Here is a list of things a computer should do to perform the task of calculating what someone should be paid for a baby sitting job.

1. Display a message "Type in your wage rate in dollars per hour:"
2. Set up computer memory for a copy of the rate for future use.
3. Set up the Java runtime library to accept input from a person.
4. Allow the user to enter the wage rate.
5. Once the user enters a number, store it in memory.
6. Display a message: "Type in how many hours you sat those babies:"
7. Set up computer memory for a copy of the number of hours for future use.
8. Allow the user to enter the number of hours worked.
9. Once the user enters a number, store it in memory.
10. Set up memory for remembering the answer before printing it.
11. Once both the wage rate and the number of hours worked are available in memory, multiply them and store the result in memory.
12. Compose the String of exactly what to write on the check, by combining literal text with the result expressed in base ten (decimal) digits.
13. Display a message that shows the amount of money earned in the form of literal text with the result expressed in base ten (decimal) digits.

I purposely wrote these a bit inexactly because people do fine figuring out what to do based on understanding the intentions of the person who wrote the instructions. I also wrote these 13 instructions because each one, in order, corresponds to a particular command to the computer that you must translate into Java for the computer to do the job. Here is the program: (I actually tested it!)

```java
public class BabySittingWageCalculator
{
    public static void main(String[] a)
    {
        System.out.println("Type in your wage rate in dollars per hour:");
        double wageRate;
        java.util.Scanner sc = new java.util.Scanner( System.in );
        wageRate = sc.nextDouble();
        System.out.println("Type in how many hours you sat those babies:");
        double hoursSat;
        hoursSat = sc.nextDouble();
        double wages;
        wages = wageRate * hoursSat;
        String checkToPrint = "Pay to the order of cash "+ wages + "\n Signed, Prof. Chaiken";
        System.out.println(checkToPrint);
    }
}
```

Looks scary?? On the next page, I interspersed the each English direction with the line of Java code that tells the computer to carry out the English written directly above it.
public class BabySittingWageCalculatorCommented
{
    public static void main(String[] a)
    {
        1. Display a message "Type in your wage rate in dollars per hour:".
        System.out.println("Type in your wage rate in dollars per hour: ");
        2. Set up computer memory for a copy of the rate for future use.
        double wageRate;
        3. Set up the Java runtime library to accept input from a person.
        java.util.Scanner sc = new java.util.Scanner( System.in );
        4. Allow the user to enter the wage rate.
        5. Once the user enters a number, store it in memory.
        wageRate = sc.nextDouble();
        6. Display a message: "Type in how many hours you sat those babies:" 
        System.out.println("Type in how many hours you sat those babies:" );
        7. Set up computer memory for a copy of the number of hours for future use.
        double hoursSat;
        8. Allow the user to enter the number of hours worked.
        9. Once the user enters a number, store it in memory.
        hoursSat = sc.nextDouble();
        10. Set up memory for remembering the answer before printing it.
        double wages;
        11. Once both the wage rate and the number of hours worked are available
            in memory, multiply them and store the result in memory.
        wages = wageRate * hoursSat;
        12. Compose the String of exactly what to write on the check, by combining
            literal text with the result expressed in base ten (decimal) digits.
        String checkToPrint =
            "Pay to the order of cash "$ + wages + "\n Signed, Prof. Chaiken";
        13. Display a message that shows the amount of money earned in the form of
            what should be written on your check.
        System.out.println(checkToPrint);
    }
}

If you copy and paste the above into DrJava, save and try to compile, it will fail horribly because
intermixed Java and English does not follow the syntax rules of the Java language.

One syntax rule is about comments. If you precede a line in your Java file with two forward-slashes in
a row, then that entire line will be ignored when the compile step analyzes the syntax. Such a line
however will be printed along with the other Java lines when you print the Java program file. Those
lines are called comments. You may want to precede some of the original Java you will write with
comments to help you remember what you wish the Java code will accomplish. But, a comment is
never a guarantee that the Java code will mean what the comment says! Anything you write in a
comment is utterly ignored by the computer when it comes time to run your program.

On the next page is the program in legal Java with the English directions explaining the purpose for
each Java line delimited as a comment. You CAN copy the code with comments, paste it into DrJava,
save it; and it will compile and run fine. (I tested it too and then copied it into this document.)
public class BabySittingWageCalculatorCommented {
    public static void main(String[] a) {
        // 1. Display a message "Type in your wage rate in dollars per hour:"
        System.out.println("Type in your wage rate in dollars per hour:");
        // 2. Set up computer memory for a copy of the rate for future use.
        double wageRate;
        // 3. Set up the Java runtime library to accept input from a person.
        java.util.Scanner sc = new java.util.Scanner(System.in);
        // 4. Allow the user to enter the wage rate.
        // 5. Once the user enters a number, store it in memory.
        wageRate = sc.nextDouble();
        // 6. Display a message: "Type in how many hours you sat those babies:
        System.out.println("Type in how many hours you sat those babies:");
        // 7. Set up computer memory for a copy of the number of hours for future use.
        double hoursSat;
        // 8. Allow the user to enter the number of hours worked.
        // 9. Once the user enters a number, store it in memory.
        hoursSat = sc.nextDouble();
        // 10. Set up memory for remembering the answer before printing it.
        double wages;
        // 11. Once both the wage rate and the number of hours worked are available
        // in memory, multiply them and store the result in memory.
        wages = wageRate * hoursSat;
        // 12. Compose the String of exactly what to write on the check, by combining
        // literal text with the result expressed in base ten (decimal) digits.
        String checkToPrint = "Pay to the order of cash $" + wages + "
        Signed, Prof. Chaiken";
        // 13. Display a message that shows the amount of money earned in the form of
        // what should be written on your check.
        System.out.println(checkToPrint);
    }
}

In this course, do not bother to memorize details like "java.util.Scanner sc = new
java.util.Scanner(System.in); wageRate = sc.nextDouble();" is the Java
way to make the computer accept a number when a person types it in. Instead, write details like this on a
reminder sheet that you can keep and consult all the time: In lectures, labs, doing homework, and, in this
course, when you take the in-class midterm and final. (Exams: closed everything but 1 note sheet.)

But, you MUST remember "= means copy". Here are examples of the questions you MUST be able to answer
during exams, quizzes and in your own mind when you are trying to figure out how programming works:

In the code wageRate = sc.nextDouble(); which part allows the user to enter the wage rate?
A) sc.nextDouble()
B) wageRate =

In the code wageRate = sc.nextDouble(); which part directs that once the user enters a
number, store it in memory?
A) sc.nextDouble()
B) wageRate =

(Yes, the computer does these two things right-to-left, backwards! Correct answers: first A) second B) )