Integrating Computational Creativity Exercises into Classes

Leen-Kiat Soh
University of Nebraska - Lincoln, lsoh2@unl.edu

Duane F. Shell
University of Nebraska - Lincoln, dshell2@unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/dberspeakers
Part of the Higher Education Commons, and the Science and Mathematics Education Commons
Integrating Computational Creativity Exercises into Classes

Leen-Kiat Soh¹ and Duane F. Shell²

¹ Department of Computer Science and Engineering
² Department of Educational Psychology

University of Nebraska, Lincoln, NE

* Special thanks to Dr. Liz Ingraham of Department of Arts and Art History
OVERVIEW

Introduction | Creative Thinking, Computational Thinking
Exercises | Overall Design & Examples
Results | Brief Overview
Logistics | Tips, Support & Feedback
INTRODUCTION

Aim to improve the learning of computational thinking by blending it with creative thinking

Creative thinking?

• Patterned in a way that tends to lead to creative results
• Not limited to the arts
• An integral component of human intelligence that can be practiced, encouraged and developed within any context
INTRODUCTION

CREATIVITY THINKING

Epstein’s Generativity Theory breaks creative thinking down to four core competencies

- *Capturing* novelty
- *Challenging* established thinking and behavior patterns
- *Broadening* one’s knowledge beyond one’s discipline
- *Surrounding* oneself with new social and environmental stimuli
Practicing these four creative thinking competencies increases our ability to be creative in our discipline (and outside it). You may be practicing these already:

• When you keep a journal or process log, or write down your thoughts, ideas or dreams, or archive an image that inspires you, you are *capturing*.

• When you are given—or give yourself—a problem with a lot of constraints, or an open-ended problem with a nearly infinite number of solutions, or you try something that probably won’t work just to try it, you are *challenging*.

• When you learn a new skill or try a new technique or use a new tool or technology, or go to a lecture outside your field, you are *broadening*.

• When you go for a walk, visit an exhibition, meet new people, read about or talk to someone working outside your field, try a new food, experience a different culture, experiment with color or sound, or change your work environment, you are *surrounding*.
INTRODUCTION

COMPUTATIONAL THINKING

A way of thinking for *logically* and *methodically* solving problems

• E.g., *purposeful, describable, replicable*

Includes *skills* such as

• Decomposition
• Pattern Recognition
• Abstraction
• Generalization
• Algorithm Design
• Evaluation
Breaking down a process into a set of smaller sub-processes to allow us to describe, understand, or execute the process better

- Dividing a task into a sequence of subtasks
- Identifying elements or parts of a complex system

Examples

- When we taste an unfamiliar dish and identify several ingredients based on the flavor, we are decomposing that dish into its individual ingredients.
- When we give someone directions to our house, we are decomposing the process of getting from one place to another.
- When we break a course project into several steps, we are decomposing the task into smaller, more manageable subtasks.
- In mathematics, we can decompose a number such as 256.37 as follows: $2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 3 \times 10^{-1} + 7 \times 10^{-2}$
Noticing or identifying similarities or common differences that will help us make predictions or lead us to shortcuts

- We look for *patterns* when we play games to decide when to do certain things
- Based on experience, we develop shortcuts mapping problem characteristics to solution

**Examples**

- We look for patterns when choosing a registrar when we checkout
- Drivers look for patterns in traffic to decide whether/when to switch lanes
- Investors look for patterns in stock prices to decide when to buy and sell
- Scientists look for patterns in data to derive theories and models
Preserving information that is relevant in a context, and forgetting or suppressing information that is irrelevant in that context to solve a problem

- We use abstraction to organize things:
  - A human is a mammal, a mammal is an animal, and so on
  - A “big picture” so we can reason without thinking about the details

Examples

- A world map is an abstraction of the earth in terms of longitude and latitude, helping us describe the location and geography of a place
- A sign of an aisle in a store—e.g., Walmart—is an abstraction of the items available in that aisle
- When we write a book report, we summarize and discuss only the theme or key aspects of the book, it is abstraction
- When we tell a story or describe a movie to our friends, why don’t we describe every single detail of the story or movie?
Identifying common or shared characteristics between two domains or problems such that models or solutions of one could be adapted or applied to the other

