Session 6L: Lightning Talks Session 1
1:45 – 3:00 p.m.

Please hold your applause and questions until all the talks are done.

Vote for your favorite talk – use Whova!
Exploring a student-centered approach to innovating computer science education, **Madeleine Lorås**

Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, **Alex Edgcomb**, Frank Vahid

Designing active mediated learning tasks: Can small failures enhance student learning?, **Cruz Izu**, Olga Sanchez Castro

*Interest-driven coding projects, Jared O'Leary*

Facilitating multiple programming languages in one space, **Jared O'Leary**

*IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball*

We Should Give Messy Problems and Make Students Reflect on What They Learn, **Paul Dickson**

*Teaching Students a Systematic Approach to Debugging, Roman Lysecky*, Frank Vahid

Improving Course Content and Providing Intelligent Support Simultaneously, **Toby Dragon**

*Recruiting Experts: Toward a Concept Inventory for Computer Science 2, Lea Wittie*, Anastasia Kurdia, Meriel Huggard

OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, **Michael Kirkpatrick**

*Developing Computer Forensics Minor: Challenges and Opportunities, Yana Kortsarts*, Adam Fischbach, Suk-Chung Yoon

Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, **Debasis Bhattacharya**
Exploring a student-centered approach to innovating computer science education,

**Madeleine Lorås**

Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid

Designing active mediated learning tasks: Can small failures enhance student learning?,

Cruz Izu, Olga Sanchez Castro

Interest-driven coding projects, Jared O'Leary

Facilitating multiple programming languages in one space, Jared O'Leary

IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball

We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson

Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid

Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon

Recruiting Experts: Toward a Concept Inventory for Computer Science 2,

Lea Wittie, Anastasia Kurdia, Meriel Huggard

OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick

Developing Computer Forensics Minor: Challenges and Opportunities,

Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon

Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
EXPLORING A STUDENT-CENTERED APPROACH TO INNOVATING COMPUTER SCIENCE EDUCATION

Madeleine Lorås, PhD candidate
Department of Computer Science, Norwegian University of Science and Technology
Students taking a course

Older students who have taken the course

Older students hired as teaching assistants (peer educators)
CURRENT ISSUES

Increasing number of computer science students

Computer science majors and non-majors

Difference in prerequisites

Retention - throughput - diversity
Interviews and surveys
During and after

Student apps
Organizational improvements

Challenge
What is an educational improvement?

Who defines what is better?
Thanks for Listening!
Questions and comments are welcome.

Madeleine Lorås, PhD candidate
Department of Computer Science,
Norwegian University of Science and Technology

@madelelo  madeleine.loras@ntnu.no  www.ntnu.edu/excited
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås

Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, **Alex Edgcomb, Frank Vahid**

Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro

Interest-driven coding projects, Jared O'Leary

Facilitating multiple programming languages in one space, Jared O'Leary

IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball

We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson

Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid

Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon

Recruiting Experts: Toward a Concept Inventory for Computer Science 2, Lea Wittie, Anastasia Kurdia, Meriel Huggard

OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick

Developing Computer Forensics Minor: Challenges and Opportunities, Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon

Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
Interactive language-neutral flowcharts and code for teaching core programming concepts

ALEX EDGCOMB¹,², FRANK VAHID²,¹, ROMAN LYSECKY³,¹

¹ZYBOOKS, LOS GATOS, CALIFORNIA
²COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY OF CALIFORNIA, RIVERSIDE
³ELECTRICAL AND COMPUTER ENGINEERING, UNIVERSITY OF ARIZONA
integer x
integer y
integer max

x = Get next input
y = Get next input

if x > y
  max = x
else
  max = y

Coral is an ultra-simple language for learning to program.

Put max to output
integer x
integer y
integer max

x = Get next input
y = Get next input

if x > y
    max = x
else
    max = y

Put max to output
integer x
integer y
integer max

x = Get next input
y = Get next input

if x > y
  max = x
else
  max = y

Put max to output

The simulator shows step by step execution and variable updates.
The simulator helps learn constructs, such as by highlighting each for loop part when the part executes.
Function calls are made crystal clear, including showing the parameters, local variables, and return variable.
An array's elements are explicitly shown in memory, including the size.

integer array(5) userNums
integer i

for i = 0; i < userNums.size; i = i + 1
    userNums[i] = i * 2
for i = 0; i < userNums.size; i = i + 1
    Put userNums[i] to output
The simulator auto-generates a flowchart for given code (mimicking the code layout) and executes the flowchart graphically.
The flowchart makes looping functionality obvious. Users can switch between flowchart and code.
The flowchart makes looping functionality obvious. Users can switch between flowchart and code.
CoralLanguage.org

The simulator is web-based and free to students and instructors, and comes with a free tutorial.

