

After all your hard work you have finally earned your Engineering degree! You find an exciting job at a top firm offering full benefits and a great salary. As a preliminary test of your basic abilities your boss asks you to design a few different circuits using “and” gates, “or” gates and encoders. Along with the circuits he also wants you to create a program in Java that will be able to keep track of and print the descriptions of the different circuits you have created.

This project will require you to write three classes and a few methods for each one. Please note that you will *not* actually be creating or designing any circuits, only the Java program to store and print their information. The following paragraphs will contain hypothetical circuit descriptions followed by a way in which to use these specifications to write a Java program. Classes will be in shown in bold, variables in italics, and methods will be underlined. Please write your program as described to receive full credit.

Part 1:

First your boss wants you to design a few circuits that implement Boolean functions using encoders. For each circuit he will give you the number of functions to be implemented as well as the number of inputs to its encoder. Your Java program should be able to calculate the number of outputs for the circuit’s encoder and then print out a description of each circuit designed with an encoder.

- Create a class named **Encoder**.
- Declare three integer class variables; *inputs*, *outputs*, and *functions*. Do not initialize them.
- Write a constructor that accepts two integer variables. One should represent the number of inputs and the other the number of functions to be implemented by a circuit built with an encoder. Within the constructor the class variables *inputs*, *outputs* and *functions* should be initialized. Hint: Recall the mathematical relationship between the number of inputs and outputs for an encoder.
- Write a method entitled description that accepts no parameters and returns nothing. This method should print out the specifications of the circuit including the number of inputs and outputs of the circuit’s encoder as well as the number of functions implemented by the circuit. The output should be reader friendly clearly defining what each value printed represents.
- Use the main method to validate the functionality of the methods in the **Encoder** class: In the main method create an **Encoder** object and pass the values 3 and 2 to its constructor. Use the reference to this object to call the description method. Make sure the proper values are printed.

Part 2:

Next your boss wants you to design several circuits that implement Boolean functions using “and” and “or” gates. You will be given the number of inputs and functions to be implemented by each circuit as well as the number of “or” gates available to construct each circuit. As for the number of “and” gates, your boss states that there are somewhere between 15 and 30 “and” gates available for each circuit. Your Java program should print out a description of each circuit designed with “and” and “or” gates.

- Create a class named **Gates**. This class will be similar to the previous class, however, there will be some notable differences to show the various ways in which variables can be accessed within and between classes.
- Declare three integer class variables; *inputs*, *functions*, and *andgates*. Also declare an integer class variable with private visibility, *orgates*. *Orgates* must be private to receive credit for this part.
- Write a constructor that accepts two integer variables. One should represent the number of inputs and the other the number of functions to be implemented by a circuit built with gates. Within the constructor the class variables *inputs*, *functions* and *andgates* should be initialized. *Andgates* should be initialized to a random number between 15 and 30. You will need to use the Random class for this part. Refer to the lecture notes and textbook for help creating and using random values.

- Write two methods; setor and getor. Setor should take an integer value, assign it to the private variable *orgates* and return nothing. Getor should simply return the value of the private variable *orgates* and accept nothing. These two methods serve as an example of a “mutator” and “accessor.” (See lecture notes and book).
- Write a method entitled description that accepts no parameters and returns nothing. It should print out the specifications of the circuit, including the number of inputs, “and” gates and “or” gates available to build the circuit, and the number of functions implemented by the circuit. The output should be reader friendly clearly defining what each value printed represents.
- Use the main method to validate the functionality of the methods in the **Gates** class:
In the main method create a **Gates** object and pass the values 4 and 7 to its constructor. Use the reference to this object to call the setor method and pass it the value 14. Also use the reference to call the description method. Verify that all values meet the specifications.

Part 3:

Your boss gives you the specifications of several circuits he wants you to design and you assure him that you can add to your program so that it will print the descriptions of the circuits. The circuits are as follows:

| Type | Inputs | Outputs | Functions | "And" Gates | "Or" Gates |
|---------|--------|---------|-----------|-------------|------------|
| Encoder | 4 | TBD | 5 | NA | NA |
| Encoder | 8 | TBD | 23 | NA | NA |
| Gates | 3 | NA | 4 | 15-30 | 23 |
| Gates | 7 | NA | 5 | 15-30 | 35 |

Your boss requests that you add the circuits then print out a description of each.

- Create a class named **Circuits**
- Within the main method create two **Encoder** objects and two **Gates** objects. Pass to their constructors the integers given in the above table.
- Use the object references to call the remaining necessary methods to print out descriptions for each circuit.