CHAPTER 2

Using Objects

Basic Programming Terminology

- Computer program process values.
  - Numbers (digits)
  - Words (Strings)
  - These values are different
  - Stored differently in the computer
- Every value has a type
  - Tells you kind of operations you can do
  - Example – you can't multiply strings.

Variables and Types

- Variable
  - Used to store values
  - Has a type, name, value
- Basic types:
  - Integer or int
  - Floating point numbers - float or double
  - Single character – char
  - Multiple character - String
- Number literal
  - Value such as 13 or 1.3

Number Literals in java

Number	Type	Comment
6	int	An integer has no fractional part.
-6	int	Integers can be negative.
0	int	Zero is an integer
0.5	double	A number with a fractional part - double
1.0	double	Still a fractional part
1E6	double	A number in exponential notation: 1 X 10^6 Always are double
2.9E-2	double	Same as above
100,000		Error: Can't have commas
3 ½		Error: Don't use fractions, use decimals.

You Try It

- What type of variable would you use.
  - Your name
  - The distance to the moon
  - Your address
  - Your social security number
  - Your total salary

Why Have Integers

- Take less storage space
- Process faster
- Don't cause rounding errors
- When to use?
  - Don't need fractional parts
- When to use double
  - When you need fractional part
- Will discuss other types in chapter 4
- Primitive types
- Expression

Variables

- Variable declaration examples:
  - String greeting = "Hello, World!";
  - printStream printer = System.out;
  - int luckyNumber = 13;

- Type must match your value
  - int myNumber = “007”; // error

Variable Declarations in Java

Variable Name	Comment
int width = 10;	Declares an integer variable and initializes it to 10.
int area = width*height;	The initial value can depend on other variables. Note: width and height have already been declared.
age= 5;	Error: the type is missing. This statement is not a declaration but an assignment.
int age = “5”;	Error: You cannot initialize a number with a string.
int width, height;	Declares two integer variables in a single statement.
double grade = 2.3;	Declares a double variable and initializes it to 2.3
String name ;	Declares a variable but does not assign a value to it.

Identifiers

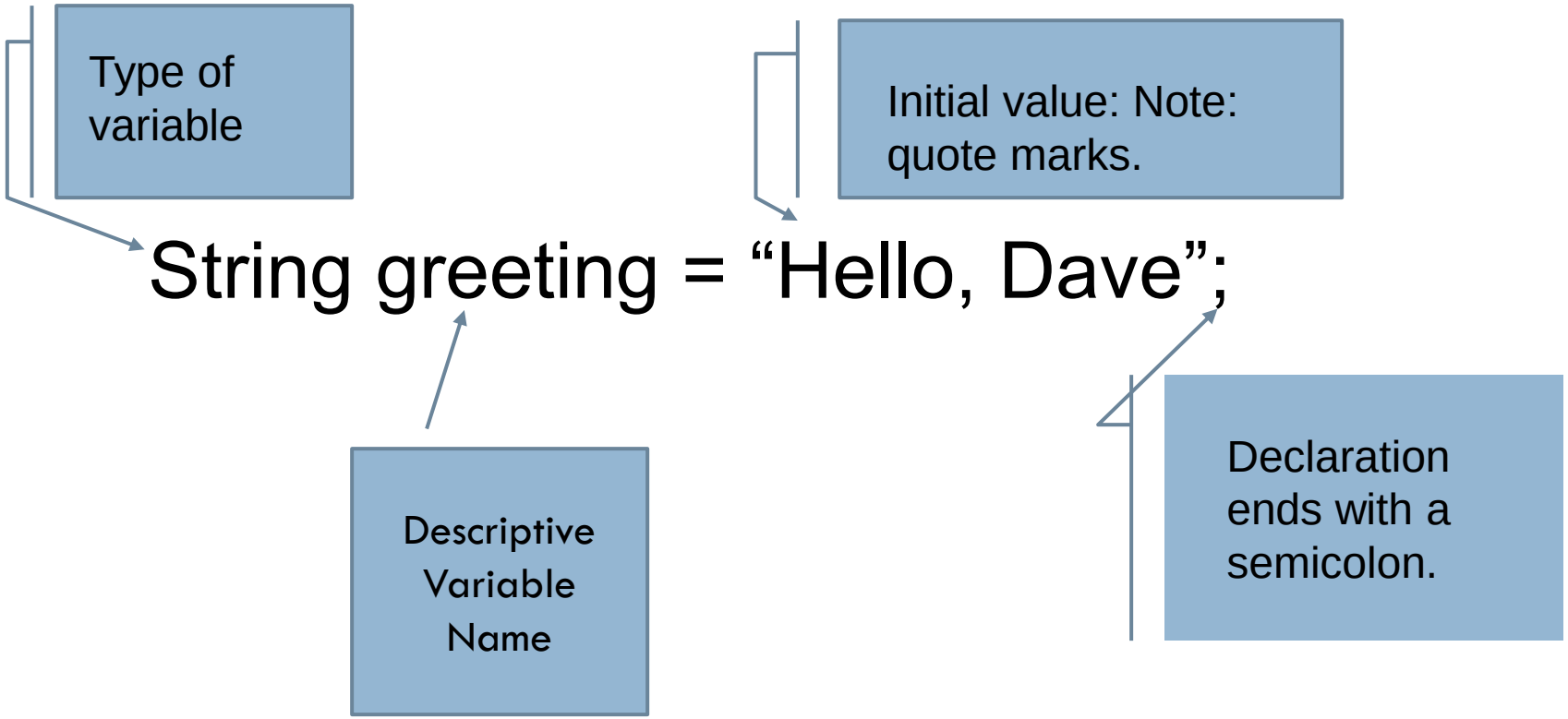
- The name of variable, method, class
- Rules
  - Can use letter, digits, underscore (_), $
  - Cannot start with digit
  - Cannot use spaces or symbols
  - Cannot use reserve words (Appendix C)
- Convention
  - Variable and method names
    - Start with lowercase
    - CamelCase ok and desirable
  - Classes start with uppercase
  - Don't use $ in name

