General Meeting - Week 4

acm @OSU

Programming Language Paradigms

Leo Sallos

Online Officer

What are Programming Paradigms?

acm @OSU

- Ways of representing programming logic on a high level
- Set of instructions vs set of logical statements

```
addi t1, zero, 10
LOOP:
brge t0, t1, ENDLOOP
addi t0, t0, 1
jal zero, LOOP
ENDLOOP:
```



```
while (i < 10) {
    i += 1;
}
```

Imperative Programming Languages

acm @OSU

- Probably the first paradigm you learned
- Describes the program step-by-step (procedural programming)
- Utilizes state to determine behavior
- Has assignments, loops, conditional branching, jumps (goto)
- Fortran was the first commercially successful high-level language in 1954
- C was developed for UNIX, and then became the default for the industry

```
int a = 12;
for (int i = 0; i < a; ++i) {
    f(a);
}
```

Object Oriented Languages

acm @OSU

- Imperative, but with objects
- Most popular paradigm today
- Idea of encapsulating data and logic into objects containing fields and methods
- Implements inheritance and subtype polymorphism
  - A parent class can have common fields and methods with its subclasses









```
class SecretNum {
    int val = 0;

    public:
    void inc() {
        val += 1;
    }

    int get() {
        return val;
    }
};
```

Functional Languages

acm @OSU

- Form of declarative programming
  - Describes logic without describing the control flow
- No side effects! The state is only transformed by functions
- Built from lambda calculus (equivalent to Turing machines)
- Lisp was the first in 1960
- Many functional features were brought into other languages like Python and Rust.

$$\lambda x.M$$

function . body







elixir

