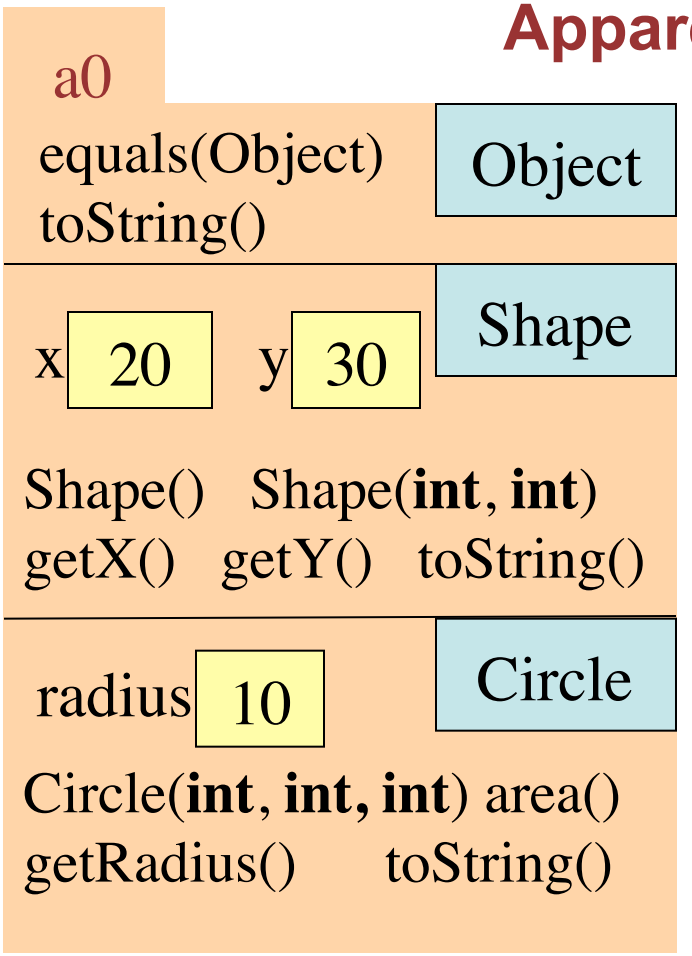
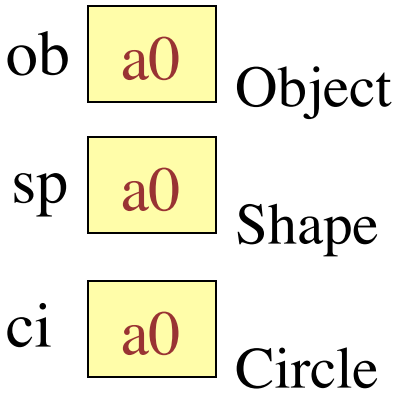
Apparent class



Object ob;

Shape sp;

Circle ci;



