w/o using max()	There is a lot of script writing for this problem. It is difficult to know if this can be completed in a reasonable time.	Lec 8-2 Multi-variable Multi-variable plotting meshgrid interp2 Plotting parametric functions (one parameter, two parameter) Code version of one parameter Code of two parameter Example of using meshgrid (written) Code to plot surfaces (surf, mesh, aesthetic commands) Walkthrough of plotting a surface (code on slides) Interpolating data, code example	Code is still difficult to see. This material all relates to week 10, making week 9 an odd placement.	Lec 9-3 files How do I save my output? Answer: Write it to a file fopen, fclose, fprintf with a filename Already seen dlmwrite Writing formatted output with fid, fprintf, fclose Example code Reading from files	Numbers appear to be off for lectures.			
Lab - Detailed Schedule		Function fitting. Data point interpolation. Basic matrix creation and editing.		Systems of linear equations		String manipulation to create titles and file names. Reading and writing data to files. 3D curve and surface.		Surfaces, interp2		
Lab - Actual	dlmread is not covered in lecture. It is not put in as a learning objective.	LO Practice turning subscript equations into matlab code Creating and accessing matrices Matrix multiplication Interpolating samples with interp 1. Write a function that generates points, the interpolate over it 2. Create a matrix and perform matrix multiplication on points 3. Wrtie your own integration function and compare it to trapz		LOs Solving systems of linear equations 1. Solve a basic system of linear equations 2. Solve a very simple system of one equation for voltage values 3. Create your own version of polyfit through solving a system of linear equations 4. EC: Use ginput for polyfit	Problem 2 uses the concept of Kirchoff's Voltage Law, which is not taught until ENGR 201	LOs String manipulation More reading and writing to files Making a simple curve and surface 1. Create and title four subplots using string manipulation 2. Create a matrix of values and write out the information to a txt file 3. Read matrix data in and write out the output to 4 files 4. Make a 3D curve using a function file and plot3 5. EC Make a pretty surface		LOs: Practice with interp2. Making and plotting surfaces. 1. Plot the data in 3 text files to create a cylindrical shape 2. Interpolate across the 3D data 3. Plot the new surface. 4. EC: Find the mean of each ring in the cylinder and plot it.		
HW - Detailed Schedule		HW7: Interp1, matrices		HW8: Matrics, 3D plotting, meshgrid, interp2		HW9: Reading and writing files, string manipulation, surfaces		Final: Matrices, strings, multi-variable functions, surfaces, meshgrid		
HW - Actual	Problem 3 requires knowledge of what a Gaussian is and how to use the struct., which is covered nowhere in lecture. Quad is briefly covered in lecture in one slide, but there is no practice or demonstration with it.	Goals: Use of interp1 for interpolation Matrix operations (creating, multiplying) Problems 1.Use a dot product to calculate the total mass of rocket components 2. Simpson's Rule, using a function file for Simpson's rule which iterates over a given function until the difference between the two latest calculated areas are within a target tolerance 3. Teacup Problem 5 [Matrices] Use matrices to perform translation and rotation 4. Euler Leaf Problem 5 [Integration] Calculate the distance a particle travels using positions as well as integrating velocity vector lengths 5. EC: Monte Carlo to calculate pi based on the ratio of the area of a circle to the area of a square	There were some obvious errors with this file since there is an updated file for the HW 7 with different Simpson files. Simpson's rule is covered in week 5. Problems 2, 4, and 5 don't align with the goals of the homework. It doesn't look like interpolation is used for any of the problems.	Goals: More fun with matrices Solving systems of linear equations Using meshgrid to create samples for two parameter functions Interp2 for interpolating data 1. Use a system of linear equations to solve for currents using Kirchoff's voltage law 2. Plot a surface using a two parameter function and interpolate over it 3. Plot a spiral staircase using parametric equations 4. Teacup Problem 6 [Pretty pictures] Use various functions, loops, and matrices to plot a "pretty picture" 5. EC: Richter magnitude scale: Fit different polynomials fits to earthquake data and select the best equation	5 Uses material from week 6, but this is fine since it is an extra credit problem.	Goals: Creating and manipulating strings for labels on graphs, filenames 3D surface plotting Reading and writing files 1. Caesar cipher; shift ASCII data read from txt file and write answers to txt files 2. Create a surface of revolution using parametric functions and a mesh 3. Epidemic Problem 6 - Create a FindEpidemic() function and use it with fzero to determine the starting population required to start an epidemic 4. Euler Leaf Problem 6 - Use the Euler functions with 3D parameters and plot various particles in various conditions	The last epidemic problem was in HW 5. This epidemic problem does not really align with the homework goals.			