• Group project A successful because of good teamwork strategy. Apply same good teamwork strategy to group project B should work too.
• Deals with trends, norms, outliers, scalability

Examples
• A customer database designer uses *generalization* when deciding what information about a customer are important to keep
• Google search identifies popular keywords in different regions at different times and suggests those keywords (in autocomplete and also correction) using a generalization-like process
• Amazon.com and Netflix model and categorize their customers, use generalization—*inferencing*—to predict what their customers are interested in, and make recommendations accordingly
Developing a step-by-step strategy for solving a problem

• An algorithm is a sequence of steps that solves a problem
  • Input → output
  • Effective
• Algorithmic thinking involves both *creation* and *execution* of an algorithm

Examples

• When a cook writes a recipe for a dish, he or she is creating an algorithm that others can follow to replicate the dish
• When your friend writes down the instruction to get to her house, she is specifying a sequence of steps—that is, an algorithm—for you to follow (See Google maps!)
• When a teacher gives a set of instructions to carry out an experiment, he or she is specifying an algorithm for you to follow to collect and analyze data
THE FRIENDSHIP ALGORITHM

DR. SHELDON COOPER, PH.D

PLACE PHONE CALL

HOME?

YES

"WOULD YOU LIKE TO SHARE A MEAL?"

LEAVE MESSAGE

NO

WHAT IS THE RESPONSE?

NO

WAIT FOR CALLBACK

YES

DINE TOGETHER

"DO YOU ENJOY A HOT BEVERAGE?"

RECREATIONAL ACTIVITY? TELL ME ONE OF YOUR INTERESTS?"

NO

Do I share that interest?

NO

"WHY DON'T WE DO THAT TOGETHER?"

CASE:

TEA

HAVE TEA

COFFEE

HAVE COFFEE

COCOA

HAVE COCOA

BEGIN FRIENDSHIP!

PARTAKE IN INTEREST

N = 0

N > 6?

CHOSE LEAST OBJECTIONABLE INTEREST

N = N + 1

TM & © Warner Bros. Entertainment Inc.

14-15
Checking to see whether a solution is good

- Fit for purpose (meeting requirements and objectives, etc.)
- Functional correctness (effectiveness, etc.)
- Performance (efficiency, speed, complexity, etc.)

Examples

- When we cook, we taste our dishes and then adjust flavoring accordingly
- When we fold a paper airplane, we test its flight, and then revise either the design or the “execution” to make it fly better
- When we jog or bike, we keep track of our breathing, joints, etc., and decide whether to stop, go slower, or go faster
- When we carry out a physics experiment, say, to find the relationship between temperature and pressure, we check our data, investigate why it does not match the theory, redo our experimental setup, and recollect data points …
<table>
<thead>
<tr>
<th>How the customer explained it</th>
<th>How the Project Leader understood it</th>
<th>How the Analyst designed it</th>
<th>How the Programmer wrote it</th>
<th>How the Business Consultant described it</th>
</tr>
</thead>
<tbody>
<tr>
<td>How the project was documented</td>
<td>What operations installed</td>
<td>How the customer was billed</td>
<td>How it was supported</td>
<td>What the customer really needed</td>
</tr>
</tbody>
</table>
INTRODUCTION | PREMISE

Our premise is

By blending computational and creative thinking students can leverage their creative thinking skills to “unlock” their understanding of computational thinking

- Should be able to make computational thinking more generally applicable to STEM and non-STEM disciplines
- Should make computer science more attractive to non-STEM and under-represented students
- Should help prepare students to compete in a workplace that required interdisciplinary collaboration and expects innovation
### EXERCISES | DESIGN PRINCIPLES

