Discussion Topics

- Working with static method and attributes
- Object Instantiation
Working with classes in programs

Classes, when instantiated, become objects.
Not all classes need to be instantiated to be used within programs.

- Static attributes (CLASS-DATA)
- Static methods (CLASS-METHODS)

To reference a static public attribute:

```plaintext
classname=>attributename
myvar = classname=>attributename
```

To call a static method:

```plaintext
CALL METHOD classname=>methodname.  "traditional
classname=>methodname( ).  "as of Web AS 6.10
```

Static vs. instance: Static uses => Instance uses -
Other syntax elements are the same.

---

Calling methods

Cannot be a space before opening parenthesis. Must be one after.
If only one importing parameter in method definition, list the actual
parameter in the parenthesis by listing data object name only.

```plaintext
class=>method( data1 ).
```

If several importing parameters, parameters must be listed in pairs:

```plaintext
formal parameter = actual parameter.
class=>method( par1 = data1 par2 = data2 ).
```

If there are parameters of types other than importing, the type of
parameter passing must be listed.

```plaintext
class=>method( EXPORTING p1 = d1 p2 = d2
IMPORTING par3 = data3
CHANGING par4 = data4 ).
```
Calling methods

Functional methods (those that return a value)
The value to be returned by be listed in the method call interface using the parameter passing type RECEIVING.

```ruby
class=>method( IMPORTING p1 = d1 RECEIVING par4 = data4 ).
```

The value to be returned may be captured through assignment (more typical).

```ruby
data3 = class=>method( d1 ).
```

Calling methods

Parameter passing types defined in the method definition correspond to parameter passing types in the caller but are not the same.

<table>
<thead>
<tr>
<th>Method Definition</th>
<th>Method Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPORTING</td>
<td>EXPORTING</td>
</tr>
<tr>
<td>EXPORTING</td>
<td>IMPORTING</td>
</tr>
<tr>
<td>CHANGING</td>
<td>CHANGING</td>
</tr>
<tr>
<td>RETURNING</td>
<td>RECEIVING</td>
</tr>
</tbody>
</table>

I am exporting a value to you

I am importing a value from you

Method Call

Method Definition
Let's Practice

Static Rules

Static methods can only access static attributes.
Static method may define local data objects.
Instance methods can access static and instance attributes.
Object Creation

Objects are created by defining an object reference variable and then creating the object.

The **START-OF-SELECTION** keyword indicates where the runtime environment should begin control flow.

```
* Object definitions here
START-OF-SELECTION.
DATA objectname TYPE REF TO classname.
CREATE OBJECT objectname.
```