The science of computing: Java study questions

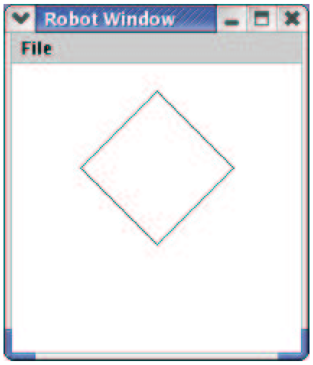
first edition

by Carl Burch

This document contains study questions to help in studying the material covered in the textbook, The science of computing: Java supplement. *Each question's label has two parts separated by a dash. Question J3–1, for example, is the first study question for the material covered in Chapter J3 of* The science of computing: Java supplement.

Question J2–1: (Solution, p JS11)

Write a program that draws a diamond in a window, as illustrated below. (The diamond need not be the same size or in the same position.)



```
import socs.*;

public class DrawDiamond {
    public static void main(String[] args) {
        RobotWindow win = new RobotWindow();
        win.show();











    }
}
```

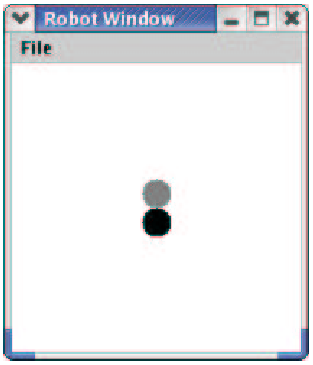
Question J2–2: (Solution, p JS11)

Suppose we had in our library a CircleStamp class with the following methods.

`CircleStamp(RobotWindow win, double d)`
: (Constructor method) Constructs a circle stamp for `win`, with a darkness level of `d`. The `d` parameter should be between 0.0 and 1.0, with 0.0 representing white and 1.0 representing black.

`void stamp(double x, double y)`
: Draws a circle with a radius of 10, centered at $(\mathtt{x}, \mathtt{y})$.

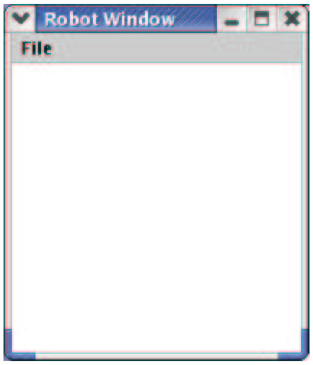
At right, complete the program so that it draws a gray circle on top of a black circle. *Your program must use CircleStamp to complete this problem.*



```
import socs.*;

public class DrawTwoCircle {
    public static void run() {
        RobotWindow win;
        win = new RobotWindow();
        win.show();












    }
}
```

Question J3–1: (Solution, p JS11)

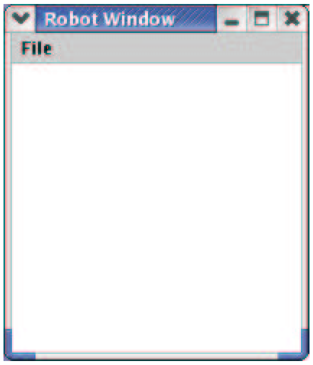
Suppose the user runs the Java program at right, typing 5 when told to choose a number. What would the computer draw?



```
import socs.*;

public class Mystery {
    public static void main(String[] args) {
        RobotWindow win;
        win = new RobotWindow();
        win.show();

        double n;
        n = win.requestInt();

        Robot robbie;
        robbie = new Robot(win, n, n);
        robbie.move(200 - 2 * n);
        robbie.turn(-90);
        robbie.move(200 - 2 * n);
        robbie.turn(-135);
        robbie.move(200 - 2 * n);
        robbie.switchOff();
    }
}
```

Question J3–2: (Solution, p JS12)

