ISSN: 2167-1435
v31 n3072

Where do Student Outcomes Begin? Developing Personal and Life Skills as a Strategy for Student Success in the First Computing Course and Beyond

Sean Humpherys
shumpherys@wtamu.edu

Jeffry Babb
jbabb@wtamu.edu

Amjad Abdullat
aabdullat@wtamu.edu

West Texas A&M University
Canyon, TX 79016

Abstract

Through the annual ABET assessment process, Computer Information Systems faculty in the College of Business at a regional institution were able to diagnose problems regarding students not satisfying our program's Student Outcomes. Often, the impediments to student success were not technical in nature and prompted faculty to consider non-technical solutions. A framework for the interaction and interdependency of the technical and non-technical skills, termed the "Blue and Green Curve" is presented that describes balancing emphasis on the technical curriculum (Blue Curve) and professional/life skills (Green Curve) that have the potential to accelerate students' acquisition of technical skills. The framework prescribes a foundation of effective habits to establish early in a student's academic career. Guided by this framework, changes to a freshman-level programming fundamentals course in the CIS program are described. Pedagogical tools with assignments and rubrics that support the curriculum changes are shared. These changes, among others factors, may have contributed to a 10% increase in the freshman persistence rate over a two semester period.

Keywords: student outcomes, programming fundamentals, computing course, student success, technical pedagogy, and professional skills

1.INTRODUCTION

There remains ample evidence that the demand for jobs in the Information Technology sector - for systems analysts, database administrators, software developers, web and mobile developers - remains high (Burns et al., 2014). In fact, the U.S. Bureau of Labor Statistics (BLS) continues to forecast growth in the 2010 to 2020 timeframe, despite challenges to enrollment in computing-related majors in the US over the last decade. While fulfilling this demand is certainly within the capabilities of programs in Computer Information Systems (CIS), attracting students into our programs is only the beginning; retention, development, and successful matriculation are often the real challenges. Our programs and curricula must produce successful outcomes that lead to a fruitful career in the computing field where lifelong learning and growth are the hallmarks and objectives of our graduates' careers.

Student retention is a somewhat universal problem in higher education that is not unique to the computing disciplines. Various solution approaches to this issue have been explored: approaches to group work and pairing to improve basic education in computer programming (McDowell et al., 2006); developing undergraduate research initiatives and programs (Peckham et al., 2007); focusing on improving the gender gap in computing (Cohoon, 2002; Mishra et al., 2014); However, the approach we advocate for in this paper is to manage and design a careful blend of "life skills" to increase patterns of maturity with technology skills to build professional competency.

The paper proceeds in this argument in the following manner: First, we examine the role of habits for student success; Next, we use the results of ABET program assessment to diagnose students' lack of proper habits as contributing to failure in the first programming course. We then propose a balance between "life skills" and "technical skills" to develop habits for mastery. We propose a solution approach that describes changes made in our program in the introductory programming class. We next examine the challenges we face to assess our approach and its effectiveness. We also consider the implications of our solution approach as they pertain to ABET-accreditation and AACSB accreditation, as our program is shaped by both. We close with future research and empirical directions to balance two learning curves - that of personal and professional development (maturity) and that of technical competency. We call these "green" and "blue " curves, respectively.

2. DIAGNOSING THE PROBLEM

A significant portion of our students are first-generation college degree seekers, from minority groups, or both. Subsequently, due to a number of these factors, not all of our students are ideally or optimally prepared for college. Our ABET and AASCB assessment processes have corroborated this under preparedness for college-level work and reveal ineffective learning habits: a lack of self-organization skills, an inability to envision and adequately plan for future possibilities, and a propensity to retreat from challenges and adversity. We accept that these are also challenges faced in other institutions over the last decade or so (Elliott et al, 1990; Rendon, 1995; Wise, 2008).

Reflections Using Multiple Sources

Our diagnosis of poor habits among our students was drawn from four sources: interviews with students corroborated with consultation among faculty, ABET assessment data, and feedback from employers and Industrial Advisory Board members. Feedback from theses sources suggested the need for an intervention. We discuss each data source to better illustrate the problem.

Reflections From Interviews With Students

In our program, the faculty meet with students at least once during the semester for advising. During these discussions, we become familiar with the students and assess their goals. As a result of this process, we have detected some troubling trends. Often, students couldn't articulate well-defined career goals or a defined plan beyond selecting which courses to take next semester. Furthermore, they relied solely on faculty to recommend courses rather than making a plan themselves. Additionally, students rarely participate in internships and had inadequate resumes (if they possessed a resume at all).

As a department we gathered empirical evidence describing several symptoms correlated to the lack of performance by students, particularly those students who were receiving poor or unsatisfactory scores. As an example of the close ties our professors have to the student facilitating observations, we describe that two of our professors interview the students in each course taught. Another professor regularly interacts through a technology club and a web development club. And all professors are involved in formalized student advising each semester.

Symptoms include: a lack of time management and no use of a calendaring system, no predefined study times except for athletes who have required study hall hours, lack of propensity to seek help from the professor, the tutor, or fellow students, working (i.e., employment) too many hours preventing time to work on assignments, no career plan or idea of what they want to do with their degree even as graduating seniors.

Reflections From ABET Assessment Outcomes

Our AACSB and ABET assessment and evaluation activities include student outcomes evaluation to determine whether our students are performing at a satisfactory level. When student outcomes are not met, we identify and ameliorate the causes with corrective action. As some of our ABET student outcomes were unsatisfactory we discovered a "missing data" problem: students were failing to submit their assignments. We approached this as behavioral issue that must be addressed in order for learning to transpire.

Our solution was to focus on our programming fundamentals class, taught predominantly to freshman. As example one course learning objective is to create an application using the Input->Process->Output programming paradigm which is mapped to our ABET Student Outcome 1: "An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution." In the Spring of 2014, 23% of the students (7 out of 31) scored unsatisfactory on the assignment. This is a poor result that runs dangerously near the 30% threshold set by our program for acceptable levels of poor and unsatisfactory performance. In this case, only one student of the seven failing students actually submitted an assignment. As educators, we must address the need to develop an entirely different skill set that was the course material calls for; we must also visit the skill set that develops the internal "compass" where submitting assigned work is accepted as part and parcel of success. While we could meet this reality with indifference, this is behavior we can ill afford to abide. We hold this position for two reasons: 1) Information Systems programs continue to suffer from low enrollments for a variety of reasons of the last 15 years (Granger et al., 2007; Burns et al., 2014); and, 2) the skills CIS graduates possess remain highly valued among the employers.

Reflections From Employers And Advisory Board Members

Our ABET process includes regular meetings with our program's industrial advisory board members. Some of these employers also collaborate with us by hosting projects for students to complete in our capstone systems development course. When we asked these stakeholders what skills they desire from our students, they included the following items beyond the expected technical skills:

1. Ability to collaborate and work in teams.
2. Time management skills and ability to deliver on time
3. Critical thinking to solve business problems
4. Ability to communicate well orally and in written form

We then compared our student outcome goals to the list of desired skills for employers and found some deficiencies. For example, historically we have used group projects in several courses as a mechanism to foster collaboration and teamwork. However, upon reflection, the college does not teach about how to be successful in small group collaboration. Further, our assessment of collaboration skills was limited to end-of-project peer; this is not sufficient as effective collaboration skills should be taught earlier. We also found that the other courses in the business curriculum were either not explicitly teaching these collaboration skills or covering them later in the curriculum cycle. However, employers were telling us that they valued these "life" skills as much as technical skills.