References

Abdulmalek, F. A., Rajgopal, J., & Needy, K. L. (2006). A classification scheme for the process industry to guide the implementation of lean. *Engineering Management Journal*, *18*(2), 15–25.

Ahlstrom, P. (2004). Lean service operations: translating lean production principles to service operations. *International Journal of Services Technology and Management*, *5*(5–6), 545–564. https://doi.org/10.1504/IJSTM.2004.006284

Alagaraja, M. (2010). Lean thinking as applied to the adult education environment. *International Journal of Human Resources Development and Management*, *10*(1), 51–62.

Bailey, T., & Forbes, J. (2005). Just-in-time teaching for CS0. In *ACM SIGCSE Bulletin* (Vol. 37, pp. 366–370). ACM.

Balzer, W. K. (2010). *Lean higher education: Increasing the value and performance of university processes*. CRC Press. Retrieved from https://books.google.com/books?hl=en&lr=&id=25aZhInmEDQC&oi=fnd&pg=PP1&dq=Lean+Applications+to+Course+Design&ots=usYkJjCSBi&sig=DbUTs3z-KNjmaNSzakr9E6x-pWM

Balzer, W. K., Brodke, M. H., & Kizhakethalackal, E. T. (2015). Lean higher education: successes, challenges, and realizing potential. *International Journal of Quality & Reliability Management*, *32*(9), 924–933.

Balzer, W. K., Francis, D. E., Krehbiel, T. C., & Shea, N. (2016). A review and perspective on Lean in higher education. *Quality Assurance in Education*, *24*(4), 442–462.

Biggs, J. (1996). Enhancing teaching through constructive alignment. *Higher Education*, *32*(3), 347–364.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (2000). *How People Learn: Brain, Mind, Experience, and School* (2nd ed.). Washington, D.C.: National Academies Press.

Brassard, M., & Ritter, D. (2010). *The Memory Jogger 2* (Second). Salem, NH: GOAL/QPC.

Brouwer-Hadzialic, L., & Wiegel, V. (2016). Case study: using value stream mapping in the educational process - how focusing on student actions can help depict the teaching process. *International Journal of Six Sigma and Competitive Advantage*, *10*(1), 50–61. https://doi.org/10.1504/IJSSCA.2016.080451

Cudney, E. A., Venuthurumilli, S. S. J., Materla, T., & Antony, J. (2018). Systematic review of Lean and Six Sigma approaches in higher education. *Total Quality Management & Business Excellence*, *0*(0), 1–14. https://doi.org/10.1080/14783363.2017.1422977

Davidson, C. (2017). *The New Education*. Hachette Book Group.

Dennis, P. (2016). *Lean Production Simplified: A Plain-Language Guide to the World's Most Powerful Production System* (3rd ed.). Baco Raton, FL: CRC Press.

Dey, A. K. (2007). A lean approach to improve course curriculum of MBA. *Business Perspective*, *9*(2), 109–128.

Douglas, J., Antony, J., & Douglas, A. (2015). Waste identification and elimination in HEIs: the role of Lean thinking. *International Journal of Quality & Reliability Management*, *32*(9), 970–981.

East, J. P., & Schafer, J. B. (2005). In-person Grading: An Evaluative Experiment. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (pp. 378–382). New York, NY, USA: ACM. https://doi.org/10.1145/1047344.1047472

El-Sayed, M., El-Sayed, J., Morgan, J., & Cameron, T. (2011). Lean program and course assessments for quality improvement. *International Journal of Process Education*, *3*(1), 65–72.

Emiliani, B. (2015). *Lean teaching: A guide to becoming a better teacher*. The CLBM, LLC, Wethersfield, CT.

Emiliani, M. L. (2004). Improving business school courses by applying lean principles and practices. *Quality Assurance in Education*, *12*(4), 175–187.