- Click "Enter execution", then "Step" to execute one statement, or "Run" to execute entire program. You can view the code or flowchart execution.
Snap (and Scratch)
Image from http://snap.berkeley.edu/

Raptor
Image from https://stackoverflow.com/a/23299479

Python
Visit: CoralLanguage.org

Suggested uses:

• CS1: Visualize branches, loop, & variable updates
• Gentle programming class: Use Coral, end with Python/Java/C++
• Brief coding intro: AP CS Principles, Intro to Computing Tech, etc.

Thoughts: alex.edgcomb@zybooks.com, vahid@cs.ucr.edu, rlysecky@ece.arizona.edu
Support: support@zybooks.com
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid

**Designing active mediated learning tasks: Can small failures enhance student learning?,**

**Cruz Izu, Olga Sanchez Castro**

Interest-driven coding projects, Jared O'Leary
Facilitating multiple programming languages in one space, Jared O'Leary
IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball
We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson
Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid
Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon
Releasing Experts: Toward a Concept Inventory for Computer Science 2,
Lea Wittie, Anastasia Kurdia, Meriel Huggard
OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities,
Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon
Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
DESIGNING ACTIVE LEARNING TASKS: CAN SMALL FAILURES ENHANCE STUDENT LEARNING?

Cruz Izu, The University of Adelaide
Olga Sanchez Castro, Flinders University
ARE FAILURES ALWAYS BAD?

“Failure is simply the opportunity to begin again, this time more intelligently.”

Henry Ford

“Failures are finger posts on the road to achievement.”

C.S. Lewis

“Science, my lad, is made up of mistakes, but they are mistakes which it is useful to make, because they lead little by little to the truth”

Jules Verne

“It's fine to celebrate success but it is more important to heed the lessons of failure.”

Bill Gates
WHAT SoLT SAYS ABOUT IT

**Avoid** failure through scaffolding

ZDP  (Lev Vygotsky)

---

**Embrace** failure as preparation for learning

Productive failure  (M. Kapur)

Prob. Solving –Instruction (K. Loibl et al)

We *deliberately* design for controlled failure

and then bootstrap that failure into something that’s productive.

---

C. Izu and O. Sanchez-Castro

*Lightning Talk – SIGCSE 2018*
**WHY PS-I WORKS?**

- **Problem solving phase (with failure)**
- **Mechanism 1: activation of prior knowledge**
- **Mechanism 2: Awareness of knowledge gaps**
- **Mechanism 3: Recognition of key features**
- **Instruction phase**
- **Outcome: well connected and organized knowledge**

Figure extracted from (Loibl, Roll and Rummel 2016) *Towards a theory of when and how problem solving followed by instruction supports learning.*

Educ Psychol Rev 1–23
CALL FOR INTERESTED PARTNERS

- Looking for **interdisciplinary** and/or **multi-institutional** partners

**Stage 1**  
Review of the literature on Productive Failure (PF)

**Stage 2**  
Define a PF intervention for CS1 course(s)  
- Apply PS-I on a course module  
- Compare with control

**Stage 3**  
Further explore PF in higher education  
- Other CS courses/concepts  
- SLA (second language acquisition)  
- Other areas
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid
Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro

**Interest-driven coding projects, Jared O'Leary**

Facilitating multiple programming languages in one space, Jared O'Leary

IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball
We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson
Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid
Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon
Recruiting Experts: Toward a Concept Inventory for Computer Science 2, Lea Wittie, Anastasia Kurdia, Meriel Huggard
OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities, Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon

Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid
Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro
Interest-driven coding projects, Jared O'Leary
Facilitating multiple programming languages in one space, Jared O'Leary

IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball
We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson
Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid
Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon
Recruiting Experts: Toward a Concept Inventory for Computer Science 2, Lea Wittie, Anastasia Kurdia, Meriel Huggard
OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities, Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon
Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
IRT In 5 Minutes: Assessing Questions

Michael Ball
February 25, 2018
http://bjc.link/sigcse-irt-lightning
What Is Item Response Theory?

Developed by psychometricians, IRT provides a way to understand how effective a question is at predicting behavior.

