Question 1.1

```
/* Saket Bakshi, Period 6
Question 1.1 of Ch 3 project. This program creates an empty class for a door class.
*/
public class Door
{

}
```

Question 1.2

```
/* Saket Bakshi, Period 6
Question 1.2 of Ch 3 project. This program creates instance variables for a door class.
*/
public class DoorV2
{
	private String name;
	private String state;
}
```

Question 1.3

```
/* Saket Bakshi, Period 6
Question 1.3 of Ch 3 project. This program creates methods for a door class.
*/
public class DoorV3
{
	private String name;
	private String state;

	public void close()
	{
		state = "close";
	}

	public void open()
	{
		state = "open";
	}
}
```

Question 1.4

```
/* Saket Bakshi, Period 6
Question 1.4 of Ch 3 project. This program creates constructors for a door class.
*/
public class DoorV4
{
	private String name;
	private String state;

	public DoorV3(String doorName, String doorState)
	{
		this.name = doorName;
		this.state = doorState;
	}

	public void close()
	{
		state = "close";
	}

	public void open()
	{
		state = "open";
	}
}
```

Question 1.5

```
/* Saket Bakshi, Period 6
Question 1.5 of Ch 3 project. This program creates an accessor method for a door class.
*/
public class DoorV5
{
	private String name;
	private String state;

	public DoorV3(String doorName, String doorState)
	{
		this.name = doorName;
		this.state = doorState;
	}
```

```
	public void close()
	{
		state = "close";
	}

	public void open()
	{
		state = "open";
	}

	public String getName()
	{
		return name;
	}

	public String getState()
	{
		return state;
	}
}
```

Question 1.6

```
/* Saket Bakshi, Period 6
Question 1.6 of Ch 3 project. This program creates mutator methods for a door class.
*/
public class DoorV6
{
	private String name;
	private String state;

	public DoorV3(String doorName, String doorState)
	{
		this.name = doorName;
		this.state = doorState;
	}

	public void close()
	{
		state = "close";
	}

	public void open()
```

```
	{
		state = "open";
	}

	public String getName()
	{
		return name;
	}

	public String getState()
	{
		return state;
	}

	public void setName(String newName)
	{
		this.name = newName;
	}

	public void setState(String newState)
	{
		this.state = newState;
	}
}
```

Question 1.7

```
/**
	A class to test the Door class.
*/
public class DoorTester
{
	public static void main(String[] args)
	{
		DoorV6 frontDoor = new DoorV6("Front", "open");
		System.out.println("The front door is " + frontDoor.getState());
		System.out.println("Expected:  open");
		DoorV6 backDoor = new DoorV6("Back", "closed");
		System.out.println("The back door is " + backDoor.getState());
		System.out.println("Expected:  closed");
		backDoor.setState("open");
		System.out.println("The back door is " + backDoor.getState());
		System.out.println("Expected:  open");
```

```
                backDoor.setName("myBackDoor");
                System.out.println("The back door is called " + backDoor.getName());
                System.out.println("Expected: myBackDoor");
                DoorV6 sideDoor = new DoorV6("side", "closed");
                System.out.println("The side door is " + sideDoor.getState());
                System.out.println("Expected:  closed");
                System.out.println("The side door is called " + sideDoor.getName());
                System.out.println("Expected: side");
                sideDoor.setState("open");
                System.out.println("The side door is " + sideDoor.getState());
                System.out.println("Expected:  open");
        }
}
```

```
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ1SBakshi> java DoorTester
The front door is open
Expected:  open
The back door is closed
Expected:  closed
The back door is open
Expected:  open
The back door is called myBackDoor
Expected: myBackDoor
The side door is closed
Expected:  closed
The side door is called side
Expected: side
The side door is open
Expected:  open
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ1SBakshi>
```

Exercise 1.8

```
/* Saket Bakshi, Period 6
Question 1.8 of Ch 3 project. This program explains what kind of variables state and newState
are
*/
/*
state is an instance variable of the DoorV6 class. Each DoorV6 object has a state
variable.
newState is a parameter variable of the mutator setState() that is created when a user
uses setState()
and deleted when the method is finished.

*/
public class DoorV7
{
        private String name;
        private String state;

        public DoorV7(String doorName, String doorState)
```

```
	{
		this.name = doorName;
		this.state = doorState;
	}

	public void close()
	{
		state = "close";
	}

	public void open()
	{
		state = "open";
	}

	public String getName()
	{
		return name;
	}

	public String getState()
	{
		return state;
	}

	public void setName(String newName)
	{
		this.name = newName;
	}

	public void setState(String newState)
	{
		this.state = newState;
	}
}
```