Emiliani, M. L. (2005). Using kaizen to improve graduate business school degree programs. *Quality Assurance in Education*, *13*(1), 37–52. https://doi.org/10.1108/09684880510578641

Emiliani, M. L. (n.d.). Evolution of Lean Teaching. Retrieved from https://bobemiliani.com/wp-content/uploads/2016/09/elt_paper.pdf

Enrollment Summary - Spring Term 2017. (2017). Office of Institutional Research, Oregon State University. Retrieved from http://institutionalresearch.oregonstate.edu/sites/institutionalresearch.oregonstate.edu/files/enroll-spring-2017.pdf

Ertmer, P. A., & Newby, T. J. (2013). Behaviorism, Cognitivism, Constructivism: Comparing Critical Features From an Instructional Design Perspective. *Performance Improvement Quarterly*, *26*(2), 43–71. https://doi.org/10.1002/piq.21143

Flumerfelt, S., Kahlen, F.-J., Alves, A. C., & Siriban-Manalang, A. B. (2015). *Lean engineering education: Driving content and competency mastery*. Momentum Press. Retrieved from https://books.google.com/books?hl=en&lr=&id=c3HtCQAAQBAJ&oi=fnd&pg=PT7&d

q=lean+engineering+education:+driving+content+and+competency+mastery&ots=ApJXSwU_MR&sig=SoSayRyS5vtSDeL2Da2aQzwOryA

Glesne, C., & Peshkin, A. (1992). *Becoming Qualitative Researchers*. Longman, 10 Bank Street, White Plains, N. y. 10606: Longman Publishing Group.

Höök, L. J., & Eckerdal, A. (2015). On the Bimodality in an Introductory Programming Course: An Analysis of Student Performance Factors. In *2015 International Conference on Learning and Teaching in Computing and Engineering* (pp. 79–86). https://doi.org/10.1109/LaTiCE.2015.25

Jacobs, H. (1997). *Mapping the Big Picture - Integrating Curriculum & Assessment K-12*. Alexandria, VA: Association for Supervision and Curriculum Development.

Karlsson, C., & Ahlström, P. (1996). Assessing changes towards lean production. *International Journal of Operations & Production Management*, *16*(2), 24–41.

Krehbiel, T. C., Ryan Jr, A. W., & Miller, D. P. (2015). Lean Learning. *Quality Progress*, *48*(2), 39.

Liker, J. (2004). *The Toyota Way*. New York, NY: McGraw-Hill.

Locke, K. (2002). The grounded theory approach to qualitative research. Retrieved from http://psycnet.apa.org/psycinfo/2001-05785-001

Mansur, A. F. U., Leite, F. C., & Bastos, H. P. P. (2017). Lean Education for Applied Science Universities: A Proposal by Federal Institutes of Applied Sciences in Brazil. In A. C. Alves, S. Flumerfelt, & F.-J. Kahlen (Eds.), *Lean Education: An Overview of Current Issues* (pp. 25–40). Springer International Publishing. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-45830-4_2

Miles, M. B., Huberman, A. M., & Saldana, J. (2014). *Qualitative data analysis: A method sourcebook* (3rd ed.). CA, US: Sage Publications.

Mirdad, W., & Eseonu, C. (2015). A Conceptual Map of the Lean Nomenclature: Comparing Expert Classification to the Lean Literature. *Engineering Management Journal*, *27*(4), 188–202.

Mirdad, W., & Eseonu, C. (2017). A cause-effect strategy map for lean process transformation. *Int. J. System of Systems Engineering*, *8*(2), 121–146.

Montgomery, D. C. (2013). *Introduction to Statistical Quality Control* (7th ed.). John Wiley & Sons Inc.

Nathan, M. J., Koedinger, K. R., & Alibali, M. W. (2001). Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In *Proceedings of the third international conference on cognitive science* (pp. 644–648).

Nathan, M. J., & Petrosino, A. (2003). Expert Blind Spot Among Preservice Teachers. *American Educational Research Journal*, *40*(4), 905–928. https://doi.org/10.3102/00028312040004905

Nicholas, J. (2011). *Lean Production for Competitive Advantage*. New York, NY: Taylor and Francis Group, LLC.