Research questions

The assessments and reflections led us to develop the follow research questions.

1. Besides technical skills, what other professional skills are needed by our students to prepare them for successful careers?

2. What frameworks and tools might be useful to include in teaching non-technical skills to students?

3. Where should these skills be taught? At the beginning, middle, or end of a student's academic career?

4. What results have been observed from experimenting with different educational treatments?

This paper shares our findings in hopes that other universities experiencing similar problems can implement some of the same solutions.

3. FRAMEWORK FOR INCREASING MET STUDENT OUTCOMES

As it became clear that some of our students lacked the habits to transition successfully to college-level work, our approach was to "put first things first." We started with our introductory programming course, as it is oriented towards freshman. Focusing on freshmen is the appropriate first step as it is well documented that the first-year experience is generally challenging (Clark, 2005). Our solution approach draws upon the lessons of Covey's (1997) seven effective habits. For instance, on the point of self-organization, Covey (1997) calls for taking a proactive stance as an effective habit. Commensurate with Covey's admonition to "begin with the end in mind," we found that freshmen demonstrated great difficulty with prioritization.

Going Beyond the Technical

With a few exceptions, ABET's Student Outcomes (SOs) are technical or analytical in nature. In Appendix D, we list how we have adopted ABET's Computing Accreditation Commission's SOs ***a*** through ***i***, plus the ***j*** outcome SOs in our CIS program. A clear majority of these SOs are technical or analytical and are assessed with assignments that require the design, development, testing, and implementation of technical artifacts. Other SOs specifically address analytical skills, such as writing a paper or reflecting on ethical considerations. Several other SOs, however, require communication skills (typically classified as a soft-skill) and collaboration within a team.

Given the stated objectives of the employers participating in program's Industrial Advisory Board the "life" skills are not fully "captured" within ABET's SOs. In order to meet our SOs, we as faculty must make up the difference. Underdevelopment or lack of these "soft" skills could impact the student's ability to perform on-time and to meet real-world project objectives.

As we reflected on these deficiencies in our students, we identified two sets of skills worth distinguishing. Notionally, we have labeled growth and development in the acquisition towards mastery of technical skills as the "Blue Curve." Correspondingly, we labeled the arc of progress in the acquisition and mastery of the non-technical skills as the "Green Curve." The "Green Curve" includes skills such as time management, study habits, prioritization, goal seeking, drive/motivation, management of risk/failure, continued personal and professional development outside the classroom, etc.

We theorized that the time spent on focusing on Green Curve skills early in a student's education, with continuous reinforcement throughout the curriculum, could develop the personal traits needed to succeed in acquiring technical skills (to ascend the Blue Curve). Our resolve was to spend more time helping students acquire Green Curve skills earlier in the program as a strategy for greater success in acquiring blue curve skills (technical skills). We propose that early care in the Green Curve may accelerate the acquisition of Blue Curve due to the students' successful habits.

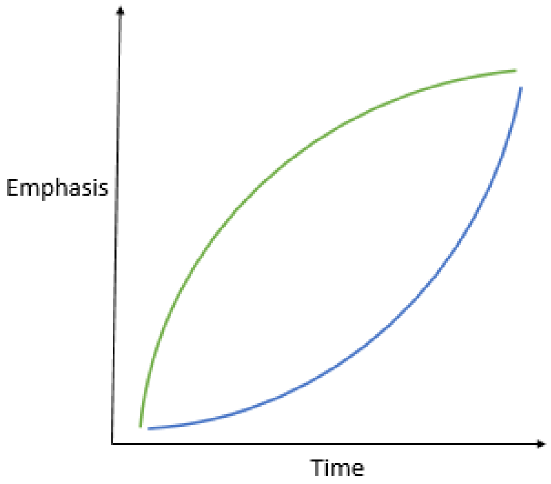


Figure 1. The Blue and Green Curve Model Towards Mastery. The green curve is the upper curve in this figure.

Figure 1 demonstrates the simplicity of the concept. The ***y*** axis represents a scale suggesting the amount emphasis and time spent on either Blue or Green curve skills. The ***x*** axis correspond to progression through the curriculum, from freshman through to graduation. While the model may be presented in an oversimplified manner for parsimony, the Green Curve is emphasized earlier so that benefits of Green Curve skills can help them acquire technical (Blue Curve) proficiency later. With the early acquisition of Green Curve skills, we propose that professors can expect higher performance in technical areas in later years. Of course, we do not entirely present this idea as original. The First-Year Experience movement in higher education has long suggested that a focus on adapting and forming the correct attitudes and habits be accepted as a pre-cursor to rigorous engagement in discipline-oriented study (Upcraft and Gardner, 1989). Furthermore, this idea has been promulgated within the CIS educators (Saulnier et al., 2006).

Covey’s Transitions

Our Blue Curve/Green Curve model is consistent with how Covey (1997) describes the relationships and transitions among the seven habits. In keeping with Covey’s model, the Blue and Green Curves should lead to Interdependence (see Figure 3). For our students, interdependence blends proficiency in leadership, business acumen, and continuous improvement in technology.

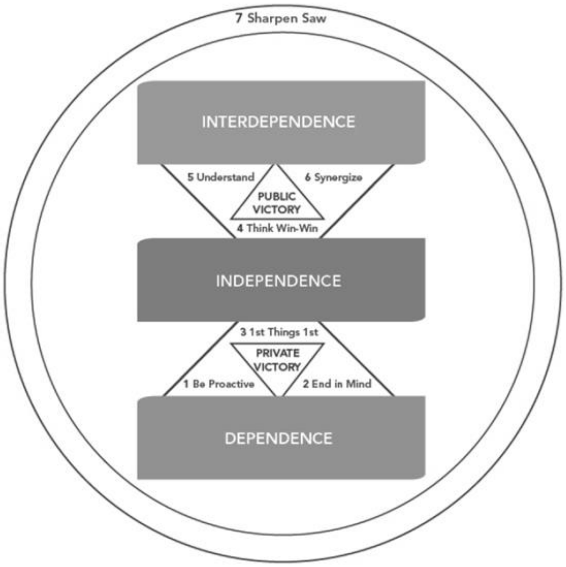


Figure 2. Covey’s (1997) Transitions

Covey’s (1997) model provides a notional guide for milestones in student achievements both the Blue and Green Curves. As freshmen/sophomores, students develop independence with the basic habits(Figure 3). Achieving the basic habits is a “private victory” in that it enables their own self-organization and orientation. This independence will be required for success in upper-level coursework and electives. It is our desire that this leads to “public victories” such that students engage in extra-curricular and co-curricular activities designed to further elicit progress on the Blue Curve. In relating our Blue Curve/Green Curve to Covey’s model (Figure 3), we reflect the role that non-technical soft skills play towards mastery and success in our program and in the discipline. An ultimate test of the efficacy of this approach is reflected in Covey’s (1997) seventh habit: to “sharpen the saw.” Our aim is to develop self-motivating and self-sustaining professionals who embrace the change and challenge inherent in the discipline. We presume that these curves move on indefinitely in the process of Covey’s seventh habit.

Challenged to Transform

Covey’s (1997) transformation model (Figure 2) is also useful when we consider the context and situation of our own institution: our implicit and tacit mission is to effect a transformation in those students who have not been given a fuller set of opportunities to develop successful habits. Our challenge in a rural and regional institution is take students who, on a scale of 1 to 10 in terms of college-preparedness, who are at the 2 or 3 level and elevate them to the 7 or 8 level. Early Green Curve emphasis gives us better options and probability of later success.