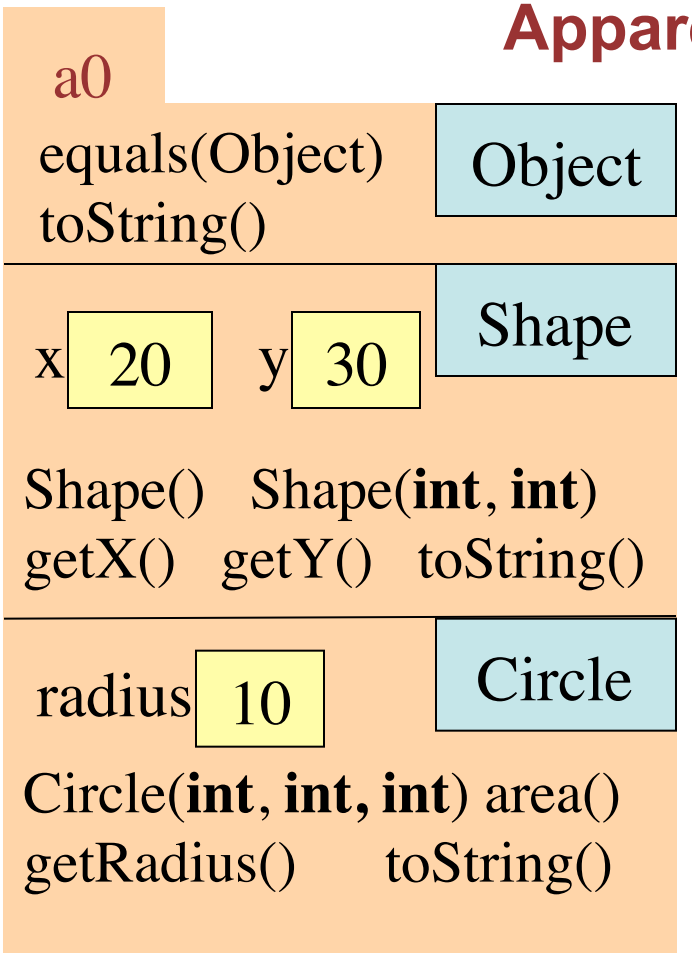
Apparent class of a variable: The class with which it is defined.

Apparent: (1) clearly seen or understood.

(2) appearing to show particular qualities or attributes that may not be genuine.

Apparently, based on its declaration, sp contains an object of class Shape.

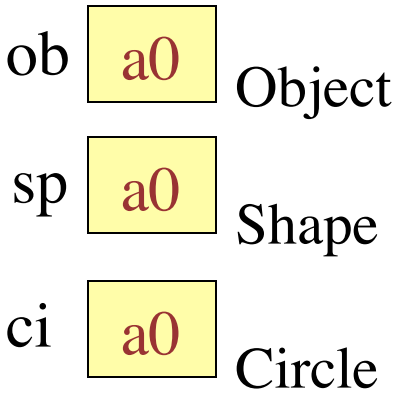
Apparent class



Object ob;

Shape sp;