Ott, C., Robins, A., & Shephard, K. (2016). Translating Principles of Effective Feedback for Students into the CS1 Context. *Trans. Comput. Educ.*, *16*(1), 1:1–1:27. https://doi.org/10.1145/2737596

Pavlović, D., Todorović, M., Mladenović, S. jan, & Milosavljević, P. ja. (2014). The role of quality methods in improving education process: Case study. *Serbian Journal of Management*, *9*(2), 219–230.

Philips, D. C., & Soltis, J. F. (2009). *Perspectives on learning* (5th ed.). New York, NY: Teachers College Press.

Pre-engineering course requirements by major. (n.d.). Retrieved February 25, 2018, from http://engineering.oregonstate.edu/pre-engineering-course-requirements-major

Pyzdek, T. (2003). *The Six Sigma Handbook* (2nd ed.). New York, NY: The McGraw-Hill Companies, Inc.

Sadler, D. R. (1989). Formative assessment and the design of instructional systems. *Instructional Science*, *18*(2), 119–144.

Shevlin, M., Banyard, P., Davies, M., & Griffiths, M. (2000). The Validity of Student Evaluation of Teaching in Higher Education: Love me, love my lectures? *Assessment & Evaluation in Higher Education*, *25*(4), 397–405. https://doi.org/10.1080/713611436

Simkins, S., & Maier, M. (Eds.). (2010). *Just-in-Time Teaching* (1st ed.). Sterling Virginia: Stylus Publishing, LLC.

Suárez-Barraza, M. F., & Rodríguez-González, F. G. (2015). Bringing Kaizen to the classroom: lessons learned in an Operations Management course. *Total Quality Management & Business Excellence*, 1–15. https://doi.org/10.1080/14783363.2015.1068594

Tatikonda, L. (2007). Applying Lean Principles to Design, Teach, and Assess Courses: By Applying Principles and Techniques Developed in Industry, Educators Can Refine the Content, Pedagogy, Organization, and Assessment Methods Employed in Their Accounting Courses to Help Ensure That Students Gain the Knowledge and Skills That Will Make Them Most Desirable to Employers. *Management Accounting Quarterly*, *8*(3), 27.

Thomas, A., Antony, J., Haven-Tang, C., Francis, M., & Fisher, R. (2017). Implementing Lean Six Sigma into curriculum design and delivery – a case study in higher education. *International Journal of Productivity and Performance Management*, *66*(5), 577–597. https://doi.org/10.1108/IJPPM-08-2016-0176

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). *Facilities planning* (4th ed.). Hoboken, NJ: John Wiley & Sons Inc.

Vihavainen, A., Airaksinen, J., & Watson, C. (2014). A Systematic Review of Approaches for Teaching Introductory Programming and Their Influence on Success. In *Proceedings of the Tenth Annual Conference on International Computing Education Research* (pp. 19–26). New York, NY, USA: ACM. https://doi.org/10.1145/2632320.2632349

Vukadinovic, S., Djapan, M., & Macuzic, I. (2017). Education for Lean and Lean for Education: A Literature Review. *International Journal for Quality Research*, *11*(1). Retrieved from http://www.ijqr.net/journal/v11-n1/3.pdf

Waterbury, T. (2011). *Educational lean for higher education: Theory and practice*.

Willingham, D. T., Hughes, E. M., & Dobolyi, D. G. (2015). The Scientific Status of Learning Styles Theories. *Teaching of Psychology*, *42*(3), 266–271. https://doi.org/10.1177/0098628315589505

Womack, J. P., & Jones, D. T. (2003). *Lean Thinking* (2nd ed.). New York, NY: Free Press.

Womack, J. P., & Jones, D. T. (2010). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation* (2nd ed.). Simon and Schuster. Retrieved from https://books.google.com/books?hl=en&lr=&id=2eWHaAyiNrgC&oi=fnd&pg=PA13&dq=lean+thinking&ots=2LQ0us6lSm&sig=8IzqoYJqtA31jbYvbBnDOD07Al0

Womack, J. P., Jones, D. T., & Roos, D. (1990). *The Machine That Changed the World: The Story of Lean Production*. New York, NY: HarperPerennial.