Project related: Let Binding

Variable Binding

In homework 2, you added variables to MSDscript

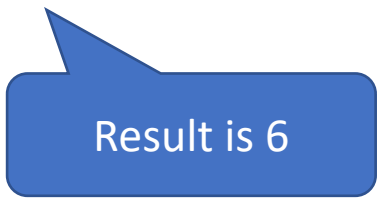
But we don’t yet have a way to give a variable a value

Next: add a declaration form called **_let**

In general, MSDscript keywords will start with an underscore

Let Binding

```
_let x = 5
_in x + 1
_
```



Let Binding

```
_let x = 5
_in x + 1
```

Similar to

```
{
  int x = 5;
  x + 1;
}
```

because **x** is visible only in the **_in** part...

Let Binding

```
_let x = 5
_in x + 1
```

... but just one expression must be after _in...

Let Binding

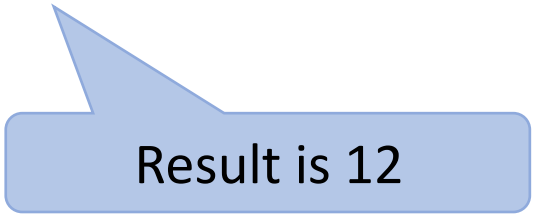
```
_let x = 5
_in x + 1
_
```

... and the whole thing is still an expression

Let Binding

```
_let x = 5
_in x + 1
```

```
(_let x = 5
 _in x + 1) * 2
```



Let Binding

```
_let x = 5
_in x + 1

(_let x = 5
 _in x + 1) * 2

_let x = 5
 _in x + 1 * 2
```

Result is 7

Let Grammar

⟨expr⟩ = ⟨number⟩
| ⟨expr⟩ **+** ⟨expr⟩
| ⟨expr⟩ ***** ⟨expr⟩
| ⟨variable⟩
| **_let** ⟨variable⟩ **=** ⟨expr⟩ **_in** ⟨expr⟩

Right-hand side RHS

body

Interpreting _let

```
_let x = 5
_in  x + 1

⇒

5 + 1

⇒

6
```

Interpreting **_let** can use **subst**

```
(new Add(new Var("x"), new Num(1)))
 ->subst("x", new Num(5))
 ->interp()
```

Interpreting _let

```
_let x = 5 + 2 -> 7
_in  x + 1

⇒

7 + 1

⇒

8
```

Interpret binding **RHS** before **subst**

```
int n = (new Add(new Num(5), new Num(2)))
        ->interp();
```

```
new Num(n)
```

Nested Let Binding: subst

```
_let x = 5
_in  _let x = 6
     _in  x + 1
```

Means 6, not 5

Nested Let Binding: subst

```
_let x = 5
_in  _let x = 6
     _in  x + 1
```

Analogous to

```
{
  int x  =  5;
  {
      int x  = 6;
      x + 1;
  }
}
```

Nested Let Binding: subst

```
_let x = 5
_in   _let x = 6
      _in  x + 1
```

Substitute x with 5 in

```
_let x = 6
_in  x + 1
```

should not change

```
_let x = 6
_in  x + 1
```

`6 + 1`

Nested Let Binding: subst

```
_let x = 5
_in   _let y = 6
      _in  x + 1
```

Substitute x with 5 in

```
_let y = 6
_in  x + 1
```

should change to

```
_let y = 6
_in  5 + 1
```

Substitution of ⟨variable⟩ with ⟨expr⟩ at **_let**:

- bind *same* ⟨variable⟩: don't substitute in the *body*
- bind *different* ⟨variable⟩: substitute in the *body*

Nested Let Binding: RHS

```
_let x = 5
_in   _let x = x + 2
      _in   x + 1
```

Substitution of ⟨variable⟩ with ⟨expr⟩ at **_let**:

- always substitute in the right-hand side