Thus far, we have witnessed some success with our Blue and Green Curve model. Recent graduates had entered our program with marginal Green Curve skills, and subsequently progressed along both curves to great success. We have graduates in major metropolitan IT markets making six-figure salaries within a few years of graduation who are living examples of our “take them from 2s and 3s to 7s and 8s” perspective. Of course there are outliers. We have 4.0 honor students who, either innately or through other means, have the Green Curve in hand and progress superbly on the Blue Curve without truly needing us. We also have students who, for whatever reason, can’t or won’t respond to any of our interventions and treatments and ultimately fail and/or choose something other than higher education. Given our program’s context, the Blue and Green Curve model is particularly useful for reaching what we call the “middle cohorts” - those for whom we would most able to effect transformation and for whom careful attention to the curves matters (Babb et al., 2014).

Who is responsible for ensuring students learn effective habits?

While our conclusion about soft skills and personal management skills were justified, we found these conclusions somewhat problematic to implement. In a technical program, whose responsibility is it to teach non-technical skills such as time management, goal setting, etc.? At the university level, freshmen often have a -required, first-year experience course to teach students how to be academically successful . At the college-level similar courses have also recently been introduced. However, at the University level these courses were inconsistently coordinated and the observed symptoms still remained. Anecdotal feedback from students suggested that these University-level courses were "blow off" classes that they don't learn much from. At the College level, some good courses in business communication and finance were offered to also address the "soft" and "life" skills, but they do not weave our discipline's Blue Curve. Also at the college level, a management course does reading of Covey's (1997) material, but this course is typically not taken untill the junior or senior year. Additionally, this exposure was only in one class and not systematically reinforced or emphasized. Thus, the "cure" was too little too late.

Since the number one reason for our unsatisfactory ABET SOs in the introductory course according to non-technical problems, we elected to deal with the problem directly at the start of our own curriculum. For student success, we had to add Green Curve to our program curriculum at the earliest possible time, in the introductory programming course; waiting for others to solve the problem was too risky. We are also accountable and responsible to all of our constituents – to students and to the employers of our students – to act.

4. CASE STUDY

This section of the paper will outline the various tools we used to implement blue and green curve into our curriculum. We'll discuss the tools by course, but nothing prevents the adoption of these tools throughout the curriculum or the insertion of other ideas and solutions. The goal is to increase Green Curve earlier into the curriculum so that students can benefit maximally throughout their academic experience. And, the goal is to have students take both the acquired blue and green curve skills into the workplace.

Green Curve's Learning Objectives introduced in the Beginning Programming Course

The technical course learning objectives for the beginning programming course focus on the acquisition of HTML, CSS, and JavaScript coding skills and is intensely hands-on with daily learning activities and assessment through project assignments.

To introduce Green Curve skills, students had several reading assignments from the book 7 Habits of Highly Effective Personal Workbook by Steven R. Covey. The personal workbook was chosen over the full-text version of the book for three reasons: 1. It is a faster and easier to read, 2. It focuses on taking action not just reading, and 3. The workbook is also used in a junior level business management course and we desired to leverage multiple exposures to the same principles.

Specific Learning Objectives:

The specific green curve learning objectives have been inserted into the beginning-programming course and describe student outcomes as abilities. The Specific Learning Objectives are listed in Appendix E.

Several of these learning objectives were incorporated into an assignment called "Career and Life Portfolio" which focuses on habits 1, 2, and 3 of Covey's 7 Habits of Highly Effective People tailored to academic success. See Appendix C for instructions and grading rubric used with this portfolio.

Three lectures were added to the course schedule to discuss the 7 Habits of Highly Effective people with reading assignments from the 7 Habits of Highly Effective personal workbook. Lectures consisted of student giving short oral presentations over the habits, discussing the habits in small groups and as a class, and doing a few of the personal exercises in the workbook. Emphasis was placed on the Career and Life Portfolio assignment as it manifests the habits in tangible form. What is provided in this paper and in the appendices is just a sample of our implementation.

Blue Curve Changes to the Beginning Programming Course

In five years of ABET assessment and reflection on this course, the professors who taught the course emphasized the replacement of multiple-choice exams with projects demonstrating skill acquisition. Quiz and multiple-choice questions are still used, but are less prominent and used to reinforce terminology and definitions. The projects include creating a multiple page website using HTML and CSS and a JavaScript-based application demonstrating implementation of an input, processing, output paradigm. The results were an incremental improvement in ABET outcomes and an increase in the cognitive-level of demands in the assignment.

The following section showcases several tools used to introduce coding or to practice coding through hedonic experiences. The authors have no affiliation with the applications showcased but have used each in the programming fundamental courses to enhance learning and motivate students' practice of logical thinking.

Learning Activities: Daily lectures culminated in several learning activities created by the instructor to practice the specific learning objectives lectured upon. Lectures were limited to 20-30 minutes of the course and then students performed hands-on learning activities that practiced what was just lectured on or expanded into related learning objectives.

Weekly Calendaring Exercise: Each week students had to demonstrate the use of a calendaring system that identifies a month's worth of academic deliverables and the dates due and pre-planned study times. Students were required to record the due date of each deliverable for all courses, not just the programming fundamentals course. Students were also required to define pre-planned study times for one week in advance.

For two weeks the professor quickly graded the calendars, which were brought to class. This was to ensure students were following directions; many had errors or deficiencies in defining pre-planned study times only putting a day to study, not a specific time to start and end times. After two weeks, students graded each other's calendar with the professor randomly checking calendars. This started the week the professor discussed Habit 3, putting first things first, and continued the rest of the semester.

Through the ABET-prescribed annual assessment process, the usefulness of this exercise was discussed and will be replicated in the remaining three intermediate to advanced programming courses, giving consistent exposure from the freshman experience through the senior-level capstone project. Anecdotal evidence suggested that some students internalized the exercise and have continued using the calendar with deliverables in their work environment. The details of the calendaring exercise are outlined in Appendix F.

Finding the Right Tools

Ultimately, the green curve is also designed to facilitate acceleration along the blue curve. In this first programming course, several tools have greatly assisted in realizing our goals for the course. A detailed explanation of each tool, and how they are used, is outlined in Appendix G.

5. DISCUSSION

Returning to the research questions presented in this paper, we share the following conclusions from our investigations and reflections.

RQ #1. Besides technical skills, what other professional skills are needed by our students to prepare them for successful careers?

For employers, industry advisory board members, and faculty observations regarding non-technical deficiencies that hinder our students achieving the ABET student outcomes, we conclude that the non-technical skills are critical to include throughout our technical program. Desired skills are time management, prioritization, career planning, goals seeking, taking responsibility for one's actions, collaboration, effectiveness in small teams, etc.

For example, the number one reason for unsatisfactory levels of achievement in ABET student outcomes came from students failing to turn in the assignments. Teaching more technical skills won't resolve this problem. Teaching prioritization and time management has a greater chance of helping these struggling students.

RQ #2. What frameworks and tools might be useful to include in teaching non-technical skills to students?

