Department of Computing Sciences

B.S. Computer Science
B.S. Information Systems
B.S. Information Technology

William M. Jones, Ph.D.
Associate Professor & Department Chair

http://cs.coastal.edu
Department Composition

Dr. Crystal Cox
Assistant Professor
CSCC 160F
843.349.2144
crystal@coastal.edu
[faculty page]

Dr. Jean H. French
Assistant Professor
CSCC 160H
843.234.3430
jennifer@coastal.edu
[faculty page]

Dr. William Jones, Jr.
Chair & Associate Professor
CSCC 160N
843.349.4142
wjones@coastal.edu
[faculty page]

Dr. Sathish A.P Kumar
Assistant Professor
CSCC 160L
843.349.2810
skumar@coastal.edu
[faculty page]

Prof. Jeff Linder
Assistant Professor
CSCC 160C
843.349.2829
jeff@coastal.edu
[faculty page]

Dr. Michael Murphy
Assistant Professor
CSCC 160D
843.349.2834
mmurphy2@coastal.edu
[faculty page]

Dr. Erin Rickard
Assistant Professor
CSCC 160E
843.349.2098
hrickard@coastal.edu
[faculty page]

Prof. Joyce Barnes
Lecturer
CSCC 160B
843.349.2359
jabarnes@coastal.edu
[faculty page]
[FA14 Schedule]

Prof. Clint Fuchs
Lecturer
CSCC 160J
843.349.2676
clintf@coastal.edu
[FA14 Schedule]

Prof. Cory Nance
Lecturer
CSCC 160I
843.349.2846
ccnance@coastal.edu
[faculty page]
[FA14 Schedule]

Prof. Eugene Collins
Professor Emeritus
CSCC 160L
843.349.2072
gene@coastal.edu
[faculty page]

Dr. Steve Sheel
Distinguished Professor Emeritus
CSCC 160A
843.349.2116
843.283.9991 (cell)
steves@coastal.edu
[faculty page]

Mrs. Lesley Holody
Admin. Specialist
CSCC 160M
843.349.2116
lholody@coastal.edu
[staff page]

8 tenure-track
3 lecturers
2 professors emeritus
1 administrative assistant
FALL 2015 ENROLLMENT

- Computer Science, 190, 48%
- Information Technology, 68, 17%
- Information Systems, 89, 23%
- Computer Science Pre-Engineering, 49, 12%

Total: 396 students
Number of Majors by Class Level - September 2015

Breakdown by Class Level
396 Total Students in Department

- Freshmen: 143 students
  - IT: 64 majors
  - CSCI: 42 majors
  - CSC-E: 18 majors
  - INFSY: 19 majors

- Sophomore: 97 students
  - IT: 53 majors
  - CSCI: 24 majors
  - CSC-E: 4 majors
  - INFSY: 3 majors

- Junior: 76 students
  - IT: 39 majors
  - CSCI: 16 majors
  - CSC-E: 3 majors
  - INFSY: 0 majors

- Senior: 80 students
  - IT: 34 majors
  - CSCI: 31 majors
  - CSC-E: 15 majors
  - INFSY: 0 majors
Let’s Take a Closer Look ...
**Computer Science**

Application development *(broad focus)*
- algorithms
- data structures
- high-performance computing
- theory of computing
- image processing
- robotics

Programming
- 130
- 140/L
- 150/L
- 170
- 330
- some electives

Gate-level hardware and above

**Information Systems**

Application development *(web-centric)*
- networking
- security
- system admin.
- forensics

**ANY 18+ HOUR Minor**

~120 Credits

according to Dr. Jones
Computer Science Continued

- CSCI 350 (Programming Languages)
- CSCI 356 (Operating Systems)
- CSCI 380 (Advanced Algorithms)
- CSCI 390 (Theory of Computing)
- CSCI 300+ (required elective)
- SCIENCE I/L (Lab-based Science I)
- SCIENCE II/L (Lab-based Science II)
- "Pick Three"
  - CSCI 440 (Graphics)
  - CSCI 445 (Image Processing)
  - CSCI 490 (Software Engineering II)
  - CSCI 425 (Databases II)
  - CSCI 473 (Parallel Programming)
  - CSCI 485 (Robotics)
- MATH or SCI. (required elective)

"Pick Three" is mandatory for students.

Courses in blue are required.
Courses in yellow are required electives.
Courses in orange are required electives.
Courses in purple are lab-based.

