expect in each case. For example, if your program calculates your pay, you should check three different cases: less than 40 hours, 40 hours, and more than 40 hours. Calculate each case by hand before running your program so that you know what to expect. You may get a correct answer for one case, but not for another case. This will help you figure out where your logic errors are.

3. Run time errors—errors that do not occur until the program is run, and then may only occur with some data. These errors emphasize the need for completely testing your program.

Task #1 Writing an Algorithm

1. Copy the file *Pay.java* (see Code Listing 1.1) from the Student Files or as directed by your instructor.
2. Open the file in your Java Integrated Development Environment (IDE) or a text editor as directed by your instructor. Examine the file, and compare it with the detailed version of the pseudocode in step number 3, section 1.6 of the textbook. Notice that the pseudocode does not include every line of code. The program code includes identifier declarations and a statement that is needed to enable Java to read from the keyboard. These are not part of actually completing the task of calculating pay, so they are not included in the pseudocode. The only important difference between the example pseudocode and the Java code is in the calculation. Below is the detailed pseudocode from the example, but without the calculation part. You need to fill in lines that tell in English what the calculation part of `Pay.java` is doing.

Display "How many hours did you work?".
Input hours.
Display "How much are you paid per hour?".
Input rate.
If Input hours is equal or less than 40 then calculate pay by multiplying hours and rate;
otherwise calculate the pay by subtracting it(total hr) by 40 and Multiply it by 1.5 rate then add the regular pay (40 x rate) to the final amount.

Display the value in the pay variable.

Task #2 Compile and Execute a Program

1. Compile `Pay.java` using the JDK or a Java IDE as directed by your instructor.
2. You should not receive any error messages.
3. When this program is executed, it will ask the user for input. You should calculate several different cases by hand. Since there is a critical point at which the calculation changes, you should test three different cases: the critical point, a number above the critical point, and a number below the critical point. You want to calculate by hand so that you can check the logic of the program. Fill in the chart below with your test cases and the result you get when calculating by hand.
4. Execute the program using your first set of data. Record your result. You will need to execute the program three times to test all your data. Note: you do not need to compile again. Once the program compiles correctly once, it can be executed many times. You only need to compile again if you make changes to the code.

Hours	Rate	Pay (hand calculated)	Pay (program result)
20	35.50	710	710
40	35.50	1420	1420
60	35.50	2485	2135

Task #3 Debugging a Java Program

1. Copy the file *SalesTax.java* (see Code Listing 1.2) from the Student Files or as directed by your instructor.
2. Open the file in your IDE or text editor as directed by your instructor. This file contains a simple Java program that contains errors. Compile the program. You should get a listing of syntax errors. Correct all the syntax errors, you may want to recompile after you fix some of the errors.
3. When all syntax errors are corrected, the program should compile. As in the previous exercise, you need to develop some test data. Use the chart below to record your test data and results when calculated by hand.
4. Execute the program using your test data and recording the results. If the output of the program is different from what you calculated, this usually indicates a logic error. Examine the program and correct any logic errors. Compile the program and execute using the test data again. Repeat until all output matches what is expected.

Item	Price	Tax	Total (calculated)	Total (output)
Candel	5 $	0.275	5.275	25.275
book	100 $	5.5	105.5	10005.5

Code Listing 1.1 (Pay.java)

```
import java.util.Scanner;  // Needed for the Scanner class

/**
   This program calculates the user's gross pay.
*/

public class Pay
{
   public static void main(String[] args)
   {
      // Create a Scanner object to read from the keyboard.
      Scanner keyboard = new Scanner(System.in);

      // Identifier declarations
      double hours;  // Number of hours worked
      double rate;   // Hourly pay rate
      double pay;    // Gross pay

      // Display prompts and get input.
      System.out.print("How many hours did you work? ");
      hours = keyboard.nextDouble();
      System.out.print("How much are you paid per hour? ");
      rate = keyboard.nextDouble();

      // Perform the calculations.
      if(hours <= 40)
         pay = hours * rate;
      else
         pay = (hours - 40) * (1.5 * rate) + (40 * rate);

      // Display results.
      System.out.println("You earned $" + pay);
   }
}
```

Code Listing 1.2 (SalesTax.java)

```
import java.util.Scanner;  // Needed for the Scanner class

/**
   This program calculates the total price which includes
   sales tax.
*/

public class SalesTax
{
   public static void main(String[] args)
   {
      // Identifier declarations
      final double TAX_RATE = 0.055;
      double price;
      double tax;
      double total;
      String item;

      // Create a Scanner object to read from the keyboard.
      Scanner keyboard = new Scanner(System.in);

      // Display prompts and get input.
      System.out.print("Item description:  ");
      item = keyboard.nextLine();
      System.out.print("Item price:  $");
      price = keyboard.nextDouble();

      // Perform the calculations.
      tax = price * TAX_RATE;
      total = price + tax;

      // Display the results.
      System.out.print(item + "        $");
      System.out.println(price);
      System.out.print("Tax          $");
      System.out.println(tax);
      System.out.print("Total        $");
      System.out.println(total);
   }
}
```