A valuable framework is the 7 Habits of Highly Effective People by Steven R. Covey, which emphasis being proactive not reactive, beginning with the end in mind (e.g., goal setting and problem decomposition), putting first things first, (e.g., prioritization and time management), think win-win for mutual beneficial solutions, seeking first to understand before being understood (e.g., listening and communication), synergy through teamwork, and sharpening the saw (e.g., continual learning and improvement, especially outside of class and in their professional careers).

RQ #3. Where should these skills be taught? At the beginning, middle, or end of a student's academic career?

In our college of business, students are introduced to the 7 Habits of Highly Effective People in a principle of management course. However, the introduction is brief, focuses on comprehension of the 7 habits rather than the acquisition of the habits, and students typically don't take the course until their junior year, which is too late to benefit from the skills in their critical years as freshman and seniors.

Since our department is ABET accredited, and since the failure to achieve student outcomes was not due to technical knowledge, and since failing to achieve a student outcome can jeopardize our accreditation, we determine it was our responsibility to teach these professional soft skills. Relying on the university or another department to fix the non-technical problems were too risky.

To help communicate to fellow colleagues and administrators our ideas, we created a framework called the green curve and blue curve. (See Figure 1). The green curve represents soft skill such as those from 7 Habits of Highly Effective People. The blue curve represents the technical skills, as outline in our adoption of ABET Student Outcomes and current technical curriculum.

The framework prescribes emphasizing green curve skills early in a student's academic time, even at the expense of deemphasizing some technical criteria to make room in the course schedule. The framework also prescribes that as a student progresses through their academic career, the amount of emphasis on green curve can be reduced since they have demonstrated green curve skill acquisition through rubrics. Because of the increase in these green curve skills, students can have an accelerated blue curve emphases and hopefully learn more technical skills that would have been possible without the acquisition of the green curve skills.

In later years, green curve skills are switched to other skills that may be of benefit, such as job interview skills, creation of a technical portfolio useful in job interviews, acquisition of internships, leadership, and team effectiveness, professional oral presentations, etc.

Earlier in this paper and in the appendices, we are pleased to share the tools, assignment and rubrics that we are experimenting with to achieve the green and blue curve skills.

RQ #4. What results have been observed from experimenting with different educational treatments?

It is too early to describe the full effect of the strategy we've outlined in this paper. Future assessment cycles, prescribed by AACSD and ABET processes, will help elucidate the achievements. However, some early indications are available. Our university has a freshman persistence rate of 63%. Increasing this rate is a top priority of university administrators. Since the inclusion of the green curve assignments into our programming fundamentals course, the persistence rate for two semesters has been 72% (N = 39)and 73% (N = 32). The first semester course was taught in a computer lab on-the-ground. The subsequent course was taught online learning objectives and tools, e.g., Cloud 9, codehs.com, etc.).

Both semesters were taught by the same professor with three student-led lecture days on 7 Habits of Highly Effective People, career and life portfolio assignment (see Appendix C), and a weekly assessment of using a calendaring system to pre-plan study times and identify school deliverables. It is a difficult attribute causality behind the results as there was not a positivist design to the treatments. The results could be attributed to many non-treatment factors, but some portion of the increase could be attributed to emphases on green curve skills. Take the results as indicative of potential success and that each college should evaluate which assignments and rubrics to implement in which courses.

6. CONCLUSION

We understand that all schools may not have the same needs or the dynamics as our regional school. We recommend a technical program evaluate their students for the attainment of green curve skills and reflect on if the acquisition of green curve skills might increase the achievement of technical student outcomes. We present to the reader our rubrics and assignment instructions that may aid in teaching green curve skills.

7.REFERENCES

Babb, J., Browning, L., Womble, L., & Abdullat, A. (2014) Designing a Student Mentor Program For Engagement and Retention in a College Of Business at a Regional Institution.

Proceedings of the 2014 Clute Institute International Academic Conference, March 16 - 18, San Antonio, Texas (USA)

Biggers, M., Brauer, A., & Yilmaz, T. (2008, March). Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers. In *ACM SIGCSE Bulletin* (Vol. 40, No. 1, pp. 402-406). ACM.

Burns, T. J., Gao, Y., Sherman, C., Vengerov, A., & Klein, S. (2014). Investigating a 21st Century Paradox: As the Demand for Technology Jobs Increases Why Are Fewer Students Majoring in Information Systems?*Information Systems Education Journal*, *12*(4), 4-16.

BLS (2014). U.S. Bureau of Labor Statistics. www.bls.gov. Accessed June 30, 2014.

Clark, M. R. (2005). Negotiating the freshman year: Challenges and strategies among first-year college students. *Journal of College Student Development*,*46*(3), 296-316.

Cohoon, J. M. (2002). Recruiting and retaining women in undergraduate computing majors. *ACM SIGCSE Bulletin*, *34*(2), 48-52.

Elliott, T. R., Godshall, F., Shrout, J. R., & Witty, T. E. (1990). Problem-solving appraisal, self-reported study habits, and performance of academically at-risk college students. *Journal of Counseling Psychology*, *37*(2), 203.

Gorgone, J., Davis, G. B., Valacich, J. S., Topi, H., Feinstein, D. L., & Longenecker, H. E. (2003). IS 2002 model curriculum and guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems*, *11*(1), 1.

Granger, M. J., Dick, G., Jacobson, C. M., & Van Slyke, C. (2007). Information Systems Enrollments: Challenges and Strategies. *Journal of Information systems education*, *18*(3), 303-311.

McCambley, W. (2014). Top Ten Computer Science Engineering Schools. *Codehs.com*. June 11.

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, *49*(8), 90-95.

McKenzie, W. B. (2006). Information Systems Curriculum Revision in a Hostile Environment: Declining Interest, Threats from Offshore, and Proprietary Certification. *Information Systems Education Journal*, *4*(105), 3-8.

Mishra, S., Draus, P., Caputo, D., Leone, G., Kohun, F., & Repack, D. (2014). Gender Rationales and Success Factors in Assessing and Selecting a Major in Information Technology at the Undergraduate Level of a University Program: A Focus Group Approach. *Information Systems Education Journal*, *12*(4), 40.

Peckham, J., Stephenson, P., Hervé, J. Y., Hutt, R., & Encarnação, M. (2007). Increasing student retention in computer science through research programs for undergraduates. *ACM SIGCSE Bulletin*, *39*(1), 124-128.

Rendon, L. I. (1995). Facilitating Retention and Transfer for First Generation Students in Community Colleges.

Saulnier, B., Brooks, N., Ceccucci, W., & White, B. A. (2006). Learning Communities in Information Systems Education: Developing the Reflective Practitioner. *Director*, 07.

Sheskin, D. J. (2003). *Handbook of parametric and nonparametric statistical procedures*. crc Press.

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker Jr, J. F., Sipior, J. C., & de Vreede, G. J. (2010). IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems*, *26*(1), 18.

Wise, B. (2008). High schools at the tipping point. *Educational Leadership*,*65*(8), 8.

Zuegel, D. 2012. CodeHS Aims to be Market Trailblazer for Pre-College Computer Science Education. *The Standford Review*. December 4.

Appendix A. List of Life Ambitions exercise

Learning Objective:

1. The student will be able to create a list of life ambitions (e.g., what do you want to learn, acquire, accomplish, places to visit, etc.)

Activity: In class, read or paraphrase to the students the following purpose that motivates the exercise.