Can be applied to Multiple Choice Questions, but much trickier to apply to open-ended work.
Item Response Function

https://www.rasch.org/rmt/rmt181b.htm
I *really* can’t do IRT in 5 minutes.
...so we’ll do something easy!
Simple Goal: Understand whether questions on an exam are predictive of a student’s score.
CORREL(\text{question\_score}, \text{exam\_score})
What can we learn?

- Overall, we want questions to be predictive of exam scores. That means students who got the question right will do well on the exam.
- Negative correlations indicate a question is “broken” in some way.
Scatter Plot: Good Question
Scatter Plot: Poor Question
Google Sheets Example Workbook

https://docs.google.com/spreadsheets/d/1pJAj_vK5M4yqdZGsygtS7YnobmDxa_t-nWT4DM5fEzI/edit?usp=sharing
Thank You

Questions, Feedback: ball@berkeley.edu
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid
Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro
Interest-driven coding projects, Jared O'Leary
Facilitating multiple programming languages in one space, Jared O'Leary
IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball

We Should Give Messy Problems and Make Students Reflect on What They Learn,

Paul Dickson

Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid
Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon
Recruiting Experts: Toward a Concept Inventory for Computer Science 2,
Lea Wittie, Anastasia Kurdia, Meriel Huggard
OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities,
Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon
Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
We Should Give Messy Problems and Make Students Reflect on What They Learn

Paul Dickson
“Paul, I think you should be more explicit here in step two.”

Based on the comic by Sidney Harris
“Paul, I think you should be more explicit here in step two.”

Based on the comic by Sidney Harris
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid
Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro
Interest-driven coding projects, Jared O'Leary
Facilitating multiple programming languages in one space, Jared O'Leary
IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball
We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson

**Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid**

Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon
Recruiting Experts: Toward a Concept Inventory for Computer Science 2, Lea Wittie, Anastasia Kurdia, Meriel Huggard
OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities, Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon

Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
Teaching Students a Systematic Approach to Debugging

Roman Lysecky\textsuperscript{1,3}, Frank Vahid\textsuperscript{2,3}

\textsuperscript{1} University of Arizona, rlysecky@ece.arizona.edu
\textsuperscript{2} University of California, Riverside, vahid@cs.ucr.edu
\textsuperscript{3} zyBooks
Many students have weak debugging skills

Program should compute $x^x$. ERROR: $x = 2$. Expected 4. Got 8.

Common behavior: Make random change & hope things improve

Error remains, and things got worse
Issue *not* just about debugging, but how people approach troubleshooting in general
First teach systematic troubleshooting process

- Troubleshooting process
  - Create a hypothesis. A hypothesis states a possible cause of a problem
  - Run a test. A test is a procedure whose result validates or invalidates a hypothesis
- Use everyday systems

### PARTICIPATION ACTIVITY

1.2.2: For each hypothesis, a test is used to validate or invalidate the hypothesis.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Test</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lamp wire is unplugged</td>
<td>View wire. Plugged in?</td>
<td>N</td>
</tr>
<tr>
<td>Bulb is broken</td>
<td>Insert bulb into working lamp. Lights?</td>
<td>Y</td>
</tr>
<tr>
<td>Outlet is dead</td>
<td>Plug working lamp into outlet. Lights?</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>
Then teach program debugging

- **Debugging process**
  - Create a hypothesis. A hypothesis states a possible cause of a problem
  - Run a test. A test is a procedure whose result validates or invalidates a hypothesis

### 2.1.3: Inserting debug output statements can help test each statement during debugging.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Test</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement 1 has a bug</td>
<td>Inspect output</td>
<td>OK</td>
</tr>
<tr>
<td>Statement 2 has a bug</td>
<td>Inspect output</td>
<td>Bad</td>
</tr>
<tr>
<td>Statement 3 has a bug</td>
<td>Visually inspect</td>
<td>Bug found</td>
</tr>
<tr>
<td>Statement 4 has a bug</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Free two-chapter zyBook (with sign in)

- Targeted at roughly the fifth week of a CS1 course
  - When students are beginning to face harder debugging challenges
- Also beneficial for any programming class beyond CS1
  - Ex: Could be used in the first few weeks of CS2
- Uses dozens of animations and learning question sets

http://www.zybooks.com/catalog/troubleshooting-basics/

Comments and feedback always welcome
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid
Designing active mediated learning tasks: Can small failures enhance student learning?,
Cruz Izu, Olga Sanchez Castro
Interest-driven coding projects, Jared O'Leary
Facilitating multiple programming languages in one space, Jared O'Leary
IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball
We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson
Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid

**Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon**

Recruiting Experts: Toward a Concept Inventory for Computer Science 2,
Lea Wittie, Anastasia Kurdia, Meriel Huggard
OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities,
Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon

Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
Improving Course Content and Providing Intelligent Support Simultaneously

Toby Dragon
The Challenge

• Wide array of teaching materials
  (Instructional Design challenge)

• Need for domain models to provide individualized support
  (Artificial Intelligence in Education challenge)
The Goal

• Support Students:
  • Offer high-level assessment
  • Suggest of materials / practice
    (use Artificial Intelligence in Education)

• Support instructors:
  • Relate various online / offline materials
  • Offer high-level student/class assessment
    (use Instructional Design)
The Approach

Concept Maps to represent course content
Using the Graph – Students

• Visualization of Instructor's Content Plan

• Quick Visualization of Assessment based on Concepts

• Automated Intelligent Suggestions
Using the Graph – Instructors

- Quick Visualization of Student Performance
- Automated Intelligent Suggestions
- Course Content Improvement
Thank you.

Improving Course Content and Providing Intelligent Support Simultaneously

Toby Dragon
tdragon@ithaca.edu
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid
Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro
Interest-driven coding projects, Jared O'Leary
Facilitating multiple programming languages in one space, Jared O'Leary
IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball
We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson
Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid
Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon

**Recruiting Experts: Toward a Concept Inventory for Computer Science 2,**
**Lea Wittie, Anastasia Kurdia, Meriel Huggard**

OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities,
Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon
Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
Recruiting Experts: Toward a Concept Inventory for Computer Science 2

Lea Wittie, Bucknell University
Anastasia Kurdia, Tulane University
Meriel Huggard, Trinity College Dublin

SIGCSE 2018
Concept Inventory

A multiple choice test

Has important and difficult topics from a course.

Each question has 1 correct answer and a bunch of misconception answers.
What to do with a Concept Inventory

Do those teaching technique changes actually improve learning?

How much did the course actually add?

What do I still not understand?

year after year
We are creating a concept inventory for **Computer Science 2**

**We want CS2 educators and researchers (experts) to do a Delphi consensus building process.**

- Send us some quick envelope info
- List the important, difficult CS2 topics
- Rank whole set for difficulty, importance
- Given stats, re-rank the set and justify when you deviate
- Given justifications, re-re-rank the set

Sign up at SIGCSE 2018 or email us at info@cs2ci.org
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid
Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro
Interest-driven coding projects, Jared O'Leary
Facilitating multiple programming languages in one space, Jared O'Leary
IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball
We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson
Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid
Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon
Recruiting Experts: Toward a Concept Inventory for Computer Science 2, Lea Wittie, Anastasia Kurdia, Meriel Huggard

OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities, Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon
Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
OpenCSF: Computer Systems Fundamentals

http://opencsf.org/

Part 1: Sequential Systems
- Intro to C
- Pointers and Memory
- Binary Representation
- Assembly Language: Data
- Assembly Language: Control
- Memory Hierarchy and Cache
- Interrupts and I/O

Part 2: Concurrent Systems
- Processes and OS Basics
- State & Timing Models
- Concurrency with IPC
- Networking Basics
- Multithreading
- Synchronization Primitives
- Synchronization Problems
- Parallel Decomposition
OpenCSF: Computer Systems Fundamentals

http://opencsf.org/

Goals:

• Open, online, interactive textbook
• ACM Computing Curriculum 2013 alignment
• Early feedback, active learning support
OpenCSF: Computer Systems Fundamentals

http://opencsf.org/

Goals:
- Open, online, interactive textbook
- ACM Computing Curriculum 2013 alignment
- Early feedback, active learning support

Future directions & needs:
- Web-based visualization & animation
- Integrated coding exercises (PythonTutor/C)
- Collaborators/readers/co-writers
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid

Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro

Interest-driven coding projects, Jared O'Leary

Facilitating multiple programming languages in one space, Jared O'Leary

IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball

We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson

Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid

Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon

Recruiting Experts: Toward a Concept Inventory for Computer Science 2, Lea Wittie, Anastasia Kurdia, Meriel Huggard

OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick

**Developing Computer Forensics Minor: Challenges and Opportunities, Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon**

Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom, Debasis Bhattacharya
Developing Computer Forensics Minor - Challenges and Opportunities