Exercise 1.9

```
/*
The explicit parameter is the String “open” and the implicit parameter is the object, backDoor.
*/
```

Exercise 2.1

```
/* Saket Bakshi, Period 6
Question 2.1 of Ch 3 project. This program prints ideas of methods for a vendingmachine class.
```

```
*/

public class Ch3ProjectQ2_1SBakshi
{
	public static void main(String[] args)
	{
		System.out.println("Methods for cans could be:\ngetCans(), which returns the number of cans in the machine\naddCans(), which adds a number of cans to a machine\nbuySoda(), which removes a can from the machine");
		System.out.println("Methods for tokens could be:\ngetTokens(), which returns how many tokens the machine has\nremoveTokens(), to empty the machine\nbuySoda(), with an explicit parameter for number of tokens added");
	}
}
```

```
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ2SBakshi> java Ch3ProjectQ2_1SBakshi
Methods for cans could be:
getCans(), which returns the number of cans in the machine
addCans(), which adds a number of cans to a machine
buySoda(), which removes a can from the machine
Methods for tokens could be:
getTokens(), which returns how many tokens the machine has
removeTokens(), to empty the machine
buySoda(), with an explicit parameter for number of tokens added
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ2SBakshi>
```

Exercise 2.2

```
/* Saket Bakshi, Period 6
Question 2.2 of Ch 3 project. This program declares a class with the methods stated in exercise 2.1.
*/
public class VendingMachine
{
	/** This returns the amount of cans in the machine
	@return the number of cans in the machine
	*/
	public int getCans()
	{
		return this.cans;
	}

	/** this adds cans to the machine
	@param cansAdded the number of cans added
	*/
	public void addCans(int cansAdded)
	{
```

```
		this.cans = this.cans + cansAdded;
	}

	/**this removes soda for the price of one token added to the machine
	@param cansBought the number of cans bought
	*/
	public void buySoda(int cansBought)
	{
		this.cans = this.cans - cansBought;
		this.tokens = this.tokens + cansBought;
	}

	/**this returns the number of tokens in the machine
	@return the number of tokens in the machine
	*/
	public int getTokens()
	{
		return this.tokens;
	}

	/**this removes all tokens from the machine
	*/
	public void removeTokens()
	{
		this.tokens = 0;
	}
}
```

Exercise 2.3

```
/* Saket Bakshi, Period 6
Question 2.3 of Ch 3 project. This program declares a class with the methods
and instance variables stated in exercise 2.1. Has all comments necessary for javadoc
documentation.
*/

public class VendingMachine
{

	//instance variables
	private int tokens;
	private int cans;

	/** This creates an object of the vending machine class that
```

```
        has a certain amount of cans that cost 1 token and the number of tokens. It can be used
by methods that
        find the number of cans or tokens in the machine. It also has methods to
        add more cans, buy cans, and remove the coins.
        @param cans the number of initial cans
        @param tokens the number of initial tokens
        */
        public VendingMachine(int cans, int tokens)
        {
                this.cans = cans;
                this.tokens = tokens;
        }

        /** This returns the amount of cans in the machine
        @return the number of cans in the machine
        */
        public int getCans()
        {
                return this.cans;
        }

        /** this adds cans to the machine
        @param cansAdded the number of cans added
        */
        public void addCans(int cansAdded)
        {
                this.cans = this.cans + cansAdded;
        }

        /**this removes soda for the price of one token added to the machine
        @param cansBought the number of cans bought
        */
        public void buySoda(int cansBought)
        {
                this.cans = this.cans - cansBought;
                this.tokens = this.tokens + cansBought;
        }

        /**this returns the number of tokens in the machine
        @return the number of tokens in the machine
        */
        public int getTokens()
        {
```

```
		return this.tokens;
	}

	/**this removes all tokens from the machine
	*/
	public void removeTokens()
	{
		this.tokens = 0;
	}
}
```