Purpose: There is story told about a NASA astronaut who when he was 16 or 17 years old he made a list of all the things he wanted to do and accomplish in his list. The list includes many things, including being an astronaut. He'd check off the items as he accomplished them. Habit 2 of 7 Habits of Highly Effective People is "Begin with the End in mind" Creating a list of life ambitions is designing where you want to be, who you want to become, and what you want to have accomplished. At a later time, you can use this list to select specific goals to work on that are in alignment with your long term goals.

Instructions: Make a list of 20 to 30 life ambitions. Make a list, not a paragraph so you can see each item independently of the other items. The following questions can help prompt ideas. Don't judge the ideas or decide how you will accomplish them. For now, just list them and get excited about them. For several of your ambitions, add photos to motivate and visually remind you of the ambition. Search the internet for photos that may visually describe your ambitions, e.g., if you want to go to the Great Wall in China, find a travel photos and paste it in the document. Even though this is for personal use, be sure to record the URL of the photo and give credit to the owner of the photos. Keep this list with you for years to come and set specific goals to help you accomplish your ambitions.

"Begin with the End In Mind" Steven Covey 7 Habits of Highly Effective People

What are your life ambitions? What do you want to accomplish? Consider what you want to accomplish in 1, 5, 10, 20 years that will make your life meaningful and fulfilled. Use this activity to dream. Do not limit your dreams. Do not consider how or when it will be done. Write everything you desire to do, learn, and accomplish. Consider the following questions and write, write, and write.

1. If money and time were not obstacles, where would you want to visit, go to, see, explore, etc.?
2. What things do you want to learn? (Languages, skills, sports, hobbies, etc.)
3. What things do you want to earn, buy, acquire, donate to, etc.?
4. What do you want to participate in, accomplish, win, race, experience, etc.?

Grading Rubric

Criteria	**Highly Ambitions**	**Low Ambition**
Beginning with the End in Mind	□ 20 to 30 life ambitions have been listed □ The four questions have been answered with some life ambitions	□ Fewer than 10 ambitions □ At least one question does not have ambitions listed
Visual Appeal	□ Ambitions are in list form not in a paragraph □ At least 5 photos were added corresponding to ambitions □ Credit was given to the intellectual property owner of the photos identifying the URL of the photo and, if possible, the owner's name.	□ Wrote a paragraph rather than a list □ Fewer than 5 photos were added (demonstrating lack of attention to details) □ Credit was not given to the intellectual property owner

Appendix B. Career Interview and Reflection on Career Preparedness

This is a multi-part assignment to cause you to reflect on what you want to do as a career and how best to prepare for it.

Part A:

Find online3 job titles *in your chosen disciplines* and range of salaries for those jobs. Consider using indeed.com or other websites. What qualifications are needed? Are any technology qualification mentioned? What is the salary? Include this data in a 4-column table when you write your reflection for Part C.

Job Title	Qualifications Required	Technical Qualifications Required	**Salary (if available)**
1.			
2.			
3.			

Part B:

Find someone in your discipline and interview them, asking questions similar to these but not limited to these.

1. What do they do? What are their responsibilities or tasks? What is a typical day like for them?
2. What technologies/hardware/software do they use?
3. What would technology or software would they like to know that would make them better as a professional?
4. Any other question that would help you understand their job and the desirability of it to you?

Part C:

Write summary of the information you found through Parts A, B and C. Maximum of two pages (spacing doesn't matter), and a minimum of two paragraphs. Grading will be on evidence of your effort and reflection of what you found. This is your life. You'll get out of it what you put into it.

Deliverable:

Deliver the MS Word document or PDF document to WTclass…\Lessons\Turn In Homework\Homework Career Interview\.

Questions and Answers:

What format should the info be in table or paragraphs?

Several of you have asked if the format of the paper should be in tables or written paragraphs. The format isn't as important to me as the synthesis and reflection on the data you gathered. Some parts lend themselves to present in a table; other parts lend themselves to writing. You choose how you want to present it.

Part A can be in table format or paragraph format. I'm also looking for a synthesis of the data you gathered. The instructions asked some thought provoking questions regarding what you found out, e.g., Can you apply for the job yet? What general skills and what technical knowledge do you need to learn in the next year to be able to apply? Are there any certificates or licensing requirements? How are you going to get experience is they say "8 years experience required"? etc. The Part B (interview) lends itself to paragraph writing also as it is a synthesis of the interview.

Writing is a process of thinking: gathering data, reflecting on the personal meaning of the data you gathered, interpret the knowledge in the data, and then communicating that new knowledge. I want this to be a valuable learning activity for you that will change your behavior in the next year or two before you graduate---so that you will be stupendously prepared for your careers.

Can we work together on the assignment?

Yes and no. You may do Part A and B with a partner. I encourage collaboration and discussion. But Part C must be written by yourself.

Grading Rubric

Criteria	Excellent	Satisfactory	Unsatisfactory
Part A Job Opps	□ 3 job opportunities are identified with qualification requirements	□ 1 jobs identified	0 jobs identified
Part B Interview	□ A professional was interviewed Interview consisted of the following: □ Questions relating to job responsibilities, typical day, what they do, etc. were asked. □ Questions related to the technical or software requirements		Failed to ask questions regarding job responsibilities Failed to ask questions regarding technical requirements
Part C Written Reflection	□ Evidenced provided for *reflecting* on Parts A □ Evidenced provided for *reflecting* on Part B	□ Completed Part A but did not reflect or apply to themselves □ Completed Part B but did not reflect or apply to themselves	Missing evidence for Part A Missing evidence for Part B
Mechanics	□ Length is sufficient to demonstrate reflection and application to the student's life □ Length is no more than two pages □ Free from distracting grammatical errors, syntax errors, or rambling.	□ Too short to demonstrate reflection and application. Leaves more questions than it answers. □ Length is longer than 2 pages □ More than two grammatical errors are distracting.	Appears as if the student vomited thoughts onto the paper with no reflection or coherent structure.
Total Points			

Appendix C. Career and Life Portfolio

Task

Demonstrate that you have improved your personal life skills and learned something from this course besides programming (i.e., the Seven Habits of Highly Effective People). Create a physical portfolio that demonstrates the personal leadership skills you possess or have learned. Create a section (Chapter divider) for the learning objectives listed below. Be creative in presentation! There is no limit to what you can do to present your portfolio, other than it must be tangible and carry-able.

Not all solutions to business problems require technical solutions. The following non-technical learning objectives will be explored in class. Create a tangible portfolio that includes evidence of completing each learning objective listed below.

Learning Objectives:

Students will be able to….

1. Communicate a clear vision of life and career opportunities, including short and long-term goals. (Habit 1: Be proactive and Habit 2 Begin with the end in mind.)
   a. List of Life ambitions, including photos of many of your life ambitions
   b. At least three specific goals, using the template patterned off the goal-setting worksheet Seven Habits of Highly Effective People provided by Dr. Humpherys
   c. Add a page on quotes and life maxims that are motivating to you or significant to you and help you succeed.
2. Outline a specific career plan through college that explores desired career options that leads to students fixing a specific career outcome
   a. Create a list of classes you can take for your college major for each semester until you are graduated. This is a plan (Habit 2: Begin with the end in mind). It is not set in stone and may change in the future. But demonstrate you have a plan, fill in classes, even electives or possible electives, for every semester until you graduate.
3. Demonstrate time management skills and good study habits -- replacing ineffective habits with desired and effective habits
   a. Provide a copy of your calendar with pre-defined study times, school deliverables (homework and exam due dates), and pre-planned fun time.
4. Demonstrate an increased usage of the Student Success Center and other on-campus resources
   a. Provide evidence that you have used one of the resources of the Student Success Center (career counseling, job & internship opportunities, mock interviews, writing center, math lab, physics lab, resume workshop, resume consultation, academic advising, etc.). Evidence can be a summary of something you attended or a photo of yourself in the success center activity, or a signature by a councilor or tutor on something, etc.
   b. Provide evidence that you have integrated yourself into WT college life. For example, take photos of yourself at the activity center climbing the rock wall, bowling, or at some campus event, concert, fine arts event, classroom or labs you frequent, attend a club, attend a sorority or fraternity event, etc. College life is more than just studies, it's a cultural adventure. Instead of photos you can also write about how you've integrated yourself into college life. If someone were to ask you, "So what are you doing at college?" you could show them this chapter and give evidence of all the exciting fun things you are doing and learning in college.