#### BALANCING OF ATTRIBUTES

<table>
<thead>
<tr>
<th><strong>Computational Thinking</strong></th>
<th><strong>Creative Thinking</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergent thinking</td>
<td>Surrounding with new social and environmental stimuli; divergent thinking</td>
</tr>
<tr>
<td>Methodical</td>
<td>Capturing novelty and spontaneous outputs</td>
</tr>
<tr>
<td>Linear and sequential “flow”</td>
<td>Challenging established solutions and algorithms with jumps and from different directions</td>
</tr>
<tr>
<td>Rational &amp; logical processes</td>
<td>Broadening possible solutions through additional, alternative paradigms</td>
</tr>
<tr>
<td>Name</td>
<td>Brief Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Everyday Object</td>
<td>Identify an “everyday” object (such as nail clipper, a paper clip, a Scotch tape) and describe the object in terms of its inputs and outputs and functionalities (Deployed &amp; Evaluated) (K-12 version: Describing an Everyday Object)</td>
</tr>
<tr>
<td>Cipher</td>
<td>Devise a three-step encoding scheme to transfer the alphabet letters into digits and encode questions for other teams to compete to decode (Deployed &amp; Evaluated) (K-12 version: Ciphering a Sentence)</td>
</tr>
<tr>
<td>Storytelling</td>
<td>Each team member develops a chapter (about 100 words) independently in week 1 and team members work to resolve all conflicts or inconsistencies in week 2 (Deployed &amp; Evaluated) (K-12 version: Writing a Story)</td>
</tr>
<tr>
<td>Exploring</td>
<td>Explore sensory stimuli at a particular site (sounds, sights, smell, etc.) and document observations (Deployed &amp; Evaluated) (K-12 version: Exploring your Environment)</td>
</tr>
<tr>
<td>Simile</td>
<td>Poses “simile” descriptions and participate in team-to-team Q&amp;As to solicit guesses and descriptions relevant to a particular object (Deployed &amp; Evaluated) (K-12 version: Guessing Game)</td>
</tr>
<tr>
<td>Machine Testing</td>
<td>Devise ways to test a black-box mysterious machine without causing harm to humans while attempting to reveal the functionalities of the machine (Developed) (K-12 version: Machine Testing)</td>
</tr>
<tr>
<td>Calendar</td>
<td>Build a calendar for a planet with two suns, four different cultural groups with different resource constraints and industrial needs (Developed)</td>
</tr>
<tr>
<td>Path Finding I</td>
<td>Create a step-by-step instruction on drawing lines to create a quilt pattern on a $n \times n$ grid and identify similar structures in other teams’ quilt patterns (Developed)</td>
</tr>
<tr>
<td>Path Finding II</td>
<td>Use rotation, reflection, and loop to generate a more complex quilt pattern based on simpler base pattern (Developed)</td>
</tr>
<tr>
<td>Marble Maze I</td>
<td>Each team member creates a sub-structure allowing a marble to travel at least for $n$ seconds in week 1 and the team puts all sub-structures together to make a super-structure in week 2 (Developed)</td>
</tr>
<tr>
<td>Marble Maze II</td>
<td>Teams are broken up and now must adapt their own sub-structure to work with other sub-structures in their new teams (Developed)</td>
</tr>
</tbody>
</table>
**Objectives:** Computational and Creative

**Tasks:** 2-3 weeks

**CS Lightbulbs**

**Questions:** Analysis and Reflections

**Exercise 0: Introduction**

- Learning how to log in to the Written Agora site [http://agora.unl.edu/](http://agora.unl.edu/)
- Learning how to create a wiki page for the group
- Learning how to add and revise content on the wiki page
- Learning how to use the group forum
**EXERCISES | EVERYDAY OBJECT**

- **Description:** For the next two weeks, you will be using language to try to clearly and thoroughly describe the functions of an ordinary object that you might use every day. You will be acting like the inventor of that object, imagining that it does not yet exist and trying to describe what need would be fulfilled by your (new) object and how (specifically) it will function. … Your challenge is to imagine that this object does not exist and to describe in written language (1) the mechanical function of your object, (2) what need is fulfilled by this object, and (3) the physical attributes and characteristics of your object. … You must describe the object’s function, the need it will fulfill and its attributes in clear, non-technical language that any user could understand. Your description must be specific enough so that someone who had never seen the object could recognize it and understand how it works and understand what benefits it provides.

- **Tasks:** Week 1: Writing the description; Week 2: Answering analysis and reflection questions

- **CS Lightbulbs:** Algorithms, Functions, Modularity

- **Tips:** Use all of your senses; think concretely; pay attention to details which you might normally overlook.
EXERCISES | STORYTELLING

Description: For the next two weeks, your group will be telling a story with several short chapters. However, you won’t write the story as a linear line from A to B. Rather, you will be given a series of story points. These story points will be the pivotal moments in the story. They will act as inputs and outputs to your individual chapters. … Each person in your group will write a chapter of the story that falls between two story points. You will all write your chapters independently of each other. … Each chapter must advance the story by *starting from the previous story point and ending at the next story point*. As an example, the first chapter should *start* with Story Point 1 and *end* with Story Point 2. Chapter 2 should start with Story Point 2 and end with Story Point 3, and so on. … Each chapter must be between 100 and 200 words.