Adam Fischbach, Yana Kortsarts, Suk-Chung Yoon
Widener University, Chester, PA
<table>
<thead>
<tr>
<th>Semester</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall 2006</td>
<td>CSCI 130 Intro Computer Forensics</td>
</tr>
<tr>
<td>Fall 2013</td>
<td>Computer Forensics Lab</td>
</tr>
<tr>
<td>Fall 2014</td>
<td>CSCI 393 Advanced Computer Forensics</td>
</tr>
<tr>
<td>Fall 2014</td>
<td>Design of Computer Forensics Minor</td>
</tr>
<tr>
<td>Spring 2015</td>
<td>Minor Application Submitted to Committee</td>
</tr>
<tr>
<td>Fall 2016</td>
<td>Computer Forensics Minor Approved</td>
</tr>
</tbody>
</table>
## Core Courses

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJ 105</td>
<td>Introduction to Criminal Justice</td>
</tr>
<tr>
<td>CJ 268</td>
<td>Cybercrime</td>
</tr>
<tr>
<td>CSCI 391</td>
<td>Practical Cryptology</td>
</tr>
<tr>
<td>CSCI 392</td>
<td>Network and Computer Security</td>
</tr>
<tr>
<td>CSCI 393</td>
<td>Computer Forensics</td>
</tr>
</tbody>
</table>
# Curriculum 30/31 Credits

## One Math/Logic Course

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATH 151</td>
<td>Discrete Math</td>
</tr>
<tr>
<td>PHIL 120</td>
<td>Symbolic Logic</td>
</tr>
<tr>
<td>PHIL 105</td>
<td>Introduction to Logic</td>
</tr>
</tbody>
</table>

## One Programming Course

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 151</td>
<td>Intro Programming (CS Majors)</td>
</tr>
<tr>
<td>CSCI 131</td>
<td>Intro Programming (Non-CS Majors)</td>
</tr>
</tbody>
</table>
## Curriculum 30/31 Credits

### Two Criminal Justice Courses

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJ 210</td>
<td>Criminal Courts</td>
</tr>
<tr>
<td>CJ 225</td>
<td>Principals of Criminal Investigation</td>
</tr>
<tr>
<td>CJ 305</td>
<td>Criminal Evidence</td>
</tr>
<tr>
<td>CJ 320</td>
<td>White Collar Crime</td>
</tr>
</tbody>
</table>

### One Ethics Course

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIL 352</td>
<td>Business Ethics</td>
</tr>
<tr>
<td>PHIL 350</td>
<td>Ethics</td>
</tr>
<tr>
<td>CJ 405</td>
<td>Ethics in Criminal Justice</td>
</tr>
</tbody>
</table>
Lab Facilities

FRED Forensic Recovery of Evidence Device
Digital Intelligence
12 workstations

Software: Access Data FTK
Academic Program Subscription
Challenges

- Administrative Issues
  - Hosting Division: Science/Social Science
  - College of Arts and Sciences
- Curriculum Issues
  - Finding Interdisciplinary Balance
  - Staffing for Required Courses
- Cost Issues
  - Software/Hardware Cost
- Student Recruitment/Advising
Exploring a student-centered approach to innovating computer science education, Madeleine Lorås
Interactive, language-neutral flowcharts and pseudocode for teaching core CS0/1 programming concepts, Alex Edgcomb, Frank Vahid
Designing active mediated learning tasks: Can small failures enhance student learning?, Cruz Izu, Olga Sanchez Castro
Interest-driven coding projects, Jared O'Leary
Facilitating multiple programming languages in one space, Jared O'Leary
IRT In 5 Minutes: Easy Ways to Better Understand An Assessment, Michael Ball
We Should Give Messy Problems and Make Students Reflect on What They Learn, Paul Dickson
Teaching Students a Systematic Approach to Debugging, Roman Lysecky, Frank Vahid
Improving Course Content and Providing Intelligent Support Simultaneously, Toby Dragon
Recruiting Experts: Toward a Concept Inventory for Computer Science 2, Lea Wittie, Anastasia Kurdia, Meriel Huggard
OpenCSF: An Online Interactive Textbook for Computer Systems Fundamentals, Michael Kirkpatrick
Developing Computer Forensics Minor: Challenges and Opportunities, Yana Kortsarts, Adam Fischbach, Suk-Chung Yoon

**Bitcoins, Blockchains and Cybersecurity: Teaching emerging topics in the classroom,**
**Debasis Bhattacharyya**
Bitcoins, Blockchains and Cybersecurity

Teaching Emerging Topics in the Classroom
Debasis Bhattacharya, JD, DBA
URL: http://maui.Hawaii.edu/cybersecurity
University of Hawaii Maui College
SIGCSE 2018 - Lightning Talk
$8K Again? Bitcoin Is Up Nearly $2K from Today's Low - CoinDesk
https://www.coindesk.com › News
8 hours ago - NEWS. Bitcoin jumped almost $2,000 from its intraday low Tuesday, making up some of the losses it sustained from 2018’s January correction. As of press time, bitcoin was trading around $7,900, after hitting a low of $5,947 a little over 12 hours ago. The world’s largest cryptocurrency by market cap opened ...