Variable Name

Variable name	Comment
farewellMessage	Camel case – good
X	Legal but bad form
Greeting	Watch out for case sensitive
6pack	Error: Cannot start with a number
Farewell message	Error: cannot contain spaces
Public	Error: reserve word

Chose descriptive names for variables.

Variable Declaration



You Try It

- Create a variable for each of these.
  - Your name
  - The distance to the moon
  - Your address
  - Your social security number
  - Your total salary

Assignment

- Change value of variable
- Example
  - int width = 10;
  - width = 20;
- Must assign a value before using a variable.
- width = width + 10;

Objects, Classes, & Methods

- Object: A value you an manipulate by calling one or more of its methods
- Method: Sequence of instructions for the object
- Class: the type of the object
- Examples
  - System.out
    - Belongs to class PrintStream
    - Method println
  - “Hello World”
    - Belongs to String class

PrintStream Object

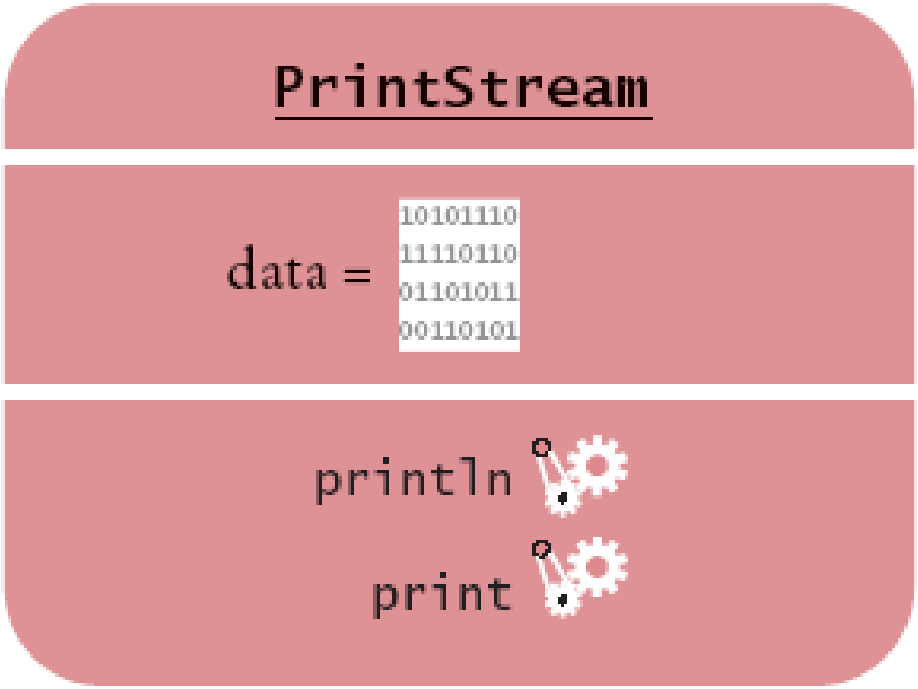
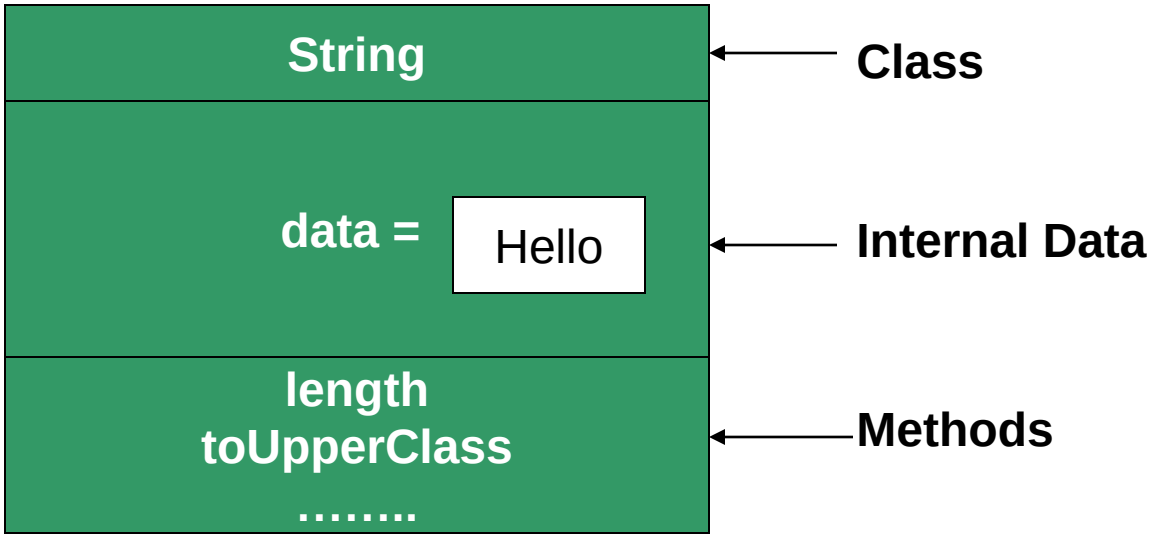