5. Samples of your deliverables demonstrating your website and programming skills. Include at least 3 samples of technical project or technical creations you have done in or as a result of this class. Consider including your homework (particularly homework 4 website and homework 5 Javascript application) or learning activities.

Deliverable
Bring your tangible portfolio to class on the day specified on the schedule.

Grading Rubric

Criteria	**Excellent**	**Satisfactory**	**Unsatisfactory**
1. Communicate a clear vision of life and career opportunities, including short and long-term goals. (Habit 1: Be proactive and Habit 2 Begin with the end in mind.)			
List of Life ambitions, including photos of many of your life ambitions	□ A prospective employer would be impressed with the quantity and thoughtfulness that went into listing your life ambitions and would say “This is someone who knows what they want in life” □ Several images are provided as visual reminders of their ambitions.	□ A teacher would say, “They put in enough effort to satisfy the assignment but not enough to make it really meaningful and useful to their future.” □ Two or fewer images were included.	□ No ambitions listed
At least three specific goals, using the template patterned off the goal-setting worksheet Seven Habits of Highly Effective People provided by Dr. Humpherys	□ One goals set □ Goal is stated in the positive and present tense □ Template was used and filled out □ Goal is measurable □ Goal has a due date □ He/she has listed compelling reasons for the goal using the template	□ Goal is stated in future tense not present tense □ Goal is too broad, not measurable □ Did not use the template provided and therefore is missing important elements of goal setting	□ No goal set
Add a page on quotes and life maxims that are motivating to you or significant to you and help you succeed.	□ One page of positive quotes and life maxims provided □ Visually appealing	□ Half page of content	□ No pages
2. Outline a specific career plan through college that explores desired career options that leads to students fixing a specific career outcome			
Graduation plan	□ Create a list of classes you can take for your college major for each semester until you are graduated (e.g., executable plan) □ Elective are specific and	□ Courses listed but not by semester (not a plan, just a checklist) □ Electives are	Not present

	not ambiguous	ambiguous □ Incomplete timeline	
3. Samples of your deliverables demonstrating your website and programming skills.			
Include at least 3 samples of technical project or technical creations you have done in or as a result of this class.	□ Three samples provided	□ Less than three provided	None present
4. Portfolio Organization			
Chapters	□ The chapters are easily distinguishable	□ Hard to find items	Items are or disorganized
Grammar & Style	□ Free from distracting grammatical and typographical errors	□ A few grammatical errors present	Disregard for quality of grammar
Effort and Proactive	□ Evidence of thoughtfulness and proactive effort put into the assignment to make the career portfolio a useful life-planning tool □ Has visual appeal Is of a quality that demonstrates time and effort and could be presented to a prospective employer or scholarship evaluation	□ Some effort demonstrates □ Evident this was created by a student and not presentable to anyone else □ Appears to go through the motion to check off the required boxes but not to make if a purposeful life planning tool.	Demonstrates lack of effort beyond satisfying minimum requirements

Appendix D. The Student Outcomes for CIS Program

These Student Outcomes (SO) were adopted by our department from ABET student outcomes, with the addition of SO#8 and SO#9.

SO1. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
SO 2. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
SO 3. An ability to use current techniques, skills, and tools necessary for computing practice
SO 4. An ability to apply knowledge of computing and mathematics appropriate to the discipline
SO 5. An understanding of processes that support the delivery and management of IS within a specific application environment
SO 6. An ability to analyze the local and global impact of computing on individuals, organizations, and society
SO 7. Recognition of the need for and an ability to engage in continuing professional development
SO 8. An ability to function effectively on teams to accomplish a common goal
SO 9. An ability to communicate effectively with a range of audiences
SO 10. An understanding of professional, ethical, legal, security and social issues and responsibilities

Appendix E. Specific Green Curve Learning Outcomes for the First Programming Course

1. Communicate a clear vision of life and career goals, including short and long-term goals. (Habit 1: Be proactive and Habit 2 Begin with the end in mind.)
    a. Deliver brief oral presentations to fellow students regarding Habits 1, 2, 3, or 7.
    b. Discuss with fellow students what they learned from the readings on Habits 1, 2, 3, and 7.
    c. Write reflective answers to a set of questions from Habits 1, 2, 3, and 7 in the personal workbook
    d. Create a list of life ambitions (e.g., what do you want to learn, acquire, accomplish, places to visit, etc.) (see Appendix A for instructions and rubric)
    e. Set three detailed goals (using a modified template from the personal workbook)
    f. Practice achieving short-term goals (using a 5-day Tiny Habits program developed by Dr. BJ Fogg. http://tinyhabits.com)
    g. Discuss with an IT/IS professional about his/her career and how to prepare for it (see Appendix B)
    h. Watch code.org videos on why everyone should learn coding and what a coding career looks like
    i. Tour a local data center and ask two questions to the host Chief Information Officer
2. Outline a specific academic plan through college that explores desired career options that leads to students fixing a specific career outcome with semester by semester, actionable plan for courses to take.
3. Demonstrate time management skills and good study habits providing a copy of a personal calendar with pre-defined study times, school deliverables (homework and exam due dates), and pre-planned fun time.
4. Demonstrate an increased usage of the Student Success Center and other on-campus resources.
5. Provide evidence of integration into college life and campus or learning community activities.
6. Create a portfolio of at least three technical projects that can be used for an entry-level job interview or internship interview.

Appendix F. Details of the Calendaring Assignment

The following instructions accompanied the calendaring assignment:

> “Many studies and experience have demonstrated that when you pre-plan a goal, it gets accomplished. When you don't pre-plan, you lose your way. You will use your calendar to demonstrate and practice time management and project management skills. Put the following on your calendar.
>
> 1. When will you study next week? Specifically, what days and what times? This is a promise you make to yourself, not to me.
>
> Poor Answer: "I'll study on Tuesday and Thursday"---Poor because it is not specific enough and distractions will stop you from succeeding.
>
> Poor Answer: ‘I don't know because my schedule changes each week.’ ---That's the exact reason to pre-plan! You can change it as needed but pre-plan.
>
> Great Answer: ‘I will study on Monday at 9pm-10pm; Wed from 8am-10am; Thursday from 8am-10am; Saturday from 8pm-10pm.’
>
> The amount of time you study is determined by your goals. A student desiring an A will put more time than one striving for a C. That you pre-plan specific times is what will be graded, not the amount of time.
>
> 2. On the appropriate due date, put all the deliverables (exams, quizzes, assignments, etc.) from each of your courses, not just this course. Do this for at least one month in advance. You do not need to put your class times, as that just clutters your calendar which should focus on helping you deliver your deliverables on time and earn a stellar grade."