# Information Systems Continued

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI 200+</td>
<td>(elective)</td>
</tr>
<tr>
<td>CSCI 335</td>
<td>(Project Management)</td>
</tr>
<tr>
<td>CSCI 370</td>
<td>(Networking)</td>
</tr>
<tr>
<td>CSCI 495</td>
<td>(Capstone)</td>
</tr>
<tr>
<td>CSCI 365</td>
<td>(SEO)</td>
</tr>
<tr>
<td>CSCI 375</td>
<td>(Multimedia)</td>
</tr>
<tr>
<td>CSCI 385</td>
<td>(Security)</td>
</tr>
<tr>
<td>CSCI 385</td>
<td>(Security)</td>
</tr>
<tr>
<td>CSCI 387</td>
<td>(Decision Support Sys.)</td>
</tr>
<tr>
<td>CSCI 415</td>
<td>(Windows System Admin.)</td>
</tr>
<tr>
<td>CSCI 416</td>
<td>(Linux System Admin.)</td>
</tr>
<tr>
<td>CSCI 425</td>
<td>(Databases II)</td>
</tr>
<tr>
<td>CSCI 434</td>
<td>(Digital Forensics)</td>
</tr>
<tr>
<td>CSCI 444</td>
<td>(Human Comp Interaction)</td>
</tr>
<tr>
<td>CSCI 475</td>
<td>(Decision Support Sys.)</td>
</tr>
<tr>
<td>CSCI 490</td>
<td>(Software Engineering II)</td>
</tr>
</tbody>
</table>

"Pick four"
CSCI 335 (Project Management)
CSCI 370 (Networking)
CSCI 385 (Security)
CSCI 415 (Windows System Admin.)
CSCI 416 (Linux System Admin.)
CSCI 427 (Systems Integration)
CSCI 444 (Human Comp Interaction)
CSCI 300+ (required elective)

MINOR (Class 5)
MINOR (Class 6)
Degree Program History

B.S. in Computer Science  
(1985)  
Updated Traditional Curriculum - ACM 2013 Specification

B.S. in Information Systems  
(2009)  
Progressive Web-centric Curriculum - AIS & ACM at the core

B.S. in Information Technology  
(2014)  
True 2+2 program with HGTC
One Degree Program in Planning

M.S. in Information Security and Data Analytics

Fall 2016?

Approved by SACS Nov. 2015
Courses being offered beginning Summer 2016

Completely online
Options: Thesis, non-thesis, and two certificates
ABET Accreditation

B.S. in Computer Science

*Initial Accreditation:* 2003
*1st Re-accreditation:* 2009
*2nd Re-accreditation:* 2014

B.S. in Information Systems

*Initial Accreditation:* 2014

B.S. in Information Technology

*Initial ABET Site Visit:* TBD
Graduation Statistics

**Recent Graduates:**

<table>
<thead>
<tr>
<th>Year</th>
<th>CSCI</th>
<th>INFSY</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>12</td>
<td>29</td>
<td>42</td>
</tr>
<tr>
<td>2013</td>
<td>12</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>2012</td>
<td>13</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>2011</td>
<td>20</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>2010</td>
<td>23</td>
<td>5</td>
<td>28</td>
</tr>
<tr>
<td>2009</td>
<td>10</td>
<td>1</td>
<td>11</td>
</tr>
</tbody>
</table>

**2015 Graduates:**

<table>
<thead>
<tr>
<th></th>
<th>Spring</th>
<th>Summer</th>
<th>Fall*</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCI</td>
<td>8</td>
<td>0</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>INFSY</td>
<td>12</td>
<td>1</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>IT</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

*anticipated
## Faculty Scholarship

### Faculty Publications

<table>
<thead>
<tr>
<th>Year</th>
<th># of Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>6</td>
</tr>
<tr>
<td>2007</td>
<td>10</td>
</tr>
<tr>
<td>2008</td>
<td>11</td>
</tr>
<tr>
<td>2009</td>
<td>13</td>
</tr>
<tr>
<td>2010</td>
<td>15</td>
</tr>
<tr>
<td>2011</td>
<td>9</td>
</tr>
<tr>
<td>2012</td>
<td>17</td>
</tr>
<tr>
<td>2013</td>
<td>12</td>
</tr>
<tr>
<td>2014</td>
<td>7</td>
</tr>
</tbody>
</table>

