GoBackwards

Go, but worse.
Much worse.

Shaquan Nelson (sdn2115)
Julian Silerio (jjs2245)
Peter Richards (pfr2109)
What is GoBackwards?

“Go is an open source programming language that makes it easy to build simple, reliable, and efficient software.”
What is GoBackwards?

GoBackwards is
dumber simpler!

The GoBackwards
Gopher

???
Language Comparison

“Hello World” in Go

```go
package main
import "fmt"

func main() {
    fmt.Println("hello world")
}
```

“Hello World” in GoBackwards

```go
func main() {
    println("hello world");
}
```
Language Comparison

“Fibonacci Sequence” in Go

```go
package main

import "fmt"

// fib returns a function that returns successive Fibonacci numbers.
func fib() func() int {
    a, b := 0, 1
    return func() int {
        a, b = b, a+b
        return a
    }
}

func main() {
    f := fib()
    fmt.Println(f(), f(), f(), f(), f())
}
```

“Fibonacci Sequence” in GoBackwards

```go
func fib(x int) {
    if (x < 2) return 1;
    return fib(x-1) + fib(x-2);
}

func main() {
    var x int;
    for(x=0; x<6; x = x+1) {
        print(fib(x));
    }
    return 0;
}
```
Language Tutorial

● Declarations
  ○ Variables and Types
    ■ Integers  var x int;
    ■ Booleans  var x bool;
    ■ Strings   var x string;
  ○ Arrays
    ■  var x[5] int;
    ■  x[3] = 3;
  ○ Functions
    ■  Built in Main Function  func main{}()
    ■  Helper functions
      func id( id type, id type,...){ } 

● Expressions
  ○ Literals
    ■ String  “Hello World!”
    ■ Number  42
    ■ Boolean True | false
  ○ Assignment
    ■ Declaration
      var x int;  x= 10;
  ○ Arithmetic Operators
    ■ +,-,*,/  print(30 / 15);
  ○ Built in Function Calls
    ■ Call Ascii ascii(“star.png”);
    ■ Call Print print(34);
Language Tutorial

- Declarations
  - Variables and Types
    - Integers: `var x int;`
    - Booleans: `var x bool;`
    - Strings: `var x string;`
  - Arrays
    - `var x[5] int;`
    - `x[3] = 3;`
  - Functions
    - Built in Main Function: `func main(){}`
    - Helper functions: `func id( id type, id type,...){ }`
Compiler pipeline

- **input_program.gob**: Top level of compiler, begin checking
- **gobackwards.ml**: Scan program for tokens
- **scanner.ml**: Parse tokens using context-free grammar
- **parser.mly**: Translation of code into LLVM-IR
- **codegen.ml**: Semantic type checking of program
- **semant.ml**: Check syntax of parsed input
- **output**:
Example program

```golang
/* explicit types of parameters and return value */
func helloworld_helper(x string) string{
    return(x);
}

/* multiple parameters and expressions in returns */
func add(x int, y int) int {
    return x + y;
}

/* main method is needed to run any gobackwards function */
func main(){
    /* implicit declarations */
    var s1 string;
    var i1 int;
    var b1 bool;

    /* explicit declarations */
    var s2 string;
    var i2 int;
    var b2 bool;

    /* like in C, we need to allocate storage before declaring variables */
    s2 = "hi";
    i2 = 10;
    b2 = false;

    println(s1);
    println(s2);
    print(i1);
    print(i2);
    printb(b1);
    printb(b2);

    /* calling helper functions */
    print(add(3,4));
    println(helloworld_helper("Hello, World!"));

    /* calling built in ascii function */
    ascii("star.png");
```
Testing

Test-Driven Development

Edge Case Testing:
making sure the user knew what was and was not allowed

Automation:
./test.sh
./exe.sh -filepath
Conclusion

● Major goals
  1. Make a Language Similar to Go
  2. Remove Some capabilities of Go to Make A More C-like Language
  3. Add picture-to-ASCII conversion capabilities to Go

● Success?
  ○ Successful implementation of basic Go syntax with C syntactic sugar
  ○ ASCII function is dependent on external C library
  ○ Overall successful group project despite tough road with many roadbumps