

# TWC: Small: Empirical Evaluation of the Usability and Security Implications of Application Programming Interface Design

PIs: Brad A. Myers<sup>1</sup>, Sam Weber<sup>2</sup>, and Robert Seacord<sup>3</sup>

Contributing researchers: Michael Coblenz<sup>4</sup>, Whitney Nelson<sup>5</sup>, Jonathan Aldrich<sup>6</sup>, Joshua Sunshine<sup>6</sup>

Carnegie Mellon University

<sup>1</sup>Human Computer Interaction Institute, <sup>2</sup>New York University, <sup>3</sup>NCC Group, <sup>4</sup>Computer Science Department, <sup>5</sup>REU summer student, <sup>6</sup>Institute for Software Research

## Challenge:

- Develop and empirically test concrete and actionable API and programming language design principles that lead to more secure code.

**Glacier:**

Person class is immutable

```
class PersonHeight {
    int feet;
    int inches;
}

@Immutable public class Person {
    String name;
    PersonHeight height;

    public void setName(String name) {
        this.name = name;
    }
}
```

Error: can't include mutable object in immutable class

Error: can't assign to field of immutable class

## Scientific Impact:

- Cilk Plus's mechanism for defining reducers is more usable than OpenMP's, and has a more familiar syntax.
- Characterized many different ways to restrict changes, and identified that programmers needs are not met by today's systems.
- Created Glacier – annotations for Java that enforce immutability.
- DSL for Blockchain programming will provide usable verification of key correctness properties, and reasoning about resource usage.

	final	Glacier
Users who made errors enforcing immutability (after all tasks)	10/10	0/10

## Solution:

- Evaluate usability of **OpenMP** and **Cilk Plus** proposed parallelism extensions to C and C++
- Understand and provide better immutability features
- Domain-specific language (DSL) for programming Blockchain programs

**Glacier:** Annotation system for Java which **statically** enforces **transitive class immutability**.

- User study showed works better than final: prevents real-world bugs and security vulnerabilities; usable with minimal training

## Broader Impact:

- All programmers will better understand and write better security-relevant code.
- Show how designers can better take into account and test usability of programming and API features.