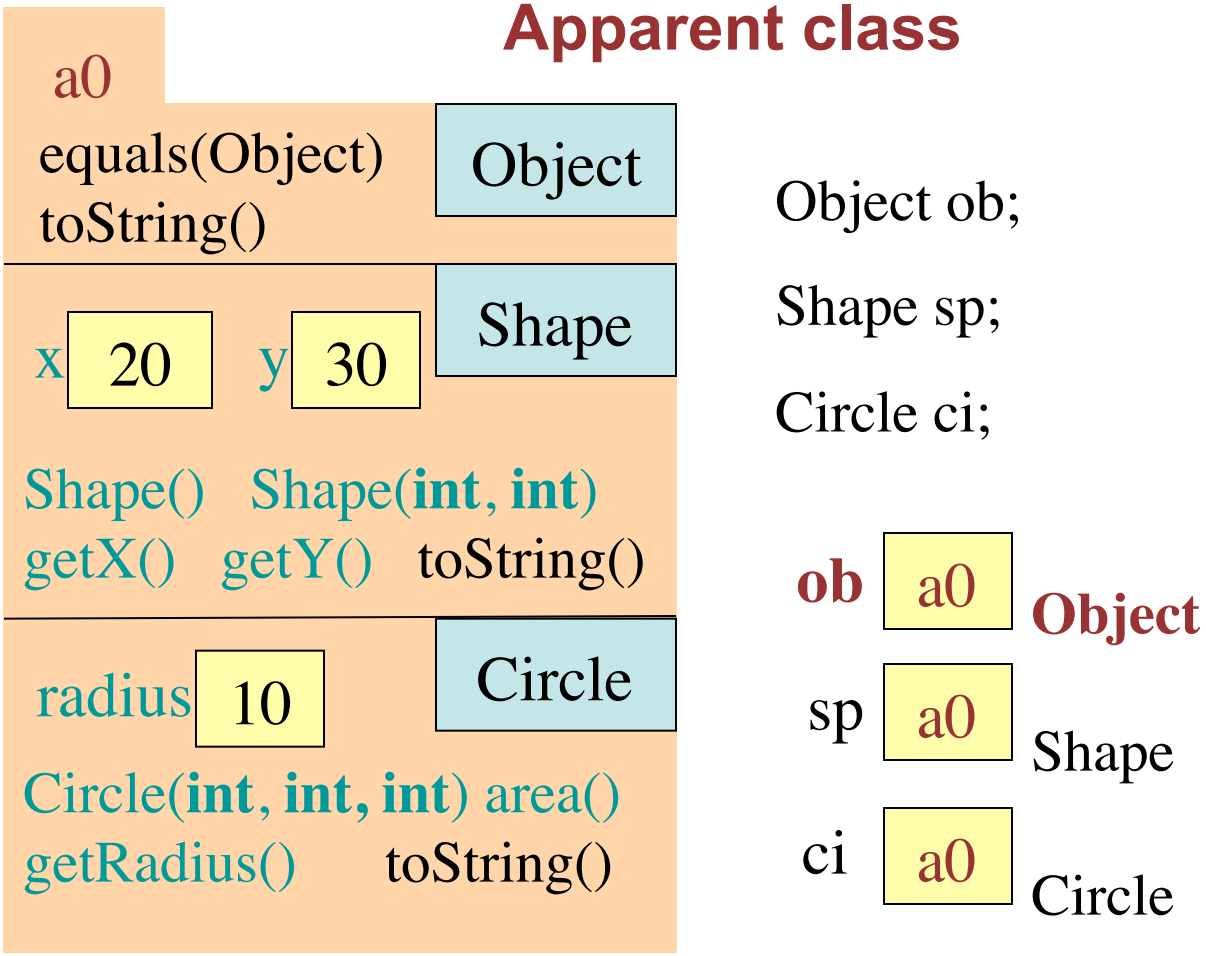
Circle ci;



Apparent class: Class with which variable is defined.

Apparent class: a syntactic property. It determines what components of the object can legally be referenced.

Apparent class

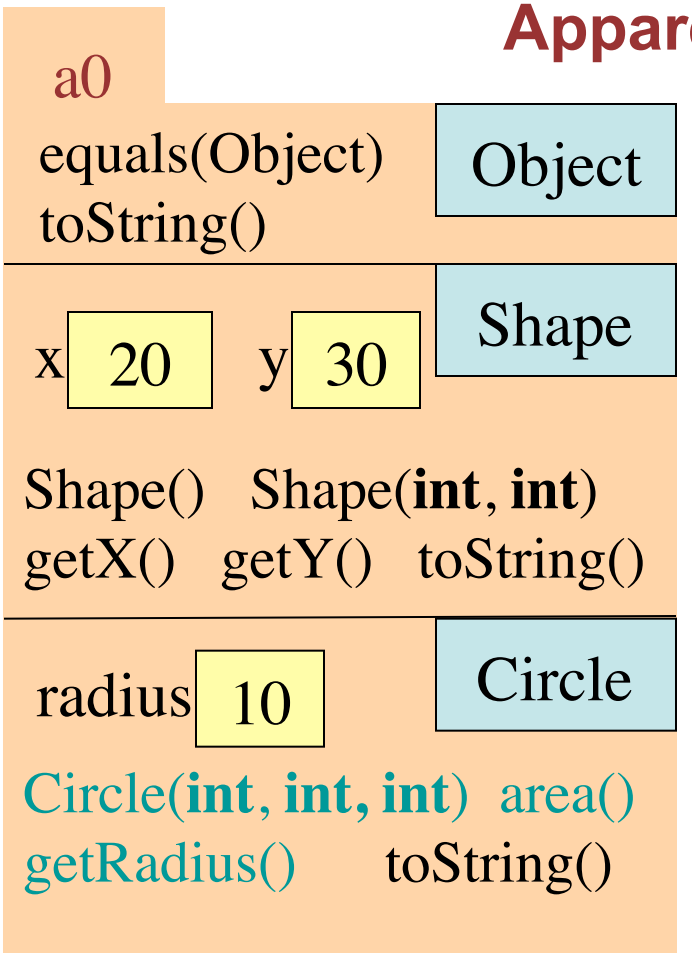


Apparent class: a syntactic property. It determines what components of the object can legally be referenced.