Figure 4 Representation of the `System.out` Object

String Object



Method Parameters

- Methods – fundamental build blocks of Java
- Perform work by calling the method
- Parameter
  - Input to a method
  - Provides info needed by method
- Two types of parameters
  - Explicit (goes in parentheses)
  - Implicit (goes in front of the method)
    - Defines the object being used

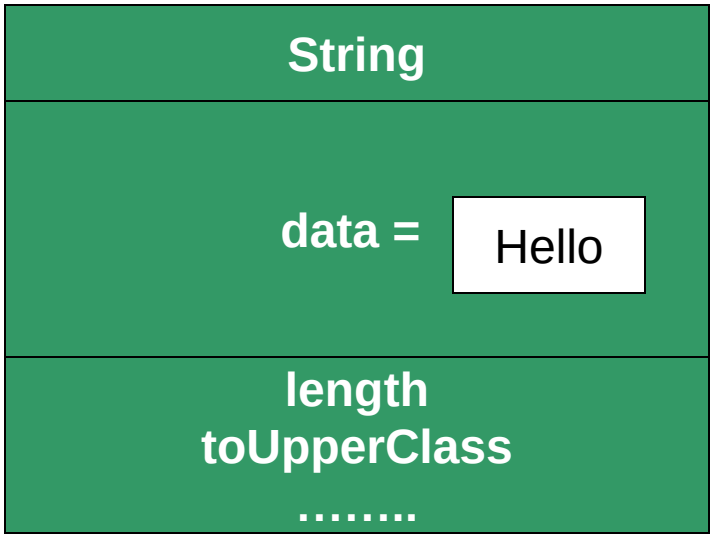
Method Parameters

- Example

```
System.out.println(greeting);
```

  - System.out
    - Explicit parameter
    - Object
  - println
    - Method
  - greeting
    - Implicit parameter
    - What to print

Example



- Look at method length
- Takes no explicit input to perform operations
- Provides some output information
  - In this case 5

Two More Methods

toUpperCase

- Converts all letter in string to upper case
- Example

```
String river = "Mississippi";
String creek = "Chickamauga";
String bigRiver = river.toUpperCase();
```

river holds Mississippi

bigRiver holds MISSISSIPPI

Return Values

- Used to provide information back to calling routine
- Provides the results
- Example
  - int n = greeting.length();
  - You will assign the value of the length of the value referred to by the variable greeting