- **Tasks**: Week 1: Writing each chapter individually; Week 2: Resolving all inconsistencies and answering analysis and reflection questions

- **CS Lightbulbs**: Functions, Modularity, Debugging

- **Tips**: Remember each story point must END with the next story point; any missing chapters in week 2 must be connected by the remaining members
**EXERCISES | CIPHER**

**Description:** For the next two weeks, you will be developing a cipher for the alphabet. What is a cipher exactly? (Check out [http://en.wikipedia.org/wiki/Cipher](http://en.wikipedia.org/wiki/Cipher) and [http://en.wikipedia.org/wiki/Morse_code](http://en.wikipedia.org/wiki/Morse_code)). If you have ever played “Hangman” or watched Wheel of Fortune, that experience will help you in this exercise. Your goal is to create a mathematical mapping of the alphabet to encode a message in a way that is not easily decipherable. However, you may create no more than three rules for your cipher. These rules are single statements that are used to transform the message.....Turn in a complete description of the three rules used for your cipher along with a one-to-one mapping of characters to their encoded values to your TA. The one-to-one mapping shows that your code is consistent and two characters can’t be mixed up ... Then, post three separate coded “sentences” (questions) of your choice on your wiki page ... Each of these sentences must be a question with a readily available answer.

- **Tasks:** Week 1: Developing rules for the cipher and post questions in “code”; Week 2: Deciphering other teams’ code and answering analysis and reflection questions
- **CS Lightbulbs:** Code generation, Encoding, Mapping rules, Encryption
- **Tips:** Ciphers must have one-to-one mapping and can’t have “secret” or impossibly obscure rules.
EXERCISES | EXPLORING

Description: Each group will be pick a location on campus from one of the 10 locations on the campus map posted as a PDF. Each member of the group must make several observations at your group’s location, as well as taking a picture of yourself making observations at this location. Use the data collection Form (Appendix A) to record your results and to verify that you’ve gathered sufficient data. … You must complete your observations during Week 1 of this exercise. Each group member must also upload as an attachment a photo of herself/himself observing in order to receive credit for Week 1. You can do this after you have created the above team wiki page. … After you have collected the data, you must record it on the above group wiki page. You will insert a table on the group wiki page following the format of Table 1 in Appendix to record your data. You can create your table in the wiki editor or cut and paste from Word into the wiki page.

- **Tasks**: Week 1: Collecting data; Week 2: Answering analysis and reflection questions
- **CS Lightbulbs**: Data structure, data organization, arrays, brainstorming, usage data collection
- **Tips**: Use all of your senses; pay attention to what’s around you; see a familiar place in a new way
Description: For these next two weeks, you’ll be playing a guessing game with your peers! Your group will be playing the guessing game as a team against another group. You can find another group by using the Groups wiki page written by the TA. … Each group will be given two photos which they will use as host, one for each of two objects assigned only to your group, together with specific verbal descriptions of these two objects. These specific descriptions will be of the form “[adjective], [adjective] [noun]”. The noun may be a compound noun (e.g., “swimming pool” or “waste basket” or “doormat”).…. You will play the following game once for each object…. The host team must lead the guesser team to the correct answer by following these steps (in a strict turn-based manner)…

- **Tasks**: Week 1: Playing the simile game twice; Week 2: Answering analysis and reflection questions
- **CS Lightbulbs**: Loops, database indexing, abstraction, generalization, relation, flowchart
- **Tips**: The hardest part as a host is not referring to the object’s function or classification; the hardest part as a guesser is to think metaphorically and broadly before closing in on a guess.
Description: For the next two weeks, you will be doing black box testing on the alien “health machine” described in Appendix A. To help you with your analysis, we recommend that you view the following YouTube video by Bret Pettichord on testing. It is a video of a keynote address he gave on testing and how the Wright Brothers had to invent the wind tunnel to test the components of the airplane before they could invent the airplane … Each group must devise a strategy for black box testing of the health machine described in Appendix A. Discuss your strategies as a group and archive them in a table form as shown in Appendix B …

- **Tasks**: Week 1: Devising a testing strategy; Week 2: Answering analysis and reflection questions
- **CS Lightbulbs**: Testing, evaluation
- **Tips**: Use all of your senses to think about EVERY way you might test this machine.