Legal: ob.equals(…) ob.toString()

Illegal: ob.x ob.y ob.getX() ob.area()
ob.getRadius()

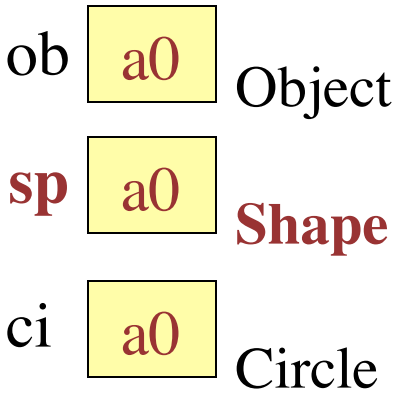
Apparent class



Object ob;

Shape sp;

Circle ci;

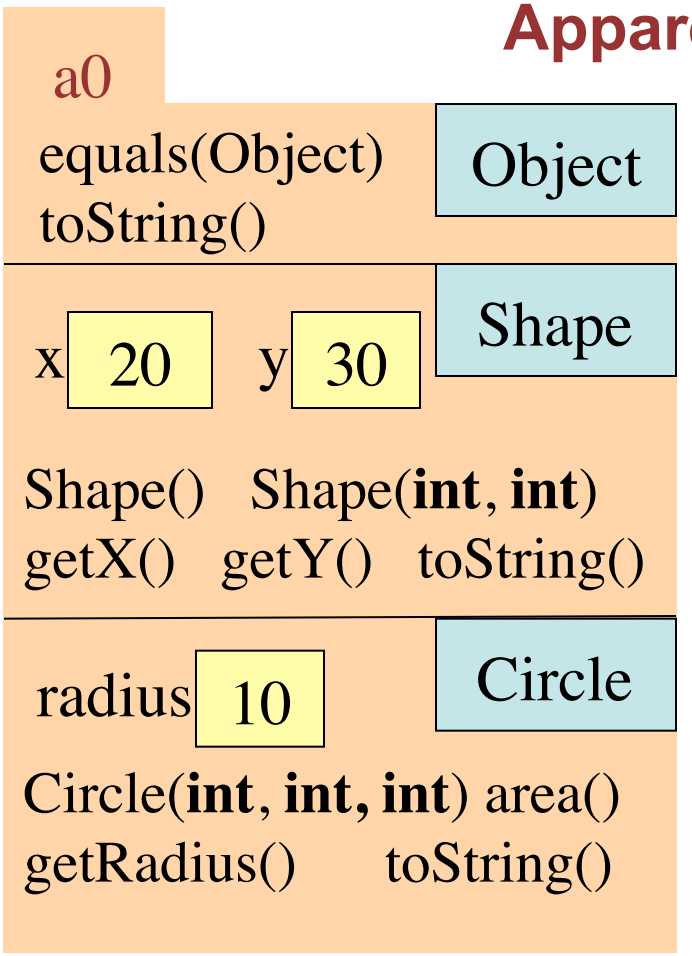


Apparent class: a syntactic property. It determines what components of the object can legally be referenced.

