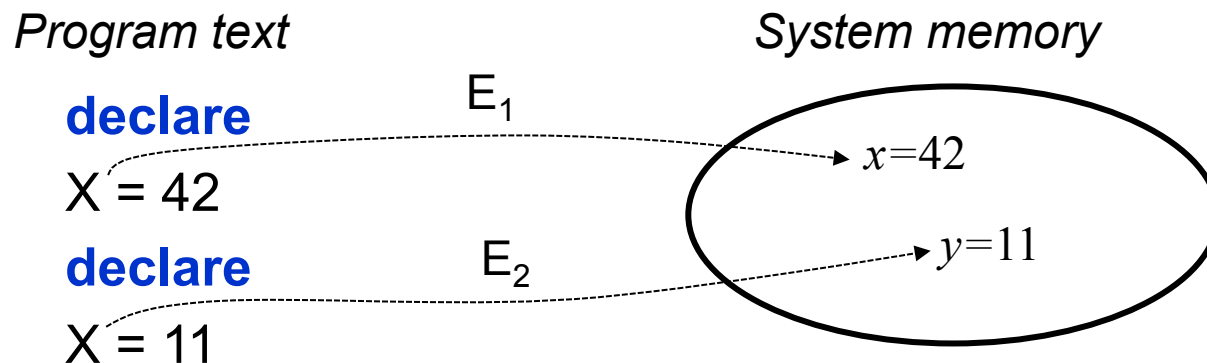


# Redeclaring an identifier



- An identifier can be **redeclared**
  - The same identifier refers to a different value
  - There is no conflict with single assignment.  
Each occurrence of X corresponds to a different variable.
- The interactive environment always has the last declaration
  - **declare** keeps the same correspondance until redeclared (if ever)
  - In this example X will refer to 11

# Scope of an identifier occurrence



```
local
  X
in
  X = 42 {Browse X}
local
  X
in
  X = 11 {Browse X}
end
{Browse X}
end
```

- The instruction **local X in <stmt> end** declares X between **in** and **end**
- The **scope** of an identifier occurrence is that part of the program text for which the occurrence corresponds to the same variable declaration
- The scope can be determined by **inspecting the program text**; no execution is needed. This is called **lexical scoping** or **static scoping**.
- Why is there no conflict between X=42 and X=11, even though variables are single assignment?
- What will the third Browse display?