Exercise 2.4

```
/* Saket Bakshi, Period 6
Question 2.4 of Ch 3 project. This program declares a class with the methods
and instance variables stated in exercise 2.1. Has all comments necessary for javadoc
documentation.
*/

public class VendingMachine
{

	//instance variables
	private int tokens;
	private int cans;

	/** This creates an object of the vending machine class that
	has a certain amount of cans that cost 1 token and the number of tokens. It can be used
by methods that
	find the number of cans or tokens in the machine. It also has methods to
	add more cans, buy cans, and remove the coins.
	@param cans the number of initial cans
	@param tokens the number of initial tokens
	*/
	public VendingMachine(int cans, int tokens)
	{
		this.cans = cans;
		this.tokens = tokens;
	}

	/** This returns the amount of cans in the machine
	@return the number of cans in the machine
	*/
	public int getCans()
```

```
	{
		return this.cans;
	}

	/** this adds cans to the machine
	@param cansAdded the number of cans added
	*/
	public void addCans(int cansAdded)
	{
		this.cans = this.cans + cansAdded;
	}

	/**this removes soda for the price of one token added to the machine
	@param cansBought the number of cans bought
	*/
	public void inserToken(int cansBought)
	{
		this.cans = this.cans - cansBought;
		this.tokens = this.tokens + cansBought;
	}

	/**this returns the number of tokens in the machine
	@return the number of tokens in the machine
	*/
	public int getTokens()
	{
		return this.tokens;
	}

	/**this removes all tokens from the machine
	*/
	public void removeTokens()
	{
		this.tokens = 0;
	}
}
```

Exercise 2.5

```
/* Saket Bakshi, Period 6
Question 2.5 of Ch 3 project. This program declares a class with the methods
and instance variables stated in exercise 2.1. Has all comments necessary for javadoc
documentation.
*/

public class VendingMachine
{

	//instance variables
	private int tokens;
	private int cans;

	/** This creates an object of the vending machine class that
	has a certain amount of cans that cost 1 token and the number of tokens. It can be used
by methods that
	find the number of cans or tokens in the machine. It also has methods to
	add more cans, buy cans, and remove the coins.
	@param cans the number of initial cans
	@param tokens the number of initial tokens
	*/
	public VendingMachine(int cans, int tokens)
	{
		this.cans = cans;
		this.tokens = tokens;
	}

	/** This returns the amount of cans in the machine
	@return the number of cans in the machine
	*/
	public int getCans()
	{
		return this.cans;
	}

	/** this adds cans to the machine
	@param cansAdded the number of cans added
	*/
	public void fillUp(int cansAdded)
	{
		this.cans = this.cans + cansAdded;
	}
```

```
	/**this removes soda for the price of one token added to the machine
	@param cansBought the number of cans bought
	*/
	public void buySoda(int cansBought)
	{
		this.cans = this.cans - cansBought;
		this.tokens = this.tokens + cansBought;
	}

	/**this returns the number of tokens in the machine
	@return the number of tokens in the machine
	*/
	public int getTokens()
	{
		return this.tokens;
	}

	/**this removes all tokens from the machine
	*/
	public void removeTokens()
	{
		this.tokens = 0;
	}
}
```

Exercise 2.6

```
/* Saket Bakshi, Period 6
Question 2.6 of Ch 3 project. This program declares a class with the methods
and instance variables stated in exercise 2.1. Has all comments necessary for javadoc
documentation.
*/

public class VendingMachine
{

	//instance variables
	private int tokens;
	private int cans;

	/** This creates an object of the vending machine class that
	has a certain amount of cans that cost 1 token and the number of tokens. It can be used
by methods that
```

```
find the number of cans or tokens in the machine. It also has methods to
add more cans, buy cans, and remove the coins.
@param cans the number of initial cans
@param tokens the number of initial tokens
*/
public VendingMachine(int cans, int tokens)
{
	this.cans = cans;
	this.tokens = tokens;
}

/** This returns the amount of cans in the machine
@return the number of cans in the machine
*/
public int getCanCount()
{
	return this.cans;
}

/** this adds cans to the machine
@param cansAdded the number of cans added
*/
public void addCans(int cansAdded)
{
	this.cans = this.cans + cansAdded;
}

/**this removes soda for the price of one token added to the machine
@param cansBought the number of cans bought
*/
public void buySoda(int cansBought)
{
	this.cans = this.cans - cansBought;
	this.tokens = this.tokens + cansBought;
}

/**this returns the number of tokens in the machine
@return the number of tokens in the machine
*/
public int getTokenCount()
{
	return this.tokens;
}
```

```
	/**this removes all tokens from the machine
	*/
	public void removeTokens()
	{
		this.tokens = 0;
	}
}
```