Legal: sp.equals(…) sp.toString() sp.x sp.y
sp.getX() sp.getY()

Illegal: sp.area() sp.getRadius()

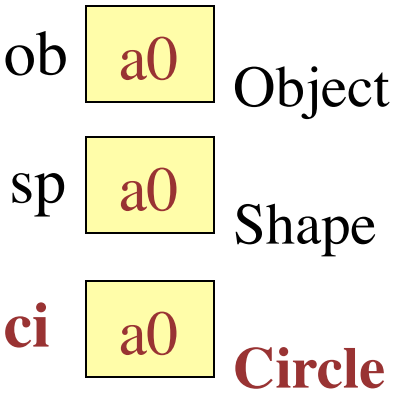
Apparent class



Object ob;

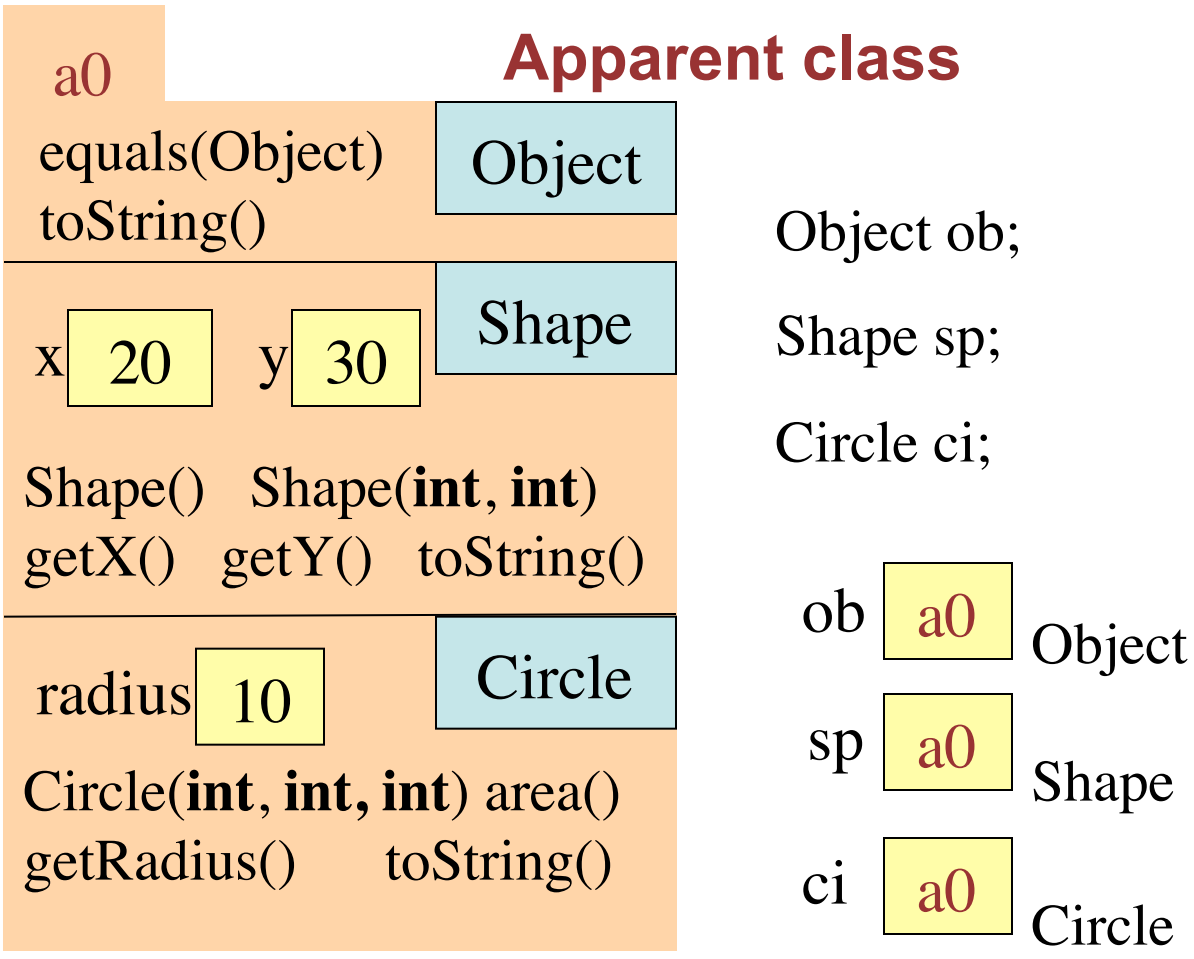
Shape sp;

Circle ci;



Apparent class: a syntactic property. It determines what components of the object can legally be referenced.

