1. 
   public class Camera {
      ...
      private Rectangle _shutter;
      private Flash _flash;
      private Battery _battery;
      ...

2. 
   public class Camera {
      ...
      public Camera() {
         ...
         _shutter = new Rectangle();
         _flash = new Flash();
         _battery = new Battery();
         ...
   
3. File “Flash.java”:

   public class Flash {
   }

   File “Battery.java”:

   public class Battery {
   }

4. import wheels.users.*;

   public class CameraApp extends Frame {
      private Camera _camera;

      public CameraApp() {
         _camera = new Camera();
      }

      public static void main(String[] args) {
         CameraApp app = new CameraApp();
      }
   }
5. 

```java
public class TakePicture extends wheels.users.Frame {
    
    private wheels.users.ConversationBubble _message;
    
    _message = new wheels.users.ConversationBubble("Smile for the paparazzi!");
    
    ...
```

6. The Camera class is not a complete application with frame, “Quit” button, “main” method, etc., but just a description of one kind of object, a camera. The TakePicture class is the application, and wheels applications extend the Frame class to provide the frame, or drawing canvas, in which all the other wheels objects are displayed.

7. This is simply the notation chosen by the authors of the textbook to indicate which variables are instance variables (i.e., properties of the object). There is no other significance. In particular, the Java language does not require this underscore character in the names of instance variables.

8. The program will compile and execute with no change. In Java programs, indentation, line breaks, and extra internal spacing are not important for the proper behavior of the program, with a few exceptions:
   - Spaces may not be introduced inside a name. For instance, “_cameraBody” is legal, but “_camera body” is not
   - A one-line comment (that is, a comment that begins with two slashes: //) is terminated by a line break
   - Inside string constants, spaces are significant (they become part of the string). Thus, "hi there" and "hi there" are different strings. Also, you cannot break a string in the middle with a line break — your program will not compile if you do this.

   However, spacing is very important for the readability of your program by yourself and others!

9. 

```
+------------------------+ +----------------------------------+
| Snowman               | | Hat                            |
| +------------------------+ +----------------------------------+
| _head: Ellipse        | | _hatBrim: Rectangle            |
| _body: Ellipse        | | _hatUpper: Rectangle           |
| _leftEye: Ellipse     | +----------------------------------+
| _rightEye: Ellipse    | | Hat()                          |
| +------------------------+ | setLocation(int x, int y): void |
| Snowman()             | +----------------------------------+
| | setOutline(): void      |
+------------------------+
```

10. NOTE: The following solution is correct, but there is a much easier way to answer the question without duplicating all of the code in the original constructor! (We’ll discuss this after the exam.)
public class Snowman {
    ...
    public Snowman() {
        // old constructor remains
    }

    public Snowman(Color c) { // new constructor
        _body = new Ellipse(c);
        _body.setSize(100, 100);
        _head = new Ellipse(c);
        _head.setSize(80, 80);
        _leftEye = new Ellipse(Color.BLACK);
        _leftEye.setSize(15, 15);
        _rightEye = new Ellipse(Color.BLACK);
        _rightEye.setSize(15, 15);

        _body.setLocation(10, 300);
        _head.setLocation(20, 240);
        _leftEye.setLocation(35, 265);
        _rightEye.setLocation(75, 265);
        setOutline(Color.BLACK, 2);
    }
    ...
}

11. No. The two rectangles that make up the hat are both placed in the default location at the center of the frame, so they don’t look like a hat.

12. public class Hat {
    ...
    public Hat() {
        // old constructor
    }

    public Hat(int x, int y) {
        _hatBrim = new Rectangle(java.awt.Color.black);
        _hatBrim.setSize(80, 20);

        _hatUpper = new Rectangle(java.awt.Color.black);
        _hatUpper.setSize(60, 60);

        setLocation(x, y);
    }

    13. public SnowCartoon() {
        _snowman = new Snowman();
        _hat = new Hat(20, 180); // USE THE NEW CONSTRUCTOR!
        ...
    }