We, of the interplanetary ship Donator, do hereby give to Outer Limits Hospital a health machine, to be used under the control of you, the Directors of the Hospital. Not only will the sick benefit from the machine but it will be educational for you to determine who should be allowed to use it. … The health machine will cure any patient of any disease whatsoever, … It can treat only one patient at a time and each patient requires 24 hours for the treatment to be completed. Patients do not need to remain close to the machine throughout that time. … Warning: (1) If removed from Outer Limits Hospital, the machine will cease to operate. (2) The machine cannot be copied. (3) The machine will automatically destroy itself if opened. No other person on the planet knows what has occurred. …
Description: Your team has won a lucrative contract to consult with the government of a distant planet called Didymos. Didymos revolves around two suns and much of the planet is desert, with a harsh climate. Economic development is centered around oases where water is at or near the surface. The inhabitants of Didymos are divided into four factions with populations of similar size: Underground Miners … Farmers … Herders … Urbanites … The nature of the Didymos binary star system means that the seasons are not constant. These varying seasons, coupled with seasonal needs for some of the factions, make it difficult to come up with a single calendar with days off and holidays suitable for all factions. However, Amendment #27 to the constitution of the Intergalactic Federation requires each star system to have such a calendar to discourage discrimination against different groups … Each member of your group will choose one of the four factions and represent their interests in your group … As a whole, your group will collaborate and negotiate to create a satisfactory calendar.

- **Tasks**: Week 1: Creating calendar; Week 2: Answering analysis and reflection questions
- **CS Lightbulbs**: Conditionals, software design, constraint satisfaction, requirements,
- **Tips**: Appreciate the “creative abrasion” resulting from competing interests; discard assumptions about the calendar being a fixed and immutable system; recognize the utility of “seasons” and “holidays”
**EXERCISES | MARBLE MAZE**

**Description:** Using ONLY the specified materials, you will be making a maze through which a marble will travel. ... The completed maze must have an intentional path, with a clear beginning and ending point. This path can include loops. ... You may start the marble’s journey through the maze by dropping it. The marble may not travel on the floor more than 4 inches. The marble must travel from a minimum height of 4 inches and must travel a minimum of 4 linear feet (in any configuration) and must reach the end of the maze without stopping. Any pathway for the marble must be elevated at least 4” off the floor. The marble should take at least 4 seconds to complete its travel from beginning to end. Bonus points will be rewarded to the team that builds a maze on which the marble has the longest traversal time.

- **Tasks:** Week 1: Developing and testing individual segments; Week 2: Integrating segments into one single maze and answering analysis and reflection questions
- **CS Lightbulbs:** Modules, unit testing, integration testing, collaboration,
- **Tips:** The biggest challenge is just getting every group member to make a module.
Description: Similar to Marble Maze I, but integrating segments from new members … segments likely will not work with existing segments …

- **Tasks**: Week 1: Integrating segments into one single maze; Week 2: Answering analysis and reflection questions
- **CS Lightbulbs**: Modules, incorporating new modules, unit testing, integration testing, collaboration,
- **Tips**: Putting together existing modules into a new maze can be much easier than the initial maze; many students like doing something physical and working with materials.
Description: Using a grid, you will be designing a geometric visual pattern. You will then create a set of written instructions (an algorithm) for another group to follow which will accurately generate your geometric pattern on another grid … Think of your pattern as a module that can be repeated, reflected and rotated to generate complex patterns…. Your group will generate a grid which is 3 x 3 with at least ¼ inch squares. You can generate this grid at [http://incompetech.com/graphpaper/lite/](http://incompetech.com/graphpaper/lite/) and download a PDF for printing if you do not have ¼ inch graph paper. Label your grid axes such that the origin (0,0) is in the upper left corner.