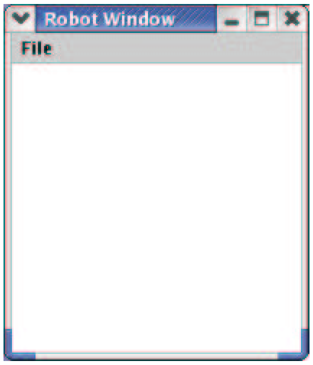
Suppose the user runs the Java program at right, typing 20 when told to choose a number. What would the computer draw?



```
import socs.*;

public class Mystery {
    public static void main(String[] args) {
        RobotWindow win = new RobotWindow();
        win.show();

        double n = win.requestDouble();
        Robot robbie = new Robot(win, 100, 100);
        robbie.move(n);
        robbie.turn(n);
        robbie.move(n);
        robbie.turn(n);
        robbie.move(n);
        robbie.turn(n);
        robbie.move(n);
        robbie.switchOff();
    }
}
```

Question J4–1: (Solution, p JS12)

Suppose a user runs the Java program at right, entering 5 when told. When the program ended, how would its window appear?



```
import socs.*;

public class Mystery {
    public static void main(String[] args) {
        RobotWindow win;
        win = new RobotWindow();
        win.show();

        int num;
        num = win.requestInt();
        int drawn;
        drawn = 1;
        while(drawn <= num) {
            Robot r2d2 = new Robot(win, 10, 20 * drawn);
            r2d2.move(20 * drawn);
            drawn = drawn + 1;
            r2d2.switchOff();
        }
    }
}
```

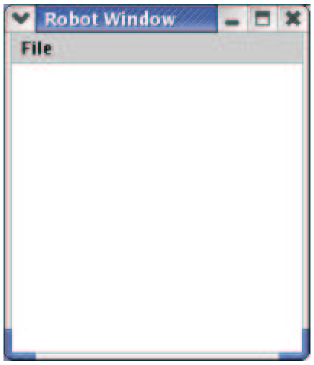
Question J4–2: (Solution, p JS12)

Suppose a user runs the Java program at right, entering 13, then 21, when told.

a. Show the values taken on by the following variables as the program runs.

`a`
`b`
`drawn`

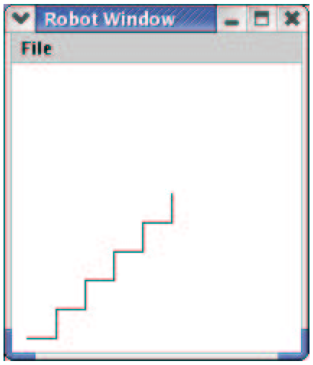
b. When the program ended, how would its window appear?



```
import socs.*;

public class Mystery {
    public static void main(String[] args) {
        RobotWindow win;
        win = new RobotWindow();
        win.show();

        int a = win.requestInt();
        int b = win.requestInt();
        int drawn = 1;
        Robot r = new Robot(win, 70, 100);
        while(drawn < 6) {
            r.move(b);
            drawn = drawn + 1;
            r.turn(90);
            int c = a + b;
            a = b;
            b = c;
        }
        r.switchOff();
    }
}
```

Question J4–3: (Solution, p JS13)

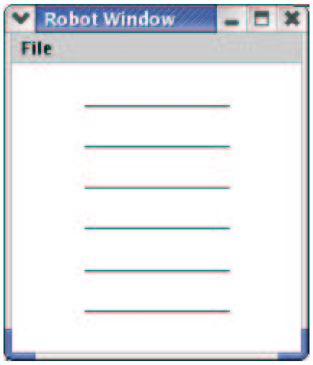
At right, complete the program so that it reads an integer from the user and draws a set of stairs with that many steps. For example, if the user were to type 5, the program's window should look like the following when the program completes.



```
import socs.*;

public class Steps {
    public static void run() {
        RobotWindow win;
        win = new RobotWindow();
        win.show();

        int steps;
        steps = win.requestInt();











    }
}
```

Question J4–4: (Solution, p JS13)

Write a program that reads an integer n from the user and draws n horizontal lines evenly spaced down the window, with each line extending from $x = 50$ to $x = 150$. For example, were the user to type 6, the program would display the following.



```
import socs.*;

public class DrawLines {
    public static void main(String[] args) {
        RobotWindow win = new RobotWindow();
        win.show();


















    }
}
```

Question J5–1: (Solution, p JS14)

Complete the program at right so that when run, it repeatedly reads integers from the user until the user enters 0. Then it should print the sum of the user's integers.

