

Plan for Today

Lvalue versus Rvalue

- summary of semantics with respect to 3-address code
- example and how to generate code for locals
- Assem(MIPS) translations for lw and sw

Integer expressions

- binary ops, constants, and identifiers

How everything works together

Testing script and test cases for integer expressions

Lvalues and Rvalues in 3-address code

Addresses in 3-address code can be:

- names (think variable names)
 - lvalue(name) = address of name
 - rvalue(name) = value stored at lvalue(name)
- constants
 - lvalue(constant) => constants may or may not have a location
 - rvalue(constant) = value of constant
- compiler-generated temporaries
 - lvalue(temp) = register temp will eventually be assigned to
 - rvalue(temp) = value stored in temp

Example where names are local variables

```
public int foo ( int p ) {
    int j;
    int i;
    i = 32;
    j = 7;
    i = j;
    p = p;

    return p;
}
```

Integer Expressions

```
Expression ::= Expression ("+" | "-" | "*" ) Expression
            | <INTEGER_LITERAL>
            | Identifier
```

AST nodes

- PlusExp
- MinusExp
- MulExp
- IdExp
- IntegerExp

Array Expressions

Expression ::= Expression "[" Expression "]"
| Expression "." "length"