Appendix G. A Description of Helpful Tools

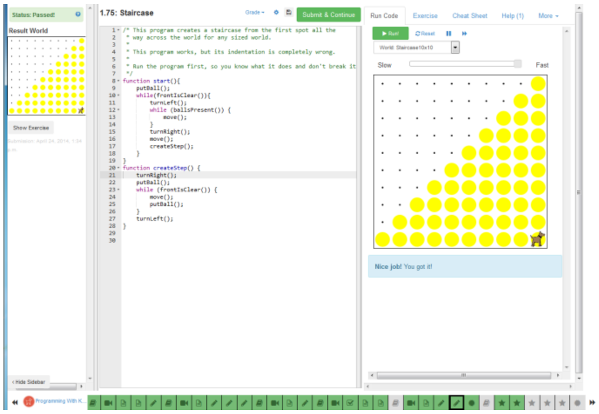
Codehs.com

Codehs.com is an online, cross-platform learning system to learn JavaScript and programming fundamentals which provides instant feedback to the student on performance.

Currently codehs.com offers a free module called “Programming with Karel” that covers basic programming concepts such as creating and calling functions, linear execution of code, problem decomposition, recursion, and control structures. The website provides 2-5 minute videos, followed by an exploration of pre-written code, followed by one or two exercises to apply the specific learning objective in a programming challenge. Furthermore, an automated assessment agent provides instant feedback to the student regarding the failure or success of the challenge. The benefit of the automated agent is that the course professor is no longer the bottleneck to students learning using this supplemental resource. For small and short lessons provided by codehs.com, instructor feedback may take a week, with the delay limiting the helpfulness of the feedback. However, the automated agent catches most errors and provides students the opportunity to learn from failure and correct the code. Figure 6 demonstrates a programming challenge and the IDE students use to program; students learn at their own pace. Class attendance is required and 2.5 weeks are dedicated to using codehs.com to learn programming fundamentals. Outside work is usually required for students to complete the exercises.

Teacher tools are also provided by codehs.com and we could monitor the progress, view code simultaneously, and answer questions online. Exercises are graded by the professor as pass, fundamentally passed (i.e., accomplish the challenge but can be improved upon), or fail. Codehs.com assigns points for each passed exercise. For the more complicated exercises that may have multiple solutions, the automatic agent can provide feedback, but a human (professor or TA) must make the final assessment of pass, fundamentally passed, or fail. Solutions and common feedback answers are provided to the teacher. Grading is very quick and not a burden, especially since the teacher replaces lecturing with providing feedback and grading of advance exercises. Course grades for the assignment are simply the percentage of earned points in codehs.com over total points available by Codehs.com.

After codehs.com the course topics resumed with integrating JavaScript into web pages and getting input from a web user and dynamically outputting results to the web page. According to codehs.com, Stanford has used Codehs.com in their computer science introductory course work (Zuegel, 2012).



1.75: Staircase

```
/* This program creates a staircase from the first spot all the
 * way across the world for any sized world.
 *
 * This program works, but its indentation is completely wrong.
 *
 * Run the program first, so you know what it does and don't break it
 */
function start(){
    putBall();
    while(frontIsClear()){
        turnLeft();
        while (ballsPresent()) {
            move();
        }
        turnRight();
        move();
        createStep();
    }
}
function createStep() {
    turnRight();
    putBall();
    while (frontIsClear()) {
        move();
        putBall();
    }
    turnLeft();
}
```

Status: Passed!

World: Staircase10x10

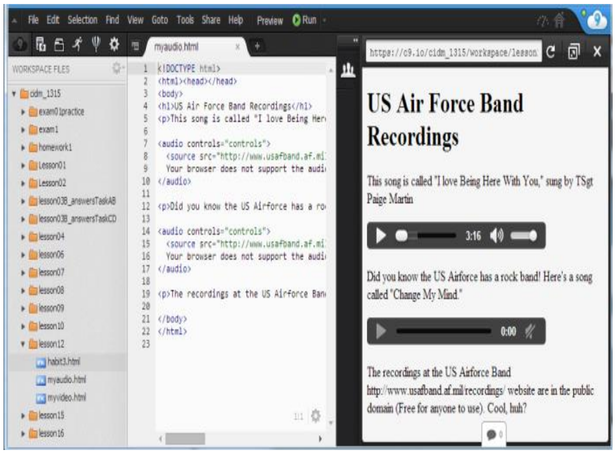
Nice job! You got it!

Submission: April 24, 2014, 1:34 p.m.

Figure 6. Codehs.com IDE for learning fundamental control structures.

Through course evaluations and informal evaluations students have responded very positively to the inclusion of in-class learning activities and the use of codehs.com, definitely preferring the hands-on exercises over hour-long lectures. Because of the inclusion of daily learning activities and use of codehs.com, the number of coding exercises performed by students increased by 600%, excluding homework assignments which also increased over the ABET assessment cycles. While it is hard to attribute student outcomes to any one factor, student outcomes did correspondingly improve with the introduction of daily learning activities and codehs.com. Plus the cognitive level of skills increased from simple memorization and syntax error finding in multiple choice exams to producing code that accomplish a mission.

Cloud 9 IDE for web development in the cloud
Cloud9 IDE (http://c9.io) is a professional online development environment for JavaScript and Node.js applications. Students can use the free services for the daily learning activities allowing them to code web pages and JavaScript applications. The advantages over coding web pages with notepad are that the students can resume their work on any computer or at home, web pages are published on the Internet, the IDE gives some feedback to common syntax errors, the IDE file system mirrors a web server file structure, there is an instant preview window, and the professor can collaborate in real-time through the IDE regardless of the location of the students (i.e., online students; see Figure 7). Another advantage of this tool is platform-independence that facilitated Mac users as well as Windows users. The course previously used Notepad++, which was not available for Mac users. Our experience was that the Google Chrome and Mozilla Firefox browsers best supported Cloud9. One downside to the free services is that the workspace environment was occasionally unresponsive because Cloud9 gives server resource priority to paid accounts.



```
<!DOCTYPE html>
<html><head></head>
<body>
<h1>US Air Force Band Recordings</h1>
<p>This song is called "I love Being Her

<audio controls="controls">
  <source src="http://www.usafband.af.mi
  Your browser does not support the audi
</audio>

<p>Did you know the US Airforce has a ro

<audio controls="controls">
  <source src="http://www.usafband.af.mi
  Your browser does not support the audi
</audio>

<p>The recordings at the US Airforce Ban

</body>
</html>
```

US Air Force Band Recordings

This song is called "I love Being Here With You," sung by TSgt Paige Martin

Did you know the US Airforce has a rock band! Here's a song called "Change My Mind."