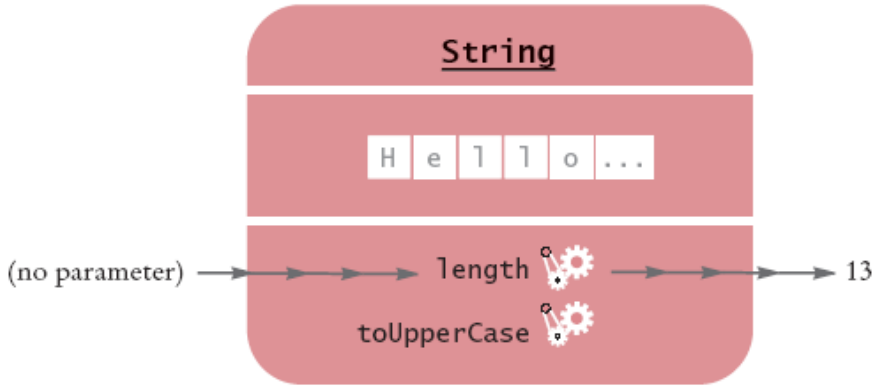


Figure 6 Invoking the `length` Method on a `String` Object

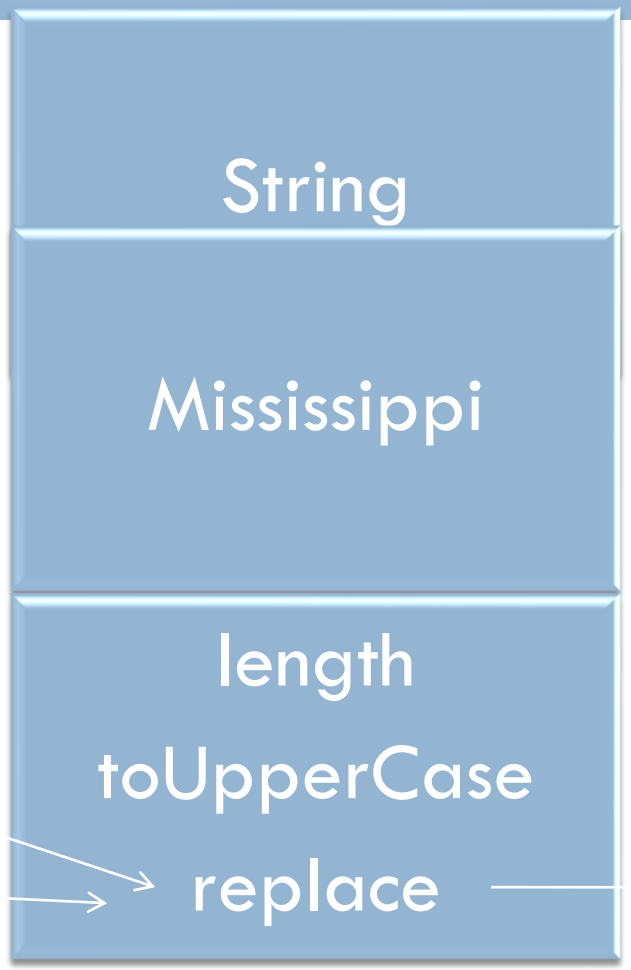
More Complex String Methods

String river = “Mississippi”;

River.replace(“ississi”, “our”);

- Replace is the method
- takes two values
  - What to replace
  - What to replace it with

Example



"issipp"

"our"

"Missouri"

Method Definitions

- Specifies the types of the explicit parameters
- Specifies the return type
- Example
  - public int length()
- Method not return anything – type void
  - public void addNumbers()

Method Definitions

- You can have multiple methods with the same name.
- Method name is overloaded if a class has more than one method with the same name (but different parameter types)
  - `public void println(String output)`
  - `public void println(int output)`

Constructing Object

- Will use Rectangle as an example
- Important note:
  - Not the drawing
  - Numbers to define the drawing
  - Need x and y coordinates of top left corner
  - Need width
  - Need height
  - Rectangle box = new Rectangle(10,10,20,30);

Constructing an Object

- Rectangle box = new Rectangle(10,10,20,30);
- What is happening
  - new – makes a new Rectangle object
  - Use the parameters 10,10,20,30 to initialize the data
  - Returns the object to the variable box
- This process is called constructing

Constructing a Rectangle

Type of Object

Name of Variable to hold object

The parameters need for the rectangle

```
Rectangle box = new Rectangle (5, 10, 20, 30);
```

Says it's a new one. Create object.

Will explain this one later.

Two Kinds of Method

- Accessor Methods
  - Allows us to access data from an object
  - Double width = box.getWidth();
- Mutator Methods
  - Change the method
  - Box.translate(15,20);
    - Moves the object 15 in the x-direction
    - Moves the object 20 in the y-direction

API Documentation

- API: Application Programming Interface
- Lists the classes and methods of the Java library
- http://java.sun.com/javase/6/docs/api/index.html
- Documents all classes in the library
- Thousands
- Appendix D = abbreviated version

Importing

- If you want to use a class from the API, you have to import it.
- Import java.awt.Rectangle;