### External Funding

<table>
<thead>
<tr>
<th>Year</th>
<th>Funding Awarded</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>$0</td>
</tr>
<tr>
<td>2007</td>
<td>$0</td>
</tr>
<tr>
<td>2008</td>
<td>$0</td>
</tr>
<tr>
<td>2009</td>
<td>$585</td>
</tr>
<tr>
<td>2010</td>
<td>$2,693</td>
</tr>
<tr>
<td>2011</td>
<td>$67,870</td>
</tr>
<tr>
<td>2012</td>
<td>$20,671</td>
</tr>
<tr>
<td>2013</td>
<td>$54,971</td>
</tr>
<tr>
<td>2014</td>
<td>$10,600</td>
</tr>
</tbody>
</table>
Grand Strand Tech Expo
2012, 2013, 2014
courtesy of Sun News
Numbers & Bytes Club

TEDxUNC (2014)

E3 Conference (2013)

CCU Meeting

Palmetto Supercomputer (2014)

Techno Security Conference (2014)
Other Achievements

Ranked #1 in University for Program Assessment

Degree In Three

2 of only 15 degree programs offered as degree-in-three

Many External Invited Speakers Including Google
CSCI 210
Digital Logic Design and Assembly Programming
CSCI 310: Computer Architecture
CSCI 445: Image Processing

Edge Detection
Face Isolation
Automatic Classification
Image Segmentation
CSCI 440: Computer Graphics

Ray-tracing

Image transformations

OpenGL
CSCI 473: Introduction to Parallel Systems

Palmetto Supercomputer 90th fastest in the world

LittleFe Portable System

Field Trips!!!
CSCI 356: Operating Systems

User Interface

Supporting Applications

Supporting Applications

Supporting Applications

Kernel

CPU Processes

Memory

File I/O

Device I/O

Cloud Computing

everything and the kitchen sink
The R-L-C Circuit

Sum the voltage drops around the loop:

$$\varepsilon = I R + L \frac{dI}{dt} + \frac{1}{C} \int I \, dt$$

where, \( \varepsilon = V_o \, e^{i \omega t} \)

Assume an unknown \( I \) of the general form:

$$I(t) = I_o \, e^{i (\omega t - \phi)}$$

Then the voltage loop equation becomes:

$$V_o \, e^{i \omega t} = I_o \, R \, e^{i (\omega t - \phi)} + i \omega L \, I_o \, e^{i (\omega t - \phi)} + \frac{I_o}{i \omega C} \, e^{i (\omega t - \phi)}$$

Simplifying:

$$V_o = I_o e^{-i \phi} \left( R + i \omega L + \frac{1}{i \omega C} \right)$$

$$Z_R \quad Z_L \quad Z_C$$

( the “complex impedances”; units = Ohms)
Other Classes

- CSCI 330: Software Engineering
- CSCI 350: Programming Languages
- CSCI 380: Algorithms
- CSCI 390: Theory of Computation
- CSCI 450: Compiler Construction
- CSCI 480: Artificial Intelligence
- Many more!
INFORMATION SYSTEMS

@ CCU: NEW AND ALWAYS IMPROVING
# Information Systems

## Areas of Specialization

*(only a suggestion, other combinations are possible)*

<table>
<thead>
<tr>
<th>Software Development</th>
<th>System Administration and Security</th>
<th>Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Web (203)</td>
<td>Computer Infrastructure (211)</td>
<td>Data Structures (220)</td>
</tr>
<tr>
<td>Application Development (409)</td>
<td>System Administration (415)</td>
<td>Organization of Programming Languages (350)</td>
</tr>
<tr>
<td>Software Engineering II (490)</td>
<td>Linux System Administration (416)</td>
<td>Operating Systems (356)</td>
</tr>
<tr>
<td>Database Systems Design (425)</td>
<td>Information System Security (385)</td>
<td>Algorithm Analysis (380)</td>
</tr>
<tr>
<td>Advanced Topics in Web Development (365)</td>
<td>Digital Forensics (434)</td>
<td>Theory of Computation (390)</td>
</tr>
</tbody>
</table>
WHAT IS INFORMATION SYSTEMS?

A combination of hardware, software, infrastructure and trained personnel organized to facilitate planning, control, coordination, and decision making in an organization.

A set of inter-related components working together to collect, retrieve, process, store and distribute information in order to facilitate the planning, analysis, control, coordination, and decision making in companies and other organizations.

An integrated set of components for collecting, storing, processing, and communicating information. Business firms, other organizations, and individuals in contemporary society rely on information systems to manage their operations, compete in the marketplace, supply services, and augment personal lives.