Exercise 2.7

```
/* Saket Bakshi, Period 6
Question 2.7 of Ch 3 project. This program declares a class with the methods
and instance variables stated in exercise 2.1. Has all comments necessary for javadoc
documentation.
*/

public class VendingMachine
{

	//instance variables
	private int tokens;
	private int cans;

	/** This creates an object of the vending machine class that
	has a certain amount of cans that cost 1 token and the number of tokens. It can be used
by methods that
	find the number of cans or tokens in the machine. It also has methods to
	add more cans, buy cans, and remove the coins.
	@param cans the number of initial cans
	@param tokens the number of initial tokens
	*/
	public VendingMachine(int cans, int tokens)
	{
		this.cans = cans;
		this.tokens = tokens;
	}

	public VendingMachine()
	{
		this.cans = 0;
		this.tokens = 0;
	}
```

```
	/** This returns the amount of cans in the machine
	@return the number of cans in the machine
	*/
	public int getCans()
	{
		return this.cans;
	}

	/** this adds cans to the machine
	@param cansAdded the number of cans added
	*/
	public void addCans(int cansAdded)
	{
		this.cans = this.cans + cansAdded;
	}

	/**this removes soda for the price of one token added to the machine
	*/
	public void buySoda()
	{
		this.cans = this.cans - 1;
		this.tokens = this.tokens + 1;
	}

	/**this returns the number of tokens in the machine
	@return the number of tokens in the machine
	*/
	public int getTokens()
	{
		return this.tokens;
	}

	/**this removes all tokens from the machine
	*/
	public void removeTokens()
	{
		this.tokens = 0;
	}
}
```

Exercise 2.8

public class VendingMachineTester

```
{
	public static void main(String[] args)
	{
		VendingMachine machine = new VendingMachine();
		machine.addCans(10); // Fill up with ten cans
		machine.buySoda();
		machine.buySoda();
		System.out.print("Token count: ");
		System.out.println(machine.getTokens());
		System.out.println("Expected: 2");
		System.out.print("Can count: ");
		System.out.println(machine.getCans());
		System.out.println("Expected: 8");
	}
}
```

```
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ2SBakshi> java VendingMachineTester
Token count: 2
Expected: 2
Can count: 8
Expected: 8
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ2SBakshi>
```

Exercise 2.9

```
/* Saket Bakshi, Period 6
Question 2.9 of Ch 3 project. This program declares a class with the methods
and instance variables stated in exercise 2.1. Has all comments necessary for javadoc
documentation.
*/

public class VendingMachine
{

	//instance variables
	private int tokens;
	private int cans;

	/** This creates an object of the vending machine class that
	has a certain amount of cans that cost 1 token and the number of tokens. It can be used by methods that
	find the number of cans or tokens in the machine. It also has methods to
	add more cans, buy cans, and remove the coins.
	@param cans the number of initial cans
	@param tokens the number of initial tokens
	*/
```

```
public VendingMachine(int cans)
{
	this.cans = cans;
	this.tokens = 0;
}

public VendingMachine()
{
	this.cans = 0;
	this.tokens = 0;
}

/** This returns the amount of cans in the machine
@return the number of cans in the machine
*/
public int getCans()
{
	return this.cans;
}

/** this adds cans to the machine
@param cansAdded the number of cans added
*/
public void addCans(int cansAdded)
{
	this.cans = this.cans + cansAdded;
}

/**this removes soda for the price of one token added to the machine
*/
public void buySoda()
{
	this.cans = this.cans - 1;
	this.tokens = this.tokens + 1;
}

/**this returns the number of tokens in the machine
@return the number of tokens in the machine
*/
public int getTokens()
{
	return this.tokens;
}
```

```
	/**this removes all tokens from the machine
	*/
	public void removeTokens()
	{
		this.tokens = 0;
	}
}
```

