**Jun (Chris) Pak**

430 Kelton Ave. Los Angeles, CA 90024 | 818.521.7654 | [jpak1996@g.ucla.edu](mailto:jpak1996@g.ucla.edu) [linkedin.com/in/chris-pak](https://www.linkedin.com/in/chris-pak) | [github.com/jpak1996](https://github.com/jpak1996)

Education

**University of California, Los Angeles** *Expected Graduation: Dec. 2018*

* Bachelor’s in Computer Science, Class of 2018
* 3.22 GPA across core Computer Science classes
* Member of ACM, a community dedicated to running hacking events that taught students how to use Javascript to build a spell-checking program, and how to build their own website
* Actively help students by participating in online Piazza discussions regarding current class projects and homework

Projects

* **SmashTime**:
* Web-app utilized by school-wide gaming tournaments to send out text messages to players based on real-time queries (Python/Django/SQLite, Javascript/Node; Challonge, Twilio)
* **Personal webpage**:
* Organizational tool for personal accomplishments and future goals (HTML/CSS, React/Redux, MAMP; WordPress, Bootstrap)
* Preferred languages: Python, Javascript, C++

Work Experience

**IT Student Consultant** - *UCLA Luskin IT Help Desk (Jan. 2017 – present)*

* Assisting graduate students with remote login, network issues, and email
* Managing Windows/Mac hardware and software installation for lab users
* Troubleshooting professors’ laptops before and during classes/presentations to ensure a smooth user experience

**Content Editor** - *UCLA Course Reader Solutions (Sept. 2016 – Jan. 2017)*

* Assisted walk-in requisitions from over 300 UCLA professors in a fast-paced work environment
* Edited/proofread readings to be uploaded for online purchase adhering to professors’ deadlines
* Contacted publishers such as Harvard Business and Penguin Random House for copyright permissions
* Managed the digital database of 14,000+ course readers and kept record of old/new requisitions

Relevant Coursework

**CS 31**: C/C++, pointers, arrays, linked lists:

* Built a quick word-search program using vectors, later optimized by instead using a binary search tree

**CS 32**: Algorithms and data structures:

* Coded a fully functional game by implementing operator overloading and interpreting keyboard inputs for a simple user interface

**CS 33**: Assembly language, I/O programming, computer architecture and memory management:

* Used GDB to successfully stack-smash an open-source webserver

**CS 35 Lab**: Open-source software tools (Git, Emacs/Vim, Python) and concurrent/parallel code:

* Parallelized a simple ray-tracing program to speed it up by a factor of ~10
* Used GNU Privacy Guard to defend against remote login from another team using asymmetric cryptography

**CS 111**: Operating systems, memory virtualization, scheduling/synchronization

* Improved shell scripting commands by implementing optional flags and file modes using multiple parent/child processes