For example, if a user ran your program and entered 10, 2, 4, and 0, the user should see the following.

```
Number? 10
Number? 2
Number? 4
Number? 0
16
```

```
import socs.*;

public class PrintSum {
    public static void run() {
        IOWindow io = new IOWindow();

    }
}
```

Question J5–2: (Solution, p JS14)

Complete the program at right so that when run, it reads a line from the user and displays how many letters precede the first lower-case *a* in the string. (Your program may assume that the user types a line containing an *a*.)

For example, a user typing the program should see the following, assuming the user types what is in boldface.

```
? This is a test.
8
```

In this example, the program displays "8" because the user's string contains eight letters before the letter *a*.

```
import socs.*;

public class FindFirstA {
    public static void run() {
        IOWindow io = new IOWindow();

    }
}
```

Question J5–3: (Solution, p JS14)

Suppose the user ran the program at right and typed what is in boldface below.

```
? Java Programming
```

What would the program then print?

```
import socs.*;

public class Mystery {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("? ");
        String str = io.readLine();
        String a = str.substring(5, 9);
        String b = str.substring(1, 3)
        io.println(a + b);
    }
}
```

Question J5–4: (Solution, p JS14)

Suppose the user ran the program at right and typed 5 when prompted.

a. Show the sequence of values taken by the variables `num`, `i`, and `k`.

`num`
`i`
`k`

b. What does the program print?

```
import socs.*;

public class Mystery3 {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("? ");
        int num = io.readInt();
        int i = 1;
        int k = 0;
        while(i < num) {
            i = i + 1;
            k = k + i;
        }
        io.println(num + k);
    }
}
```

Question J5–5: (Solution, p JS14)

Suppose the user ran the program at right and typed 5 when prompted. What does the program print?

```
import socs.*;

public class Mystery4 {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("? ");
        int num = io.readInt();
        int i = 1;
        String k = "0";
        while(i < num) {
            i = i + 1;
            k = k + i;
        }
        io.println(num + k);
    }
}
```

Question J5–6: (Solution, p JS15)

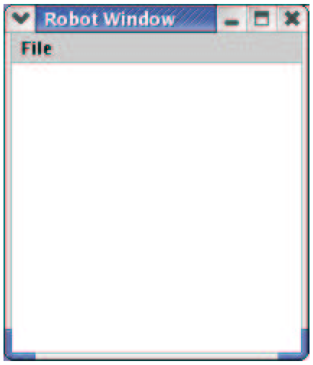
At right, complete the program so that it reads a line from the user and then prints the characters of that string, one character per line. For example, a user who runs the program and types “P Engel” should see the following

```
? P Engel
P

E
n
g
e
l
```

```
import socs.*;

public class WritingDown {
    public static void run() {
        IOWindow io = new IOWindow();









    }
}
```

Question J6–1: (Solution, p JS15)

Suppose a user runs the Java program at right. When the program ended, how would its window appear?



```
import socs.*;

public class Mystery {
    public static void run() {
        RobotWindow win = new RobotWindow();
        win.show();
        Robot rob = new Robot(win, 50, 50);
        rob.turn(45);
        int drawn = 0;
        while(drawn < 7) {
            rob.move(20);
            if(drawn % 2 == 0) {
                rob.turn(-90);
            } else {
                rob.turn(90);
            }
            drawn++;
        }
        rob.switchOff();
    }
}
```

Question J6–2: (Solution, p JS15)

Complete the program at right so that it reads a sequence of numbers ending in -1, whereupon it prints the number of even numbers typed.