The focus of IS

Information Systems (IS) is

• concerned with the information that computer systems can provide to aid a company, non-profit or governmental organization in defining and achieving its goals

• concerned with the processes that an enterprise can implement and improve using information technology

http://computingcareers.acm.org/?page_id=9
WHAT AN IS PROFESSIONAL DOES

IS professionals

- must understand both technical and organizational factors, and must be able to help an organization determine how information and technology-enabled business processes can provide a foundation for superior organizational performance

- serve as a bridge between the technical and management communities within an organization
## CAREERS IN INFORMATION SYSTEMS

<table>
<thead>
<tr>
<th>Information Systems Educators</th>
<th>Systems Designers</th>
<th>Decision Support Systems</th>
<th>Database Managers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems Analysts</td>
<td>Database Administrators</td>
<td>Systems Integrators</td>
<td>Interface Specialists</td>
</tr>
<tr>
<td>Product Support Professionals</td>
<td>Software Engineers</td>
<td>Network Administrators</td>
<td>Network Engineers</td>
</tr>
<tr>
<td>Programmers</td>
<td>Programmer analysts</td>
<td>Information and knowledge officers</td>
<td>Managers of Information Systems</td>
</tr>
<tr>
<td>Telecom Planning and Management</td>
<td>Disaster Planning and Recovery</td>
<td>Technical Writing and Training</td>
<td>Graphical User Interface (GUI) Design</td>
</tr>
<tr>
<td>Data Warehousing and Mining</td>
<td>Application Software Development</td>
<td>Academic Researchers</td>
<td>End-User Computing Support</td>
</tr>
</tbody>
</table>
Undergraduate Student Research
Motivating Factors

An application’s optimal checkpoint interval can be estimated as a function of the dump time and mean time to application interrupt (M). The estimation of these underlying parameters impacts the overall application efficiency and by extension, application throughput. Using discrete-event-driven simulations, we demonstrate the impact of sub-optimal checkpoint intervals on overall system performance and find that relatively large errors in requisite parameter estimates do not significantly degrade average application efficiency.

Simulation Workload

PINK (01/2007 to 01/2008) job state

Job size and usage distributions used in the simulation. These are based on a 1-year trace of jobs on LANL’s PINK cluster. Note that although there are significant numbers of smaller jobs, they do not constitute a large fraction of the total usage.

Efficiency Metrics

Average Weighted Application Efficiency Defined

\[ \text{ArgMax} = \frac{\sum I_{A1} - t_{c1}}{\sum I_{A1}} \]

Example runtime breakdowns where green is the active time and red represents overheads due to checkpoints, restarts, reloads.

Simulation-based results that show the total loss in average application efficiency as a function of error in the estimation of M and end of failure load of system. Note that even when there is an order of magnitude error in M, for SMTTIF > 170, the results are less than 10% lower in application efficiency.

Simulation-based results that show the impact to queue waiting time as a function of error in checkpoint interval assignment. Note that while the previous results suggest minimal impact on application efficiency, small errors in checkpoint interval assignment can have a dramatic impact on queuing delay, and by extension, turnaround time and job throughput.

Theoretical-based results using parameter ranges to demonstrate impact on application efficiency of using checkpointing for one of the DARPA streaming exascale-scale systems. Note that these exa-scale supercomputers are capable systems implying that mission critical codes will be run across the entire platform. With over 3P of RIM, checkpoint dump times will likely be in the several hour range.

Application Monitoring

As the size of HPC systems continues to scale, the mean time between system failures decreases. This trend is exacerbated by the fact that applications fail due to hardware component failures as well as buggy software, and in some cases, an interaction between the two. In order to react to these failures, a root-cause analysis would provide valuable information, however, in many cases, the data necessary for such an analysis is simply not available. At LANL, users have two applications that monitor the computing environment: MoJo and Zeemos. MoJo is an application monitoring program that checks jobs for progress. Zeemos is a system monitor which collects data on system health. Data on failures is gathered by both, yet the databases are separate and only system administrators can access these. These databases need to be combined so that correlations may be drawn. Even when this data is merged, we still may not know why a job has failed.

MoJo

MoJo monitors applications by checking a user-defined measure of progress, such as file output. It provides for customized handling of actions to be taken once a failure has been detected, such as resubmission to the job queue. Users can be notified when MoJo detects stalls or errors, or at any point MoJo’s internal state machine transitions. During operation, it collects data on job outcomes and this data is stored in a centralized database on a per-job basis.

Zeemos

Zeemos monitors systems for hardware-related issues. It collects data based on event type and severity. Again, data is gathered into a centralized database on a per-node basis.