Implementing a Test Program

- Steps
  - Provide a tester class
  - Supply a main
  - Inside the main method, construct 1+ objects
  - Apply methods to the objects
  - Display the results
  - Display the values expected

```
import java.awt.Rectangle;
public class MoveTester
{
	public static void main(String [] args)
	{
		Rectangle box = new Rectangle(5,10,20,30);
		box.translate(15,25);
		System.out.print("x ");
		System.out.println(box.getX());
		System.out.println("Expected: 20");
		System.out.print("y ");;
		System.out.println(box.getY());
		System.out.println("Expected: 35");
	}
}
```

Object References

- Object reference describes the location of an object
- The new operator returns a reference to a new object

  Rectangle box = new Rectangle();
- Multiple object variables can refer to the same object

```
Rectangle box = new Rectangle(5, 10, 20, 30);
Rectangle box2 = box;
box2.translate(15, 25);
```

- Primitive type variables ≠ object variables

Object References

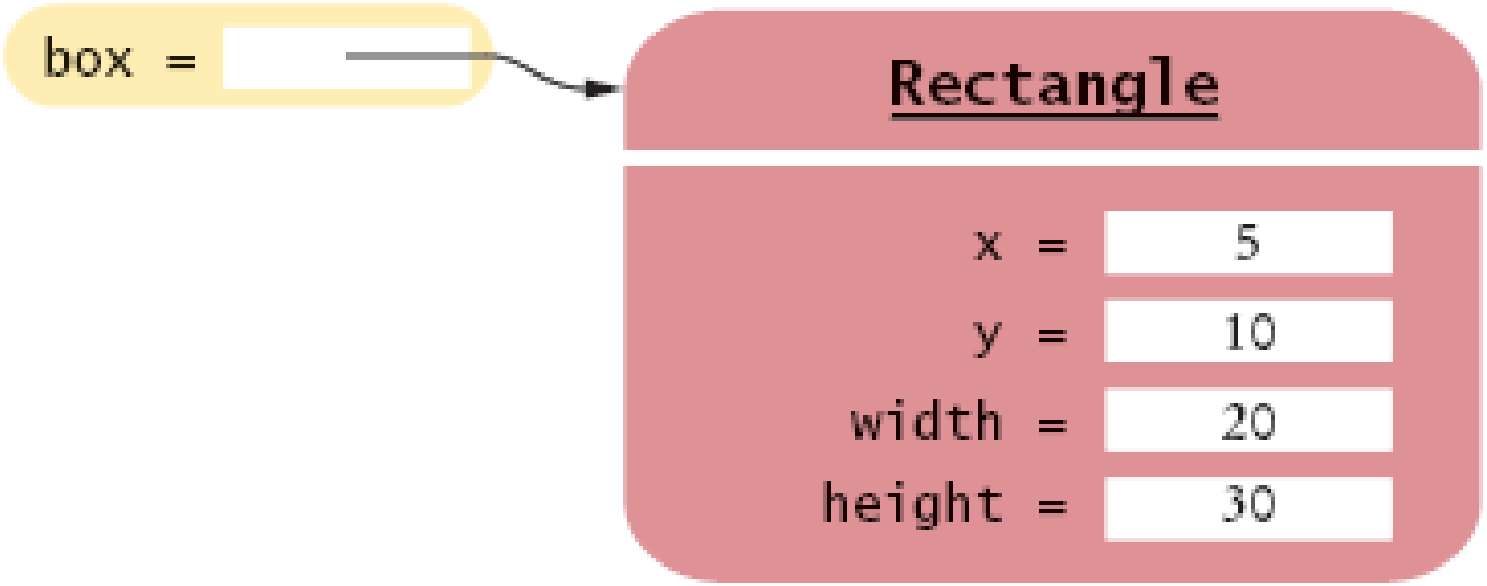


Figure 16 An Object Variable Containing an Object Reference

Object and Number References

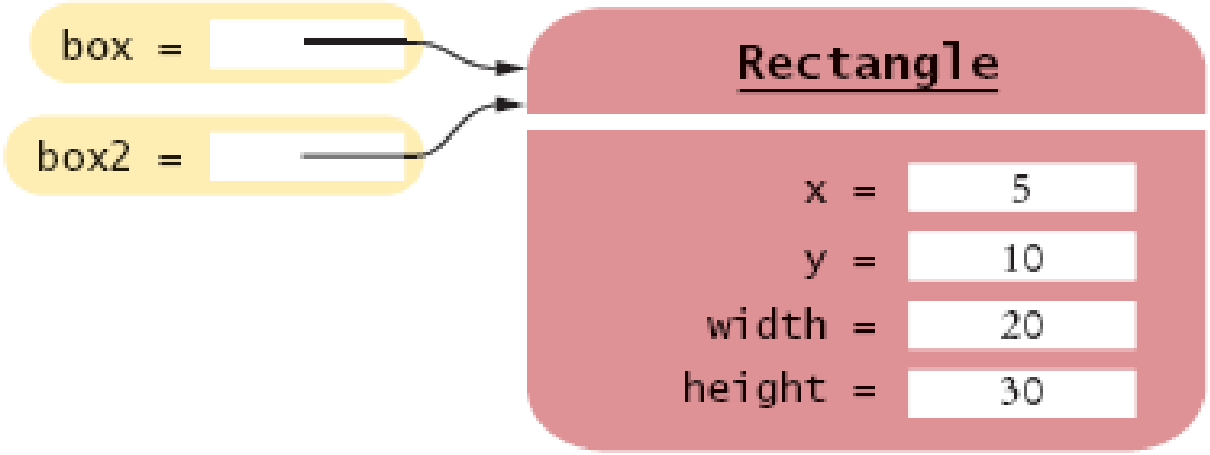


Figure 17 Two Object Variables Referring to the Same Object

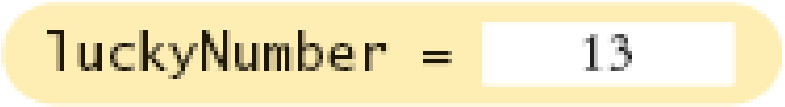


Figure 18 A Number Variable Stores a Number

Object and Number References

```
int luckyNumber = 13;

int luckyNumber2 =
luckyNumber;

luckyNumber2 = 12;
```

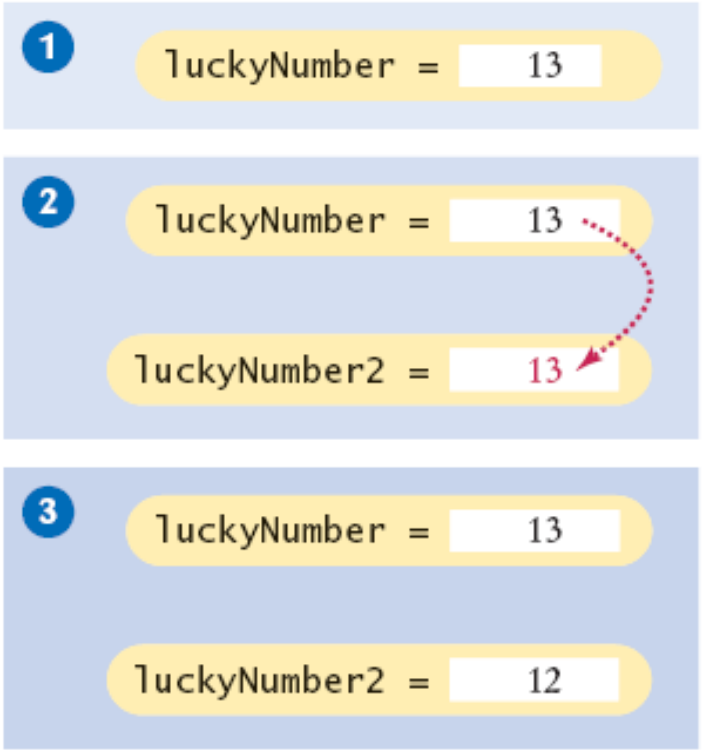
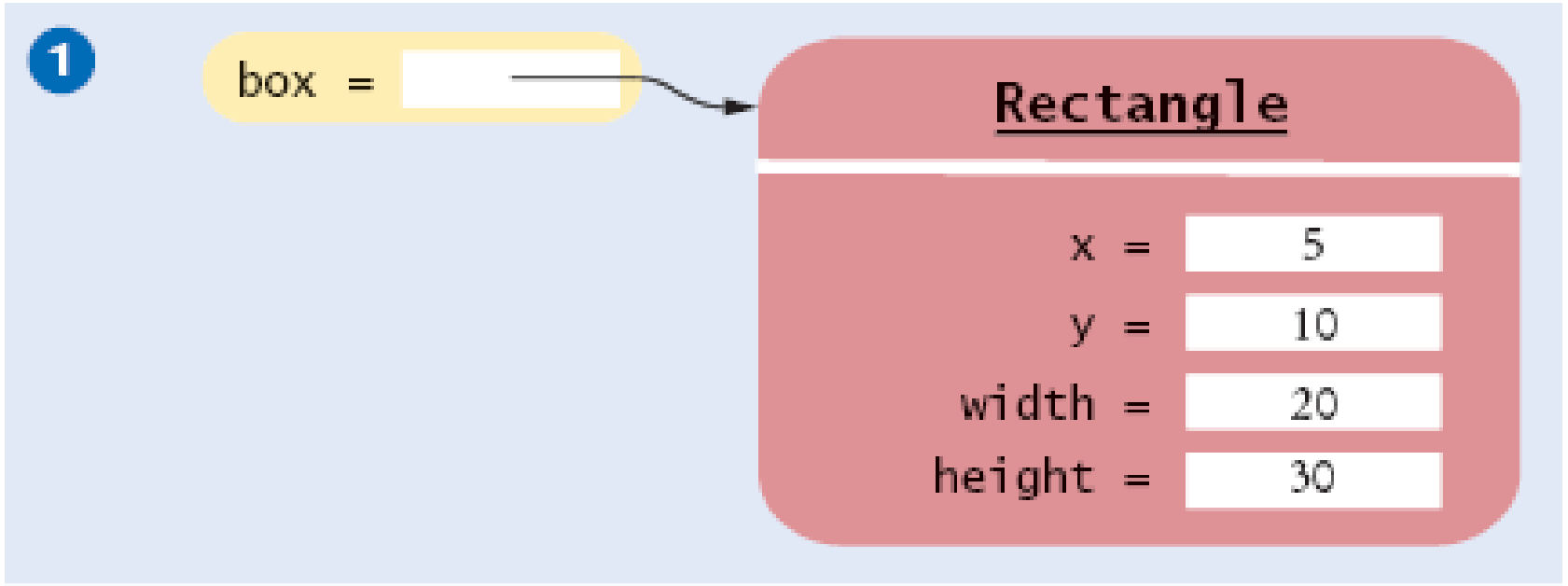