I should see the following were I to run your program and enter the numbers 2, 77, -34, 104, and -1.

```
Number? 2
Number? 77
Number? -34
Number? 104
Number? -1
3
```

```
import socs.*;

public class CountEvens {
    public static void run() {
        IOWindow io = new IOWindow();









    }
}
```

Question J6–3: (Solution, p JS16)

Complete the program at right so that it reads two lines from the user and displays either “same” or “different” depending on whether they are the same. For example, a user typing the program should see the following, assuming the user types what is in boldface.

```
? First
? Second
different
```

```
import socs.*;

public class SameDifferent {
    public static void run() {
        IOWindow io = new IOWindow();










    }
}
```

Question J7–1: (Solution, p JS16)

Suppose we run the program at right and see the following. (Boldface indicates what the user types.)

```
: 1
: 3
: 2
: 4
: 0
```

What does the program print now?

```
import socs.*;

public class Mystery {
    public static void run() {
        IOWindow io = new IOWindow();
        int[] a = new int[5];
        int i = 0;
        while(i < 5) {
            io.print(": ");
            a[io.readInt()] = i;
            i++;
        }
        i = 4;
        while(i >= 0) {
            io.print(a[i]);
            i--;
        }
    }
}
```

Question J7–2: (Solution, p JS16)

Suppose the user ran the program at right and typed what is in boldface below.

```
3
5
2
4
1
```

What would the program then print?

```
import socs.*;

public class Mystery {
    public static void run() {
        IOWindow io = new IOWindow();
        int[] a = new int[5];
        int i = 0;
        while(i < 5) {
            a[i] = io.readInt();
            i++;
        }
        i = 0;
        while(i < 5) {
            i = a[i];
            io.println(i);
        }
    }
}
```

Question J8–1: (Solution, p JS16) Explain the difference between using an *instance method* in a Java program and using a *class method*.

Question J8–2: (Solution, p JS16)

Suppose we had in our library a StringUtil class with the following class method.

`static String reverse(String what)`

Returns a string with the same characters as `what`, except in reverse order.

At right, complete the program so that it reads a line from the user and displays “palindrome” or “not palindrome” depending on whether that line is a palindrome. (A **palindrome** is a word that reads the same forwards and backwards, such as *civic* or *noon*, but not *bad* or *palindrome*.)

For example, if a user should see the following on running your program and typing “good doog”.

```
? good doog
palindrome
```

Your program must use StringUtil’s `reverse` method to accomplish this task.

```
import socs.*;

public class Palindrome {
    public static void run() {
        IOWindow io = new IOWindow();









    }
}
```

Question J8–3: (Solution, p JS16)

Suppose we were to execute the method at right passing $\langle 5, 2, 7, 0, 6, 1\rangle$. for the array parameter `intArray`.

a. Show the sequence of values taken by the variables `k`, `sk`, and `i`.

```
k
sk
i
```

b. What value does the method return?

```
public static int myst(int[] intArray) {
    int k;
    int sk;
    if(intArray[0] > intArray[1]) {
        k = intArray[0];
        sk = intArray[1];
    } else {
        k = intArray[1];
        sk = intArray[0];
    }
    int i = 2;
    while(i < intArray.length) {
        if(intArray[i] > k) {
            sk = k;
            k = intArray[i];
        } else if(intArray[i] > sk) {
            sk = intArray[i];
        }
        i++;
    }
    return sk;
}
```

Question J8–4: (Solution, p JS17)

a. The Java method at right is set up to take an array of floating-point numbers as a parameter and return a floating-point number. Complete it so that it returns the minimum number in its parameter array.

```
import socs.*;

public class FindMin {
    public static double getMinimum(double[] nums) {







    }
}
```

b. Complete the Java program at right so that it uses the method of part **a.** to compute the minimum of the array `scores`, which it should then print on the IOWindow.

```
import socs.*;

public class UseFindMin {
    public static void run() {
        IOWindow io = new IOWindow();
        double[] scores = new double[10];
        int i = 0;
        while(i < scores.length) {
            scores[i] = io.readDouble();
            i++;
        }



    }
}
```

Question J8–5: (Solution, p JS17)

Suppose we were to execute the `run` method at right. What would the program print?

```
import socs.*;

public class Mystery {
    public static double f(double x) {
        return x + x;
    }
    public static double g(double y) {
        y = f(y);
        return y * f(y);
    }
    public static void run() {
        IOWindow io = new IOWindow();
        double z = 2.0;
        io.print(g(z));
        io.println(" " + z);
    }
}
```

