Concepts of programming languages

Language XXXX

The authors

What problems does Rust (intent to) solve?

- Memory safety
- “Fearless” concurrency
- Performance

Memory safety

Null pointers

- Easy to forget
- The Option enum (similar to Maybe in Haskell)
- The Result enum

Memory safety

Null pointers

```
enum Option<T> {
    Some(T),
    None,
}
enum Result<T, E> {
 Ok(T),
 Err(E),
}
```

Memory safety

Dangling references

- No garbage collector!
- Borrowing rules/Lifetimes

Memory safety

Buffer overruns

- Safety
- Index in array
- Compile/Runtime checks

“Fearless” concurrency

Borrowing rules

- Only one owner
- No aliassing
- Easier debugging

“Fearless” concurrency

Message passing

```
let (tx, rx) = mpsc::channel();

thread::spawn(move || {
    let val = 5;
    tx.send(val).unwrap();
});

let received = rx.recv().unwrap();
```

“Fearless” concurrency

Shared state

```
let m = Mutex::new(0);

thread::spawn(move || {
    let mut num = m.lock().unwrap();
    *num = 5;
});
```

Performance

- No garbage collector
- Fewer run time checks