Figure 19
Copying Numbers

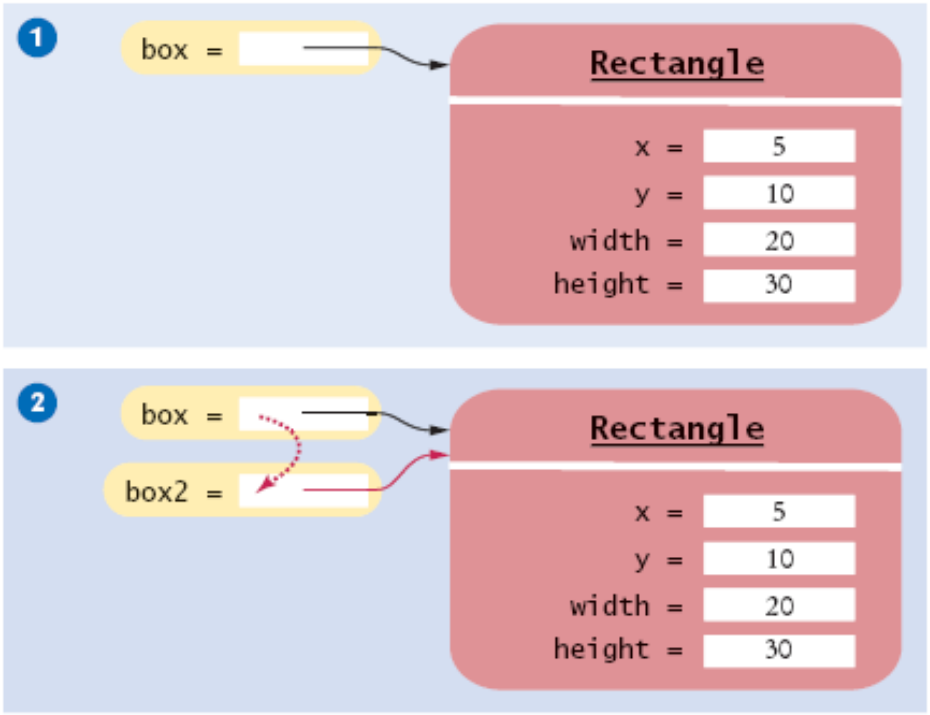
Copying Object References

```
Rectangle box = new Rectangle(5, 10, 20, 30);
```



Copying Object References

```
Rectangle box = new Rectangle(5, 10, 20, 30);
Rectangle box2 = box;
```



Copying Object References

```
Rectangle box = new Rectangle(5, 10, 20, 30);
Rectangle box2 = box;
Box2.translate(15, 25);
```