- **Tasks**: Week 1: Creating visual patterns and generating written instructions; Week 2: Generating other teams’ visual patterns by following their instructions and answering analysis and reflection questions
- **CS Lightbulbs**: Boundary conditions, instruction set, algorithm, execution of algorithm
- **Tips**: Think of this as a mashup of drawing and game playing with simple rules (but potential complexity)
Description: An extension of Pathfinding ... Using a base 3 x 3 pattern from Exercise Path Finding I, build a complex 9 x 9 pattern using three operations: (1) shift, (2) reflect, and (3) rotate. The shift operation allows a base pattern to be replicated at a different location. The reflect operation allows a base pattern to be reflected horizontally or vertically. The rotate operation allows a base pattern to be rotated 90 degrees clockwise or counterclockwise ... You are also allowed to use a loop structure to repeat to generate multiple shifts, reflections, or rotations.

- **Tasks:** Week 1: Creating visual patterns and generating written instructions; Week 2: Generating other teams’ visual patterns by following their instructions and answering analysis and reflection questions

- **CS Lightbulbs:** Looping, variables, relative coordinates

- **Tips:** Week 2 is where the fun begins as the patterns can become complex and yet still be generated by simple rules
RESULTS | BRIEF OVERVIEW

We designed and deployed four computational creativity exercises during the Fall 2012 semester at the University of Nebraska, Lincoln, NE using the Written Agora system

• Found a “dosage” effect that higher grades and learning of core computational thinking principles were associated with increasing exercise completion from 0-1 to 4 exercises

The same four Computational Creativity Exercises (CCEs) we previously used were implemented in Spring 2013 in the introductory CS1 course tailored to engineering majors

• Confirmed our prior findings that the CCEs positively impact student learning of computational thinking and CS knowledge and skills
LOGISTICS | TIPS

When exercises were fully integrated into the class, students enjoyed them (e.g., in Ramsay’s CSCE155T) or appreciated the rationales behind the exercises (e.g., in Soh’s CSCE155N)

• Discuss each exercise in class (5-10 minutes) going through the assignment

• Explicitly map activities in exercise to topics in class

• Relate both computational and creative objectives to real-world problems

• Count 3-5% towards the final course grade depending on the number of exercises you assign
LOGISTICS | TIPS

When exercises were fully integrated into the class, students enjoyed them (e.g., in Ramsay’s CSCE155T) or appreciated the rationales behind the exercises (e.g., in Soh’s CSCE155N)

- Discuss each exercise in class (5-10 minutes) going through the assignment
- Explicitly map activities in exercise to topics in class
- Relate both computational and creative objectives to real-world problems
- Count 3-5% towards the final course grade depending on the number of exercises you assign
LOGISTICS | SUPPORT

The Computational Creativity team will provide

- Support revision and integration of exercises into course
- Support for grading of exercises
- Support for the Written Agora platform
- Mentoring for exercise deployment
- Statistics and results after the semester
LOGISTICS | EVALUATION

For Grant Evaluation and understanding the impacts of the exercises and how to implement, the evaluation will include:

• Surveys of students at the start, middle, and end of the semester-the same surveys done last year for control classes.

• A post-course Web evaluation of the exercises-done with the end of the semester surveys.

• Focus groups with students (2-3 per class).

• Interviews with you at the end of the semester to get your feedback.

Dr. Shell and his evaluation team will handle all recruitment and administration of surveys and will conduct focus groups and interviews.

Focus groups and interviews are done outside class time.

Surveys will be done in class or lab like they were for control classes.
Feedback from Instructors is important for us to improve the both the content and delivery

- Mapping to the subject matter in your class
- Logistical issues
- Observation of students
  - E.g., monitor student group work and address problems

Help us encourage student responses to our surveys

Report errors and suggestions to the content and design of exercise to us

- Including ideas for new exercises

We will do a formal evaluation interview with each of you at the end of the semester, but value and encourage ongoing feedback during the implementation.
REFERENCES


IC2Think project website: http://cse.unl.edu/agents/ic2think

K-12 versions available on Google’s Exploring Computational Thinking website: https://www.google.com/edu/computational-thinking/

Versions also available on Ensemble’s Computing Portal: http://www.computingportal.org/ (search key: IC2THINK)

Email: lksoh@cse.unl.edu, dshell2@unl.edu

This material is based upon work supported by the National Science Foundation under Grant No. 1122956 and No. 1431874. Additional support was provided by a UNL Phase II Pathways to Interdisciplinary Research Centers grant.