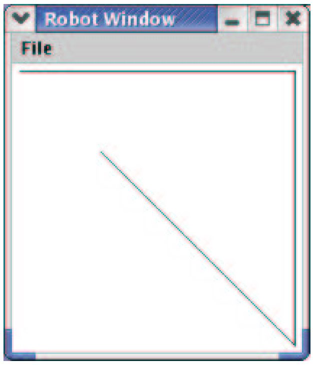
Solution J2–1: (Question, p JS1)

```
import socs.*;

public class DrawDiamond {
    public static void main(String[] args) {
        RobotWindow win = new RobotWindow();
        win.show();

        Robot diamond;
        diamond = new Robot(win, 100, 125);
        diamond.turn(45);
        diamond.move(75);
        diamond.turn(90);
        diamond.move(75);
        diamond.turn(90);
        diamond.move(75);
        diamond.turn(90);
        diamond.move(75);
        diamond.switchOff();
    }
}
```

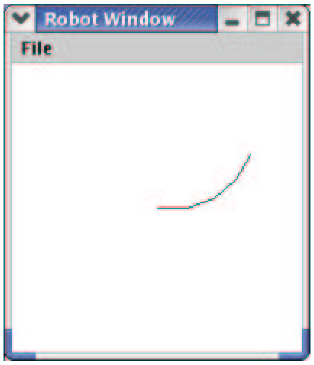
Solution J2–2: (Question, p JS1)

```
import socs.*;

public class DrawTwoCircle {
    public static void run() {
        RobotWindow win;
        win = new RobotWindow();
        win.show();

        CircleStamp gray;
        gray = new CircleStamp(win, 0.5);
        gray.stamp(100, 90);

        CircleStamp black;
        black = new CircleStamp(win, 1.0);
        black.stamp(100, 110);
    }
}
```

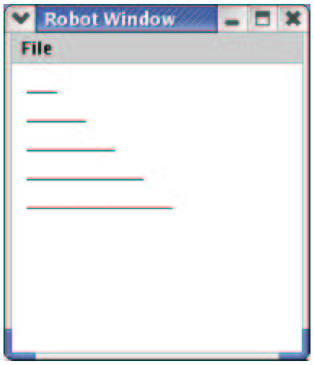
Solution J3–1: (Question, p JS2)



Solution J3–2: (Question, p JS2)



Solution J4–1: (Question, p JS3)

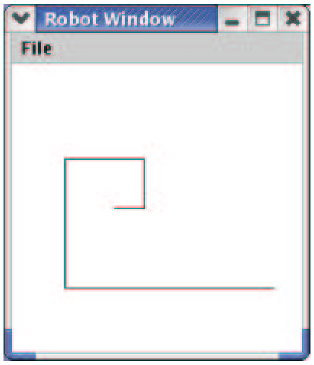


Solution J4–2: (Question, p JS3)

a.

`a`	13 21 34 55 89 144
`b`	21 34 55 89 144 233
`drawn`	1 2 3 4 5 6

b.



Solution J4–3: (Question, p JS4)

```
import socs.*;

public class Steps {
    public static void run() {
        RobotWindow win;
        win = new RobotWindow();
        win.show();

        int steps;
        steps = win.requestInt();

        Robot r = new Robot(win, 10, 190);
        int i;
        i = 0;
        while(i < steps) {
            r.move(20);
            r.turn(90);
            r.move(20);
            r.turn(-90);
            i++;
        }
        r.switchOff();
    }
}
```

Solution J4–4: (Question, p JS4)

```
import socs.*;

public class DrawLines {
    public static void main(String[] args) {
        RobotWindow win = new RobotWindow();
        win.show();

        int num = win.requestInt();
        double gap = 200.0 / (num + 1);
        int y = gap;
        while(y < 200.0) {
            Robot line = new Robot(win, 50, y);
            line.move(100.0);
            line.switchOff();
            y += gap;
        }
    }
}
```