Legal: ci.equals(…) ci.toString() ci.x ci.y
ci.getX() ci.getY()
ci.area() ci.getRadius

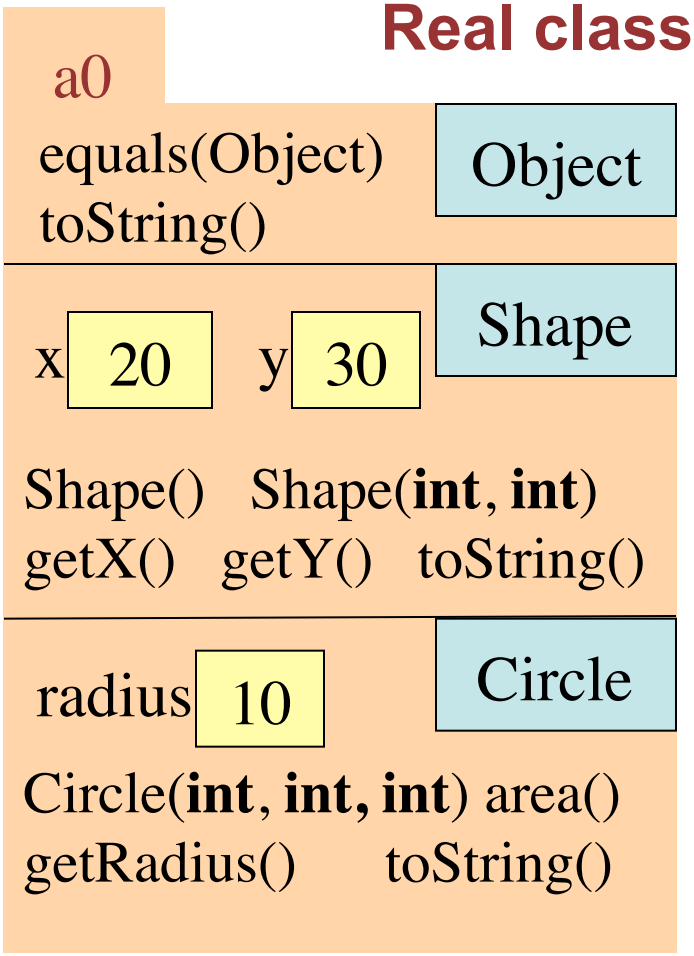


Apparent class: Class with which variable is defined.

Apparent class: a syntactic property. It determines what components of the object can legally be referenced.

Rule: For a variable x of some class-type C, the only legal references of the form x.*variable* or x.*method-call* are to variables and methods defined in or inherited by class C.

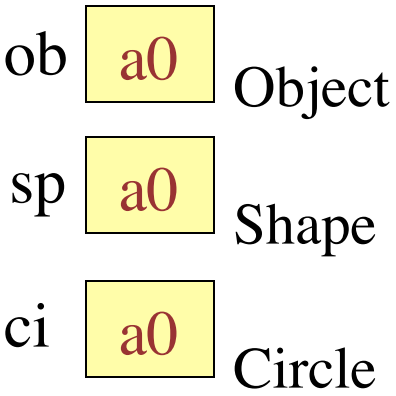
Real class of a variable



Object ob;

Shape sp;

Circle ci;

