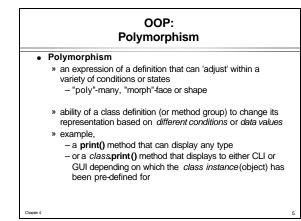


but <u>nothing</u> about "faculty wage info" (why should it?) » (most important aspect of OOP)



OOP: Inheritance

• Inheritance

• Inheritance

• the creation of a new definition based on the structure of a previous definition, but most importantly, without requiring complete knowledge of the previous definition

• technique of a class being created from another class, with

- the child automatically "inheriting" all attributes & methods of the parent class
- » example,
 - a new JApplet <u>extends JApplet</u> (the new applet class inherits all data and methods of the standard JApplet)
 creating a new **StringUpp** class based on the standard
- String class, except it maintains all data in uppercase » terms:
- child="sub-class", "parent"="super-class"

Formalised Abstraction: Abstract Data Types (ADTs) – OOP Encapsulation

- technique used in storing "real-world" descriptions
- an ADT implies a *class* implementation in Java
- » a container for data items, and methods to act on that data
- encourages information hiding and encapsulation
- the class's user interface allows programmers to use the ADT (class),
- » descriptions, parameters, and names of its methods
 Implementation:
 - » private instance variables
 - » method definitions (implementations) are hidden
 - » another programmer cannot see or change the *class*
 - » only sees the "interface" is seen or ever required

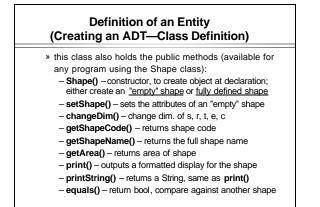
......

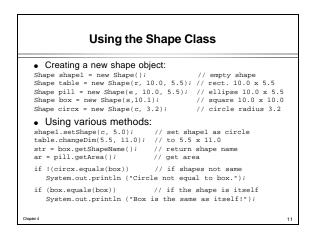
Variables Review: Class (Reference) Type vs. Primitive Type

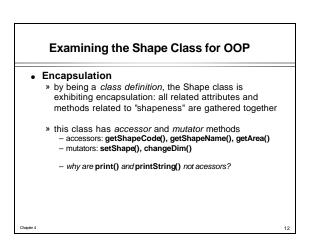
- a primitive type variable directly holds the value of the variable (identifier name is <u>directly-associated</u> with the memory address storing the data value)
- A reference variable type holds a *memory address*, that <u>indirectly points</u> to the an *object in memory*,
 - » objects in memory have multiple member variables, and member objects, of different types, along with "pointers" to the class's methods
 - note: all objects of the same class share the actual same methods, but use different data from the object

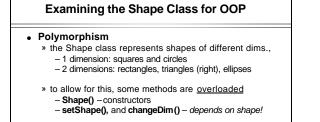
Definition of an Entity (Creating an ADT—Class Definition)

- Definition of a "Shape" class (see Shape.java)
 - » this class holds the statistics of a general shape, such as:
 - ${\color{black}{shapename}}$ ${\color{black}{square}}$, rectangle, triangleright , elllipse, and circle
 - shapecode s, r, t, e, c
 - -width, height dimensions for s, r, t, and e
 - radius dimension for c
 - area area of shape, based on code and dimensions
 - » the "shape" class encapsulates all the attributes that are needed to describe a simple geometric shape









Inheritance

Chapter 4

» the Shape class does not explicitly show inheritance

13

5/31/03