Bitcoin News: News
https://news.bitcoin.com/ ▼
news.Bitcoin.com is the world’s premier 24/7 news feed covering everything bitcoin-related.

Is Bitcoin's Risk On Trade Back In Play? - Forbes
3 hours ago - Bitcoin and other cryptocurrency prices moved higher on Tuesday. They were coming off lows and mirroring the Risk On trade seen in the equity markets.

Bitcoin price warning: Cryptocurrency to reach $100,000 if it keeps ...
https://www.express.co.uk › Finance › City & Business ▼
13 hours ago - BITCOIN’s prices fell dramatically amid a frenzied sell-off following news the Chinese Government and Lloyd’s bank would ban buying BTC with credit cards. But independent crypto trend analyst Ronnie Moas said bitcoin’s value could reach $100,000 if it maintains its current market share.
1 BTC = $7,353.16

Interactive Chart
TRANSACTIONS PER DAY

The number of bitcoin transactions in the last 24 hours.

2 2 8 3 0 3

Transactions since Mon Feb 05 2018 7:46:28 PM.

MARKET CAP: $115,222,434,845.00

HASH RATE: 23,938,939.78 TH/s
Bitcoin ‘SKYROCKETS’
Cryptocurrency soars 25 per cent in 24 hours as 'investors celebrate'

A BITCOIN resurgence could be underway as the cryptocurrency soared over 24.5 per cent in the last 24 hours that has surely given investors an excuse to celebrate, it has been revealed.

By JOSEPH CAREY
PUBLISHED: 05:40, Wed, Feb 7, 2018 | UPDATED: 05:41, Wed, Feb 7, 2018
CEO of Major UK-Based Cryptocurrency Exchange Kidnapped in Ukraine

Wednesday, December 27, 2017  Mohit Kumar

Pavel Lerner, a prominent Russian blockchain expert and known managing director of one of the major crypto-exchanges EXMO, has allegedly been kidnapped by "unknown" criminals in the Ukrainian capital [...]

Largest Crypto-Mining Exchange Hacked; Over $70 Million in Bitcoin Stolen

Wednesday, December 06, 2017  Mohit Kumar

Bitcoin is breaking every record—after gaining 20% jump last week, Bitcoin price just crossed the $14,800 mark in less than 24 hours—
Separating the hype from the technology

• Crypto currencies are becoming popular with banks, consumers and various industries.
• There is a need for consumers and students to understand the basic underlying technology behind these crypto currencies and the underlying value, security risks and concerns.
Core Topics

• Cryptocurrencies
  • Bitcoin (BTC, started in January 2009 after paper by “Satoshi Nakamoto”
  • Altcoins - Ethereum, Ripple, Bitcoin Cash, Cardano, Litecoin, NEM, Stellar, NEO, IOTA, Dash, Monero, TRON

• Blockchain (invented by Satoshi Nakamoto in 2008, as public transaction network for Bitcoin)

• Mining
• Proof of Work
• Wallet
• Exchange
• Initial Coin Offering (ICO)
• Regulations
• Cybersecurity
Embed within Traditional Programs/Courses

• Accounting
• Finance
• Business
• Computer Science
• Information Technology
• Cybersecurity
• Administration of Justice
• Law
• Etc...
Examples from CompSci and Business

• Computer Science
  • Proof of Work Protocol
  • Economic measure to deter DDOS or Spam
  • Extend understanding to create transaction block in a chain
  • Mining computers, hash rate
  • Electricity consumption

• Business: Accounting, Finance, Supply Chain, Security
  • Tax and government regulations
  • Distributed ledger for private/public supply chain
  • Cryptocurrencies for payments
  • Initial Coin Offering (ICO) for startups
  • Security of Wallets and Exchanges
Session 8L: Lightning Talks Session 2

3:45 – 5:00 p.m.