Jake Scott Pitkin
160 East 1st Avenue Apt. 305
Salt Lake City, UT 84103
mobile: (801) 209-1994
email: jakepitkin@gmail.com
github: github.com/jspitkin

Work Experience:

University of Utah **August 2016 to Present**
School of Computing CS 2420 - Introduction to Algorithms & Data Structures
Head Teaching Assistant

Working alongside Prof. Miriah Meyer for a second time. Working to improve this fundamental course that is the beginning of most students theoretical computer science education. Creating new assignments, refining course goals, and providing stronger feedback to students.

Autodesk - Delcam **May 2016 to August 2016**
Salt Lake City, Utah
Software Engineer Intern

Worked with a C/C++ legacy code base of CAD/CAM software. Updated existing C/C++ code to C# to improve maintainability and address memory issues. Worked with the main projects API to write external software that generates HTML and XLS documents that detail machine setup for part manufacturing.

University of Utah **January 2016 to May 2016**
School of Computing CS 2420 - Introduction to Algorithms & Data Structures
Head Teaching Assistant

Teaching first-year computer science undergraduates about the basics of algorithms and data structures using Java. I run weekly labs, have weekly office hours where students can get help and proctor exams. Additionally, I coordinate the other teaching assistants by making grading assignments and rubrics.

University of Utah **August 2015 to December 2015**
School of Computing CS 1410 - Object-Oriented Programming
Head Teaching Assistant

Helped teach first-year computer science undergraduates using C++ and Arduino kits. I guided students through labs where we build simple projects with the Arduino kits. Additionally, I designed grading tests and rubrics to evaluate their work and coordinate the other teaching assistants.

University of Utah **January 2015 to April 2015**
School of Computing CS 3500 - Software Practice I
Teaching Assistant

Ran lab sessions teaching students about proper software practices using C# and Visual Studio. Helped students build a desktop spreadsheet program and an online multiplayer Boggle clone using a RESTful API. Designed grading rubrics and organized the grading of student's code and exams.

Skills:

Languages:	C/C++, Python, Swift, Java, C#, JavaScript, PHP
Operating Systems:	OS X, Linux, Windows
Applications:	Microsoft Word, Excel, PowerPoint, Visual Studio, Eclipse
Markup:	LaTeX, HTML, XML
Other:	Unix Shell, Git

Relevant Projects:

Malware Proxy - In Python, created a proxy that checks a web response for malicious content. A trusted database of known malicious content is queried and the content is blocked if something malicious is detected.

Secure Messaging **-** Created a chat client to explore network security. Using the PyCrypto library along with the SHA-1 algorithm, RSA keys, and the 3DES algorithm. A robust message can be sent over a network, safe from malicious attacks that attempt to modify or read the secure message.

Asynchronous Socket - A non-blocking socket object that encapsulates connecting to a server. Written in C++ with Boost libraries. Built to help abstract away the details of connecting to a server. Built for Dr. Peter Jensen during Spring 2015 to assist in his grading of the final projects for Software Practice II.

Multicircle - Browser-based game where two players compete to control all the circles on the map. Built in a team of three using JavaScript and Node.js. Allows players to join a game with a random player or to make a game lobby their friend can join.

TA Application - Web-based application where users can apply to be a teaching assistant or manage applications if they are an admin. Developed with HTML, CSS, JavaScript and PHP. Built to practice web scraping, input validation, password hashing, Ajax, responsive UI's and other web techniques.

Boggle – Created a client GUI and server that allows two players to play "Boggle" over a network using sockets. The results of finished games are stored to a MySQL database and then presented on a website hosted on the server.

Spreadsheet - Desktop spreadsheet program written in C#. Included a formula parser, a cell dependency graph and a UI. In a team of four, a server was implemented so multiple users, from different machines, could edit spreadsheets over a network. The spreadsheets were saved in a MySQL database.

Education:

B.S./M.S. in Computer Science **August 2013 - Spring 2018**
University of Utah, Salt Lake City, UT **GPA: 3.76/4.0**
Dean's List: Fall 2013, Spring 2014, Fall 2014, Spring 2015, Fall 2015