Exercise 3.1

```
/* Saket Bakshi, Period 6
Question 3.1 of Ch 3 project. This program prints a class name for a postcard sender.
*/

public class Ch3ProjectQ3_1SBakshi
{
	public static void main(String[] args)
	{
		System.out.println("A black box name could be \"Postcard\"");
	}
}
```

```
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ3SBakshi> javac *.java
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ3SBakshi> java Ch3ProjectQ3_1SBakshi
A black box name could be "Postcard"
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ3SBakshi>
```

Exercise 3.2

```
/* Saket Bakshi, Period 6
Question 3.2 of Ch 3 project. This program declares a Postcard class for sending
postcards with the same message to different recipients.
*/

public class Postcard
{
	//instance variables
	private String message;
	private String sender;
	private String recipient;

	/**Makes an object for a postcard class with a sender and a message.
	Has methods to change recipients.
	@param aSender the sender
	@param aMessage the message
```

```
	*/
	public Postcard(String aSender, String aMessage)
	{

		this.message = aMessage;
		this.sender = aSender;
		this.recipient = "";
	}

	public void changeRecipient(String newRecipient)
	{
		this.recipient = newRecipient;
	}
}
```

Exercise 3.3

```
/* Saket Bakshi, Period 6
Question 3.3 of Ch 3 project. This program declares a Postcard class for sending
postcards with the same message to different recipients.
*/

public class Postcard
{
	//instance variables
	private String message;
	private String sender;
	private String recipient;

	/**Makes an object for a postcard class with a sender and a message.
	Has methods to change recipients.
	@param aSender the sender
	@param aMessage the message
	*/
	public Postcard(String aSender, String aMessage)
	{

		this.message = aMessage;
		this.sender = aSender;
		this.recipient = "";
	}

	public void changeRecipient(String newRecipient)
	{
```

```
		this.recipient = newRecipient;
	}

	public void print()
	{
		System.out.println("Hello, " + this.recipient);
		System.out.println(this.message);
		System.out.println("From,"\n+this.sender);
	}
}
```

Exercise 3.4

```
/* Saket Bakshi, Period 6
Question 3.4 of Ch 3 project. This program declares a Postcard class for sending
postcards with the same message to different recipients.
*/

public class Postcard
{
	//instance variables
	private String message;
	private String sender;
	private String recipient;

	/**Makes an object for a postcard class with a sender and a message.
	Has methods to change recipients.
	@param aSender the sender
	@param aMessage the message
	*/
	public Postcard(String aSender, String aMessage)
	{

		this.message = aMessage;
		this.sender = aSender;
		this.recipient = "";
	}

	public void setRecipient(String newRecipient)
	{
		this.recipient = newRecipient;
	}

	public void print()
```

```
	{
		System.out.println("Hello, " + this.recipient);
		System.out.println(this.message);
		System.out.println("From,\n" +this.sender);
	}
}
/* Saket Bakshi, Period 6
Question 3.4 of Ch 3 project. This program tests the Postcard class.
*/
public class PostcardPrinter
{
	public static void main(String[] args)
	{
		String text = "I am having a great time on\nthe island of Java. Weather\nis great.
Wish you were here!";
		Postcard postcard = new Postcard("Janice", text);
		postcard.setRecipient("Sue");
		postcard.print();
		postcard.setRecipient("Tim");
		postcard.print();
	}
}
```

```
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ3SBakshi> javac *.java
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ3SBakshi> java PostcardPrinter
Hello, Sue
I am having a great time on
the island of Java. Weather
is great. Wish you were here!
From,
Janice
Hello, Tim
I am having a great time on
the island of Java. Weather
is great. Wish you were here!
From,
Janice
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ3SBakshi>
```

Exercise 4.1

```
/* Saket Bakshi, Period 6
Question 4.1 of Ch 3 project. This program prints the number
of classes I would use to print several cars and houses.
*/

public class Ch3ProjectQ4_1SBakshi
{
	public static void main(String[] args)
```

```
	{
		System.out.println("I would have a class for the frame, a class for making the components, a class for making the cars, and a class for making the houses.");
	}
}
```

```
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ4SBakshi> javac *.java
PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ4SBakshi> java Ch3ProjectQ4_1SBakshi
I would have a class for the frame, a class for making the components, a class for making the cars, and a class for making the houses.

PS C:\Users\saket\JAVA\Projects\project3\Ch3ProjectQ4SBakshi>
```