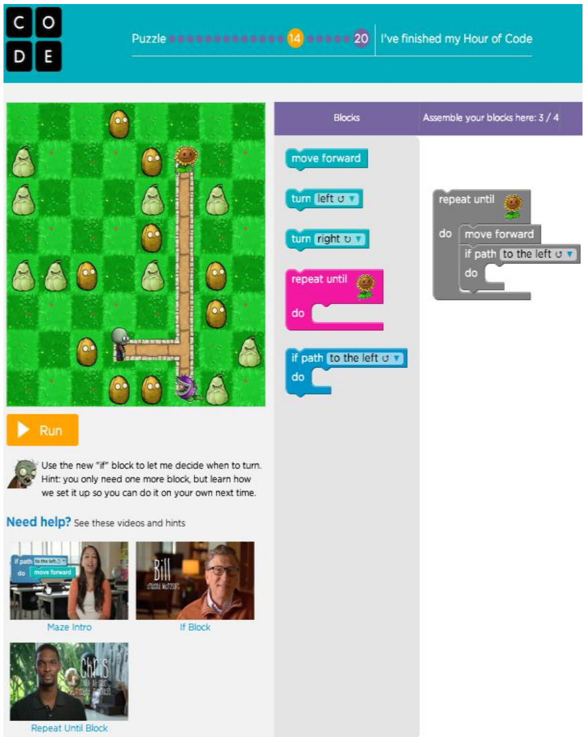
The recordings at the US Airforce Band http://www.usafband.af.mil/recordings/ website are in the public domain (Free for anyone to use). Cool, huh?

Figure 7. Cloud9 IDE and a learning activity regarding HTML5 audio tag

Hour of Code
We have three business courses that participate in Hour of Code: programming fundamentals, management of information systems, digital collaboration and communications. The purpose is to expose non-majors to the world of coding. Students submit a certificate of completion as evidence for their assignment. Experienced coders can submit something learned from one of the mobile app development tutorials, as very few have mobile app development experience, or participate in an hour of self-directed learning. The purpose is to provide highly motivating positive experiences of non-majors or beginning programmers while introducing them to videos of successful professional coders from Microsoft, Dropbox, Facebook, etc.

In addition we host approximately 700-1000 middle school and high school students at the College of Business throughout the semester and use Angry Birds Hour of Code (code.org) to provide younger students hands-on coding experience. Each visiting student sits at a computer for approximately 30 minute session and codes in the Angry Birds Plants versus Zombie tutorial.

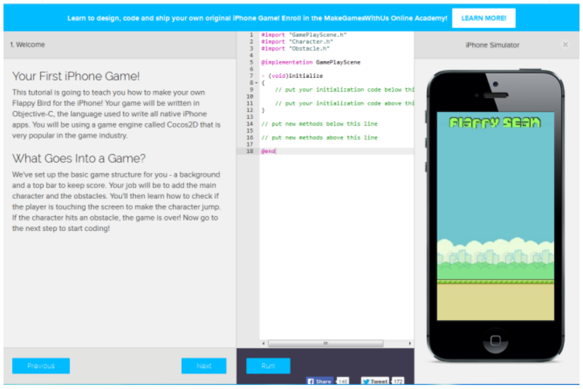
Code.org: http://csedweek.org/learn. Code.org provides several hours-long tutorials and resources for any beginner to learn to code while having fun. They are one of the major contributors to Hour of Code, which introduces millions of students to coding in a one hour session in any course. Code.org features Angry Bird & Plants versus Zombie tutorial for beginners http://learn.code.org/hoc/1 which teaches procedural programming, if else statements, while loops, and problem decomposition in an hour. It features motivational videos of well-known coders from Microsoft, Dropbox, Facebook, etc. Several other hour-long tutorials are available. Learning objectives are problem decomposition, procedural programming, decision-making control structures, and recursion control structures. Students are not required to type code, but instead drag blocks of predefined code that snap together to create procedures. The block has properties that are customizable. The block-coding technique allow students to focus on the creation and control structures without worrying about syntax errors. While apparently targeted to younger students, even our adult students enjoy this activity as many have never coded before and the challenges get progressively more difficult even for a novice adult coder.



Use the new "if" block to let me decide when to turn. Hint: you only need one more block, but learn how we set it up so you can do it on your own next time.

Figure 8. Code.org's Angry Bird and Plant's versus Zombie coding tutorial for Hour of Code participation teaching basic control structures without worrying about syntax errors.

MakeGamesWithUs:https://www.makegameswith.us/build-an-ios-game-in-your-browser/. Students learn to make a complicated game in a free online simulated iPhone. Students type their code in a browser based IDE following tutorial prompts. Knowledge of variables, methods and objects is beneficial.


Learn to design, code and ship your own original iPhone Game! Enroll in the MakeGamesWithUs Online Academy!
LEARN MORE!
1. Welcome
iPhone Simulator
Flappy Sean
Previous
Next
Run
Share
Tweet

Your First iPhone Game!

This tutorial is going to teach you how to make your own Flappy Bird for the iPhone! Your game will be written in Objective-C, the language used to write all native iPhone apps. You will be using a game engine called Cocos2D that is very popular in the game industry.

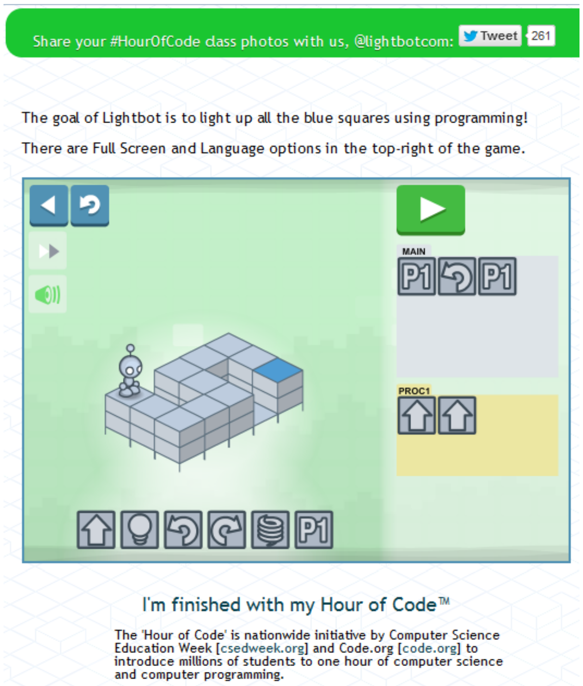
What Goes Into a Game?

We've set up the basic game structure for you - a background and a top bar to keep score. Your job will be to add the main character and the obstacles. You'll then learn how to check if the player is touching the screen to make the character jump. If the character hits an obstacle, the game is over! Now go to the next step to start coding!

```
#import "GamePlayScene.h"
#import "Character.h"
#import "Obstacle.h"

@implementation GamePlayScene

- (void)initialize
{
    // put your initialization code below thi

    // put your initialization code above thi
}

// put new methods below this line

// put new methods above this line

@end
```

Figure 9. MakeGamesWithUS.us is an Hour of Code option that let students create an iPhone game in a browser.

LightBot Hour of Code: http://light-bot.com/hoc.html
This is a free app for both iPhone and Android devices or playable in a Flash-enables browser. The application teaches procedural logic, recursion, problem decomposition, and use of functions through a hedonic experience (see Figure 9). Students must command a robot through a series of mazes to turn on the lights using predefined procedural options such as move forward, jump, turn left or turn right, and turn on the light. Students report the addictive nature of this app and not wanting to stop. Because of its hedonic nature, this app can be used as an out-of-class assignment to practice procedural programming.


Share your #HourOfCode class photos with us, @lightbotcom:

The goal of Lightbot is to light up all the blue squares using programming!

There are Full Screen and Language options in the top-right of the game.

I'm finished with my Hour of Code™

The 'Hour of Code' is nationwide initiative by Computer Science Education Week [csedweek.org] and Code.org [code.org] to introduce millions of students to one hour of computer science and computer programming.

Figure 10. Lightbot Hour of Code is an addictive game to help students experience coding or practice coding to solve logical problem.