Brett Gurman

brett.gurman@tufts.edu • (203) 803-8395 • 44 Emery Street, Medford MA 02155

EDUCATION:

Tufts University, Medford, MA **GPA: 3.33**

- Bachelors of Science in Computer Science and Cognitive and Brain Sciences, expected May 2017

WORK AND RESEARCH EXPERIENCE:

Tufts Computer Science Department | Medford, MA

Teaching Assistant | September 2014 - Present

- Grade assignments for Data Structures and Introduction to Computer Science
- Teach computer laboratories for intermediate programming students
- Guide students in completion of major course assignments during office hours

JoMI (Boston Startup) | Boston, MA

IT Engineer | June 2015 - August 2015

- Created web content for a video-based medical journal startup company

Tufts Human-Robot Interaction Laboratory | Medford, MA (Comp11, Comp15, Comp40)

Research Assistant | November 2014 - August 2015

- Implemented methods to help incorporate a physics simulator into a decision making algorithm for use with robots

EXTRACURRICULAR ACTIVITIES:

Tufts Varsity Swimming and Diving Team | September 2013 - Present

- Attend daily 4-hour practices and weekly meets

SKILLS:

Programming Languages:	C, Python, C++, Assembly Language, Javascript, HTML, CSS, Scheme, Standard ML, Java, Bash, SQL, R,
Software:	Microsoft Office, GNU, Git, Linux, Heroku
Foreign Languages:	Chinese
Relevant Coursework:	Machine Structure and Assembly Language, Programming Languages, Computer Graphics, Data Structures, Web Programming, Computation Theory, Operating Systems, Natural Language Processing (Spring 2016), Human Computer Interaction (Spring 2016), Algorithms (Spring 2016)

EXAMPLE WORK:

Lost and Found App | Fall 2015 (Tufts Fall 2015 Polyhack Hackathon)

- Developed a flask- and postgresql-based web app that allows finders of lost items to advertise their finds and helps owners of lost items to recover their belongings

Universal Machine | Fall 2014

- Designed and implemented a 32-bit segmented memory Universal Machine in C, utilizing several high-level abstraction techniques from the Hanson C library, as well as lower level algorithms like bit-packing