Exercise 4.2

```
/* Saket Bakshi, Period 6
Question 4.2 of Ch 3 project. This program creates a house class for a position, with the
top left corner of the rectangle of the house being the constuctor explcit argument.
*/

import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Line2D;

public class House
{
	private int xLeft;
	private int yTop;

	/**Constructs a house with a given top left corner for the house rectangle.
	@param x the x-coordinate for the corner
	@param y the y-coordinate for the corner
	*/
	public House(int x, int y)
	{
		this.xLeft = x;
		this.yTop = y;
	}

	/**Draws the house
	@param g2 the graphics context
	*/
	public void draw(Graphics2D g2)
	{
		Line2D.Double ceiling = new Line2D.Double(xLeft, yTop + 40, xLeft + 110, yTop + 40);
```

```
                Line2D.Double roof1 = new Line2D.Double(xLeft, yTop + 40, xLeft + 55, yTop);
                Line2D.Double roof2 = new Line2D.Double(xLeft + 110, yTop + 40, xLeft + 55,
yTop);
                Rectangle houseBody = new Rectangle(xLeft + 10, yTop + 40, 90, 70);
                Rectangle door = new Rectangle(xLeft + 40, yTop + 60, 30, 50);
                Ellipse2D.Double knob = new Ellipse2D.Double(xLeft + 45, yTop + 85, 5, 5);

                g2.draw(ceiling);
                g2.draw(roof2);
                g2.draw(roof1);
                g2.draw(houseBody);
                g2.draw(door);
                g2.draw(knob);
        }
}
```

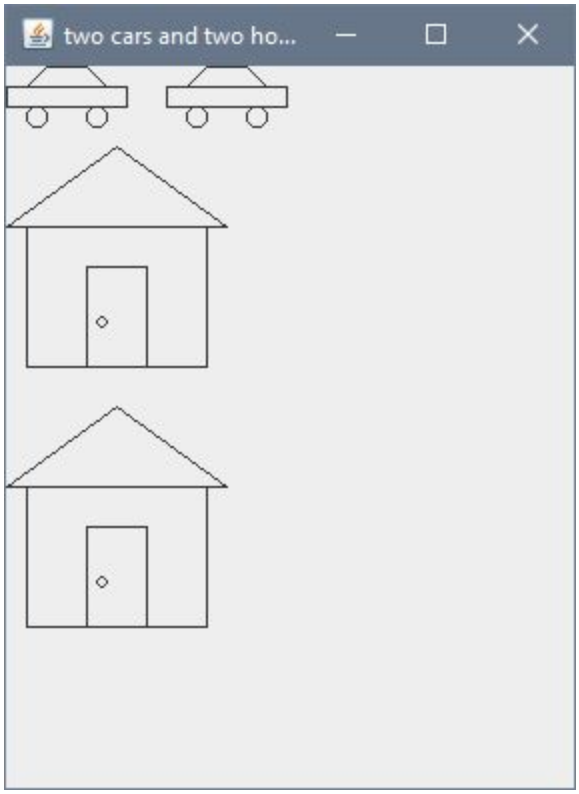
Exercise 4.3

```
/* Saket Bakshi, Period 6
Question 4.3 of Ch 3 project. This program creates a component class for printing
two houses and two cars.
*/

import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JComponent;

/**
        This component draws 2 cars and 2 houses
*/
public class Component extends JComponent
{
        public void paintComponent(Graphics g)
        {
                Graphics2D g2 = (Graphics2D) g;

                Car car1 = new Car(0,0);
                Car car2 = new Car(80,0);
                House house1 = new House(0,40);
                House house2 = new House(0,170);

                car1.draw(g2);
                car2.draw(g2);
                house2.draw(g2);
```

```
                house1.draw(g2);
        }
}
```

Exercise 4.4

```
/* Saket Bakshi, Period 6
Question 4.4 of Ch 3 project. This program views
two houses and two cars.
*/

import javax.swing.JFrame;

public class Viewer
{
        public static void main(String[] args)
        {
                JFrame frame = new JFrame();

                frame.setSize(300,400);
                frame.setTitle("two cars and two houses");
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                Component component = new Component();
                frame.add(component);

                frame.setVisible(true);
        }
}
```


two cars and two ho...