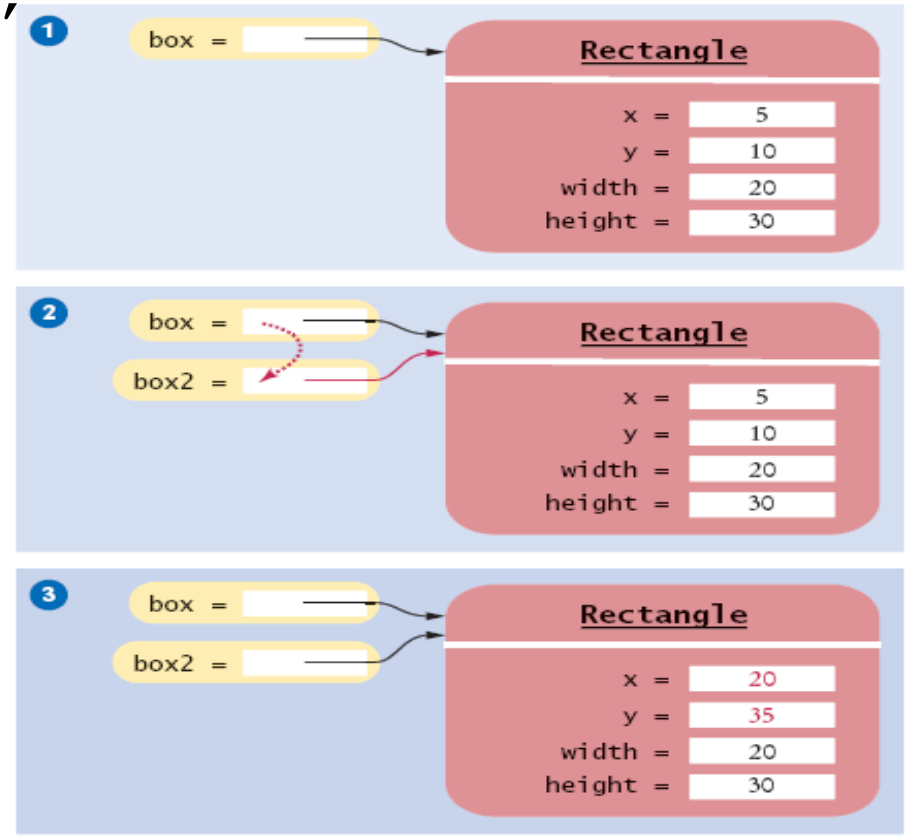


Figure 20 Copying Object References

Graphical Applications

- Display drawing inside a window
- Shows information inside a frame
- Frame – window with a text bar

To Show a Frame

- Construct an object of the JFrame Class;
  - JFrame frame = new Jframe();
- Set the size
  - frame.setSize(300,400);
- Set the title of the frame
  - frame.setTitle("First frame");
- Set the default close operation
  - frame.setDefaultCloseOperation(Jframe.EXIT_ON_CLOSE);
- Make the frame visible
  - frame.setVisible(true);

```
import javax.swing.JFrame;
public class EmptyFrameViewer2
{
	public static void main(String [] args)
	{
		JFrame frame = new JFrame();
		frame.setSize(300,400);
		frame.setTitle("My first frame");
		frame.setDefaultCloseOperation
				(JFrame.EXIT_ON_CLOSE);
		frame.setVisible(true);
	}
}
```

Drawing on a Component

- Cannot draw directly on a frame
- Must construct a component object
- Add the component object to the frame

Component

```
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import javax.swing.JComponent;
public class RectangleComponent extends JComponent
{
        public void paintComponent(Graphics g)
        {
                Graphics2D g2 = (Graphics2D) g;
                Rectangle box = new Rectangle (5,10,50,50);
                g2.draw(box);
                box.translate(15,25);
                g2.draw(box);
        }
}
```

Rectangle Viewer

```
import javax.swing.JFrame;
public class RectangleViewer
{
	public static void main(String [] args)
	{
		JFrame frame = new JFrame();
		frame.setSize(300,400);
		frame.setTitle("Two Rectangles");
		frame.setDefaultCloseOperation
			(JFrame.EXIT_ON_CLOSE);
		RectangleComponent component = new
					RectangleComponent();
		frame.add(component);
		frame.setVisible(true);
	}
}
```

Ellipses, Lines, Text, and Color

- To draw an ellipse, you specify its bounding box
- Like a rectangle
  - Specify x and y starting position
  - Specify the height
  - Specify the width
- Ellipse2D.Double ellipse = new Ellipse2D.Double(x,y,width,height);
- Use same draw method as rectangles

Draw Lines

- Specify the two end points
- Line2D.Double segment = new Line2D.Double(x1,y1,x2,y2);
- Can define from point
- Point2D.Double from = new Point2D.Double(x1,y1)
- Can define to point
- Point2D.Double to = new Point2D.Double(x1,y1)
- Line2D.Double segment = new Line2D.Double(from,to);

DrawText

- Used to put text inside a drawing
- Label some of the parts
- Use the drawString method
- Specify the string and the x and y positions of the base point of 1st character

- g2.drawString("Message", 50,100);

Colors

- Initially all drawing is in black
- Must change the color
- Predefined colors (page 67)
- Can define own colors.

- g2.setColor(Color:RED);
- g2.draw(circle);
- g2.fill(circle);