# COEN 175

Phase 6 - Week 2

# TAs

- Chris Desiniotis: cdesiniotis@scu.edu
  - Office Hours: Friday 12 - 2 pm
- Antonio Gigliotti: agigliotti@scu.edu
  - Office Hours: Thursday 11 - 1 pm

# Extra Help/Tutoring

- Tau Beta Pi Tutoring
- Link to Tutoring schedule
  - https://sites.google.com/scu.edu/scutaubetapi/tutoring?authuser=1&pli=1

# Phase 6 - Code Generation

1. Address and Dereference
2. Finish Assignment
3. Write Label class
4. Strings
5. Return statement
6. Finish the rest of the compiler (rest of statements)

- **Due Friday March 12, 11:59PM**

# 1. Address and Dereference

- Address::generate()
  - Remember, &*E == E
  - Use the helper function isDereference() for this case
- Dereference::generate()
  - Depending on size (1 or 4 bytes), use ‘movl’ or ‘movsbl’

```
void Address::generate()
{
    Expression *pointer;

    if (_expr->isDereference(pointer)) {
        pointer->generate();

        if (pointer->_register == nullptr)
            load(pointer, getreg());

        assign(this, pointer->_register);

    } else {
        assign(this, getreg());
        // leal _expr, this;
    }
}
```

```
void Dereference::generate()
{
    _expr->generate();

    if (_expr->_register == nullptr)
        load(_expr, getreg());

    // movl (_expr), _expr
    // OR
    // movsbl (_expr), _expr

    assign(this, _expr->_register);
}
```

# 2. Finish Assignment

- Modify Assignment::generate() to handle when LHS is a dereference

```
void Assignment::generate()
{
    Expression *pointer;

    _right->generate();

    if(_left->isDereference(pointer))
    {
        pointer->generate();

        // load pointer
        // load right
        // ...
        // Don't forget to account for the byte case
    }
    else
    {
        // case for Assignment that you should have already written

        // load right
        // ...
        // Don't forget to account for the byte case
    }
}
```

# 3. Write Label class

- Write Label.cpp and Label.h
- Check lecture notes for code

```
class Label {
    static unsigned _counter;
    unsigned _number;

public:
    Label();
    unsigned number() const;
};

ostream &operator <<(ostream &ostr, const Label &label);
```

# 4. Strings

- All strings belong in **.data** section of assembly file
  - Suggested approach: Store all strings in a global map data structure so we can easily refer to the strings by Label later on. Allocate storage for strings at the end of the assembly file
- Modify String::operand and generateGlobals() to accommodate strings
- Create global map from string to label
- String::operand()
  - Check if string exists in map already
    - If not, create a label and add to map
  - Write the label to the ostr argument
- generateGlobals()
  - Print out assembly for all strings in the map

```
# Assembly file
.
.
.

        .data
.L10:       .asciz      “hello”
.L15:       .asciz      “coen175”
```

# 5. Return statement

- Generate return value and load into eax
- Jump to return label
  - Hint: the name of the current function is a global variable declared at top of generator.cpp
- Free register for _expr

# 6. Finish the rest of the compiler (rest of statements)

- Write Expression::test() **first**
  - Check notes for this
- Suggested order from here on out:
  - LogicalAnd, LogicalOr
  - While (check notes)
  - If
  - For
  - More test() functions
    - Logical Expressions
    - Comparative Expressions
    - Not
      - _expr->test(label, !ifTrue);