Failure Test Harness

Due to incompatibility between layers in a typical HPC software stack and the heterogeneity among such stacks for disparate systems, reporting of system code, signals, and other valuable error-diagnosing information are often not exported as available to the end user, or in some cases, to system administrators. In order to quickly determine which pieces of this information are reported for a given system, the Failure Test Harness was developed and can be used to correlate systematically generated application errors with the data autonomously gathered using MoJo.
Visualizing the Properties of Carbon-based Molecules
Cameron D. Collins¹, Dr. Steven J. Stuart², Dr. Joshua A. Levine³
Coastal Carolina University¹, Clemson University²-³
Department of Computer Science¹, Department of Chemistry², School of Computing³

Abstract
Most visualization tools used in chemistry are capable of visualizing basic properties such as bond distance, angle, etc., but lack the ability to visualize other more specific properties. The software developed as a result of this project measures chemical and mathematical properties that are exclusive to carbon-based molecules. Processing is a graphics-based programming language that uses Java syntax to build the software. The first step involved reading the file provided and developing a three-dimensional model. Next a Processing library was imported which allows the user to rotate and zoom in on the model. The final step was to develop a user interface which allows the user to select which properties he/she wants displayed, which will color the atoms, bonds, and rings based upon those properties. All of the properties being visualized are difficult to visualize without computer simulation making the use of color necessary. Future work will involve transferring the software to a language that is more well-known in the scientific community, such as Python, and making a more user-friendly graphical interface.

Motivation
Currently there are no software tools to effectively measure the properties that the computational chemists are looking to measure. In order to do this, we are developing a program from scratch using the Processing language designed for visualization.

Methods & Materials
- Processing – a graphical extension of Java
- Used a data-file that consists of 3D coordinate points of a graphite molecule to develop a model
- Proscene – a library for the Processing language that allows the user to manipulate the model.

Results
Figure 1. The first step was to simply develop a 2D model based upon the file that was provided. The reason the first model was 2D was for simplicity purposes to make sure that the file was correctly read in before developing the model.

Figure 2. Once the file was properly read in, changes were made to make the 2D model into a 3D model. To do this, we had to change the ellipses to spheres, and change the size function to handle 3D models.

Figure 3. After developing the 3D model, the program was redesigned to be object-oriented, which allowed the properties to be measured more easily. Rotate, zoom, and pan functions were also added to the program.

Discussion
Processing proved to be a very useful language for scientific visualization. The language is easy to learn and allows the user to get a prototype running quickly. For physics and other fields requiring vector mathematics, there is a vector class built into Processing that makes calculating equations involving vectors much simpler. There is also a lot of tutorials and support for the language as well. Some of the more advanced properties being measured are the: pi-orbital axis vector, pyramidalization angle, and sigma/pi orbital hybridization. The need for a user-interface became more prominent as the project progressed. If too many properties are being visualized at one time, it can be difficult for scientists to distinguish between them.

Conclusions
- Processing is a good language to use for scientific visualization, despite being originally developed for artists.
- It is important to develop a user-interface that allows scientists to choose which visualizations to display.

Future Work
Future work will involve:
- Developing a user-interface to allow the user to select which properties are being displayed.
- Transferring the program to another language such as Python.
- Measuring the ring properties of the molecule.

References
Lowe, C., & Stuart, S. C60 Bombardment of a Graphene Bilayer. .
Stuart, S. (n.p.) Visualization Tool for Carbon-Based Nanostructures. June 1, 2014

Acknowledgements
Charlie Lowe, Department of Chemistry, Clemson University; REU Funded by NSF ACI Award 1359223 Vetria L. Byrd, PI

Technical SOE: Research Experience for Undergraduates in Collaborative Data Visualization Applications • June 2 – July 25, 2014 • Clemson University • Clemson, South Carolina

© 2014 Clemson University; REU Funded by NSF ACI Award 1359223

Clemson University
Clemson, South Carolina
Modernists’ Letters: A Journey into the Modernist Mind
Hali Gallagher, Coastal Carolina University, Conway
Gabriel Hankins, Digital Humanities, Clemson
Jill Gemmill, Clemson Computing and Information Technology

Abstract
This research looks at the letters from the modernist writers F. Scott Fitzgerald, Ernest Hemingway, Virginia Woolf, and Katherine Mansfield. The goal in this project is to make clearer connections between these writers through the use of the visualization tool Palladio. The data were collected by reading and entering metadata into table format, and then exporting it into Palladio. The metadata included the letter ID, author of the letter, letter recipient, date the letter was written, destination place, author place, letter type, where the original letter is archived, and the reference page from secondary source. Palladio can display connections through the creation of a network graph. From the data it was concluded that these writers made connections to other influential writers of their time through a relatively small set of shared connections.