Solution J5–1: (Question, p JS5)

```
import socs.*;

public class PrintSum {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("Number? ");
        int num = io.readInt();
        int sum = 0;
        while(num != 0) {
            sum += num;
            io.print("Number? ");
            int num = io.readInt();
        }
        io.println(num);
    }
}
```

Solution J5–2: (Question, p JS5)

```
import socs.*;

public class FindFirstA {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("? ");
        String line = io.readLine();
        int index = 0;
        while(!(line.substring(index, index + 1)).equals("a")) {
            index++;
        }
        io.println(index);
    }
}
```

Solution J5–3: (Question, p JS5) `Progav`

Solution J5–4: (Question, p JS6)

a.

`num`	5
`i`	1 2 3 4 5
`k`	0 2 5 9 14

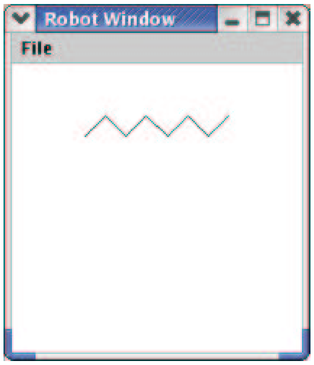
b. 19

Solution J5–5: (Question, p JS6) `502345`

Solution J5–6: (Question, p JS6)

```
import socs.*;

public class WritingDown {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("? ");
        String line = io.readLine();
        int i = 0;
        while(i < line.length()) {
            String letter = line.substring(i, i + 1);
            io.println(letter);
            i++;
        }
    }
}
```

Solution J6–1: (Question, p JS7)



Solution J6–2: (Question, p JS7)

```
import socs.*;

public class CountEvens {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("Number? ");
        int num = io.readInt();
        int count = 0;
        while(num != -1) {
            if(num % 2 == 0) {
                count++;
            }
            io.print("Number? ");
            num = io.readInt();
        }
        io.println(count);
    }
}
```

Solution J6–3: (Question, p JS7)

```
import socs.*;

public class SameDifferent {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("? ");
        String a = io.readLine();
        io.print("? ");
        String b = io.readLine();
        if(a.equals(b)) {
            io.println("same");
        } else {
            io.println("different");
        }
    }
}
```

Solution J7–1: (Question, p JS8) `31204`

Solution J7–2: (Question, p JS8)

```
3
4
1
5
```

Solution J8–1: (Question, p JS8) Instance methods are messages that are sent to objects of a class; for example, since `move` is an instance method of a Robot class, we can send the `move` message to an individual Robot object (created using `new`). Class methods, however, apply to the class itself: That is, a class method is a message we send to the class, not to objects of that class. Since `pow` is a class method of the Math class, we send it to the Math class, not to individually created Math objects.

Solution J8–2: (Question, p JS9)

```
import socs.*;

public class Palindrome {
    public static void run() {
        IOWindow io = new IOWindow();
        io.print("? ");
        String s = io.readLine();
        String r = StringUtil.reverse(s);
        if(s.equals(r)) {
            io.println("palindrome");
        } else {
            io.println("not palindrome");
        }
    }
}
```

Solution J8–3: (Question, p JS9)

a.

`k`	5 7
`sk`	2 5 6
`i`	2 3 4 5 6

b. 6

Solution J8–4: (Question, p JS10)

a.

```
import socs.*;

public class FindMin {
    public static double getMinimum(double[] nums) {
        double min = nums[0];
        int i = 0;
        while(i < min) {
            if(nums[i] < min) {
                min = nums[i];
            }
            i++;
        }
        return min;
    }
}
```

b.

```
import socs.*;

public class UseFindMin {
    public static void run() {
        IOWindow io = new IOWindow();
        double[] scores = new double[10];
        int i = 0;
        while(i < scores.length) {
            scores[i] = io.readDouble();
            i++;
        }

        double min = FindMin.getMinimum(scores);
        io.println("Minimum is " + min);
    }
}
```

Solution J8–5: (Question, p JS10) `32.0 2.0`