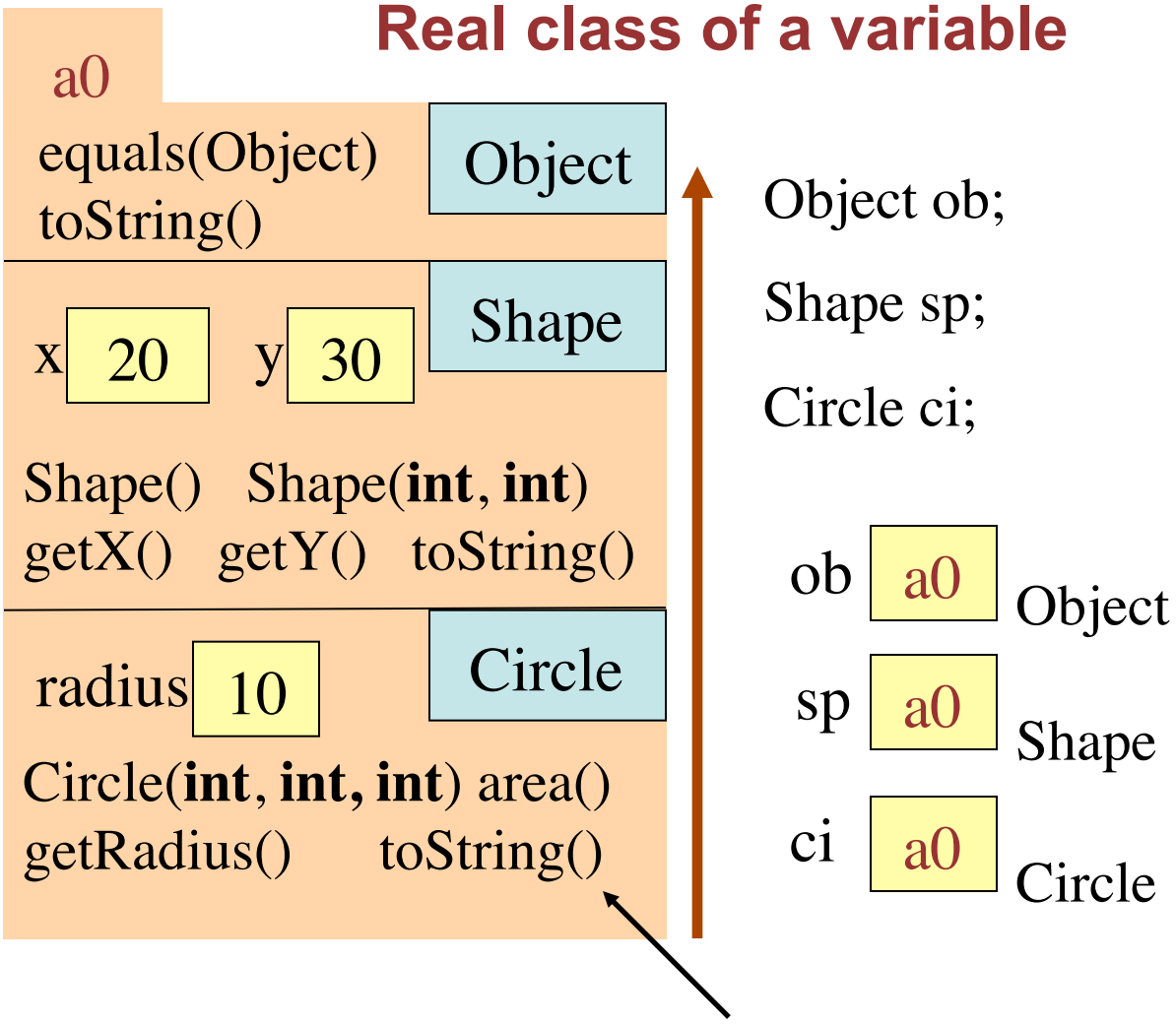


Apparent class: Class with which variable is defined.

Real class: What the object in the variable *really* is.

Real class: Has to do with execution. Can change during execution, when an assignment to the variable is executed.

Real class of a variable



Ob.toString() is legal. It calls this method.

Consequence of the bottom-up rule: the overriding method is called. This is an important aspect of OO!

We get the most information about the object by calling function toString of partition Circle.