Introduction & Background
This project is part of a collaborative project called Twentieth Century Literary Letters Project. Traditionally, scholars have needed to consult printed volumes for information on specific letters, authors, and social groups. By visualizing this textual data in a form that is searchable and useable, time will be saved on future related projects. This project will add on to the Twentieth Century Literary Letters Project by storing data from letters of the 20th century modernist writers F. Scott Fitzgerald, Ernest Hemingway, Virginia Woolf, and Katherine Mansfield. The collection of letters used for this project were roughly from the years 1918-1922.

Methods & Materials
● Collections of letters gathered in secondary sources were read.
● Massive amounts of data were entered into Microsoft Excel by hand.
● Data were exported into the visualization tool Palladio for further examination.

Acknowledgements
REU Funded by NSF ACI Award 1359223

Results
Figure 1: Network Graph of Hemingway and Fitzgerald Contacts with the use of Palladio

Discussion
Even though Hemingway and Fitzgerald have a few common contacts, those contacts are not written to a lot by both parties. Hemingway’s common contact that he sent the most letters was Gertrude Stein. Fitzgerald’s common contact that he sent the most letters was Maxwell Perkins. Gertrude Stein was known as a profound editor. Mansfield and Woolf’s common contact of interest that both writers sent the most letters to was Samuel Koteliansky. Koteliansky was known as a publisher. This could show all writers were interested in furthering their careers, since their common points of interest are influential people in literary fields.

Conclusions
From Figure 1, it was concluded that the common contacts of interest for Fitzgerald and Hemingway were Sherwood Anderson, Rascue Burton, Zelda Fitzgerald, Edmund Wilson, and Maxwell Perkins. From Figure 2, it was concluded that the common contacts of interest for Woolf and Mansfield were Sydney Waterlow, Strachey Lytton, and Samual Koteliansky. Further investigation into the data shows that Fitzgerald and Hemingway’s common contact of interest that both sent the most letters to was Gertrude Stein; Woolf and Mansfield’s common contact of interest that both sent the most letters to was S.S. Koteliansky, which shows each writers’ interest in communication with influential literary figures of their time.

Study Limitations
Secondary sources varied in standard data. Some sources may have listed only letters the authors sent while other sources would list letters that were sent to and from the author. Some sources would also only list the location of the author and not where the author was sending the letter to.

Future Work
● Continue to edit existing visualizations for further improvements.
● Additional authors will be added to the collection of data.

Selected References
Graduate School
(Looking down the road)

Question and Answer Discussion Panel
Numbers and Bytes Club
Computer Science
Prerequisite Map
Fall 2014 – Spring 2015

MATH 129L
(Arithmetic & Algebra)

To assist you in planning your coursework, this chart illustrates the prerequisites for your Computer Science Foundation and Major courses.

Courses at the 300-level are typically offered in either the Fall or Spring semester, while many 400-level courses are offered alternating years.

Some courses in this category have additional prerequisites. Check the University Catalog for details.
Students from CCU’s Department of Computing Sciences participated in internships and REU’s at the following companies/schools during the Summer of 2015.

- Mathematical Biosciences Institute
- MERCM
- Stinger Ghaffarian Technologies
- Myrtle Waves Water Park
- Charleston Defense Contractors Association
- JVZoo.com
- Rochester Institute of Technology
- WetStone
- Arizona State University
Students from CCU’s Department of Computing Sciences participated in internships and REU’s at the following companies/schools during the Summer of 2014:

- Harsco Rail
- WetStone
- University of Arkansas at Little Rock
- Clemson University
- MERCOM
- JVZoo
- Cisco
- Textron
- USAA
Graduates of CCU’s Department of Computing Sciences have been and/or are currently employed at the following companies:

- WetStone
- Google
- Amazon
- HTC
- Accenture
- SKUTCHI Designs
- JVZoo.com
- Dell
- UNT
- Unitrends
- Coldlight
- Deloitte
- tkacz Engineering, LLC
- L3
- CareCore National
- Santee Cooper
- Los Alamos National Laboratory
- Allstate
- Washington College
- Horry Electric Cooperative, Inc.
- National Security Solutions – E&TS
- Lockheed Martin