

# Java Logic Flow Chart Practice Exercise

This exercise was designed for use in Introductory Java programming courses, specifically CIT-111: Introduction To Programming Java taught at CCAC.

## Introduction

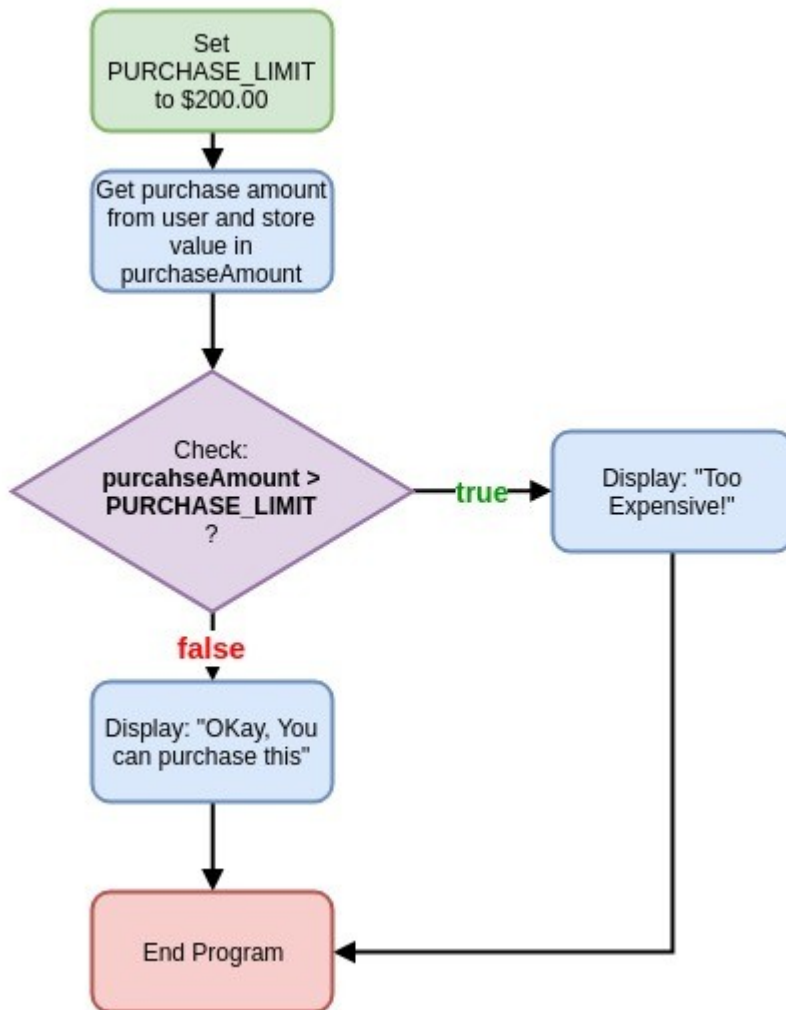
The power of programming comes from writing logic into our code that can test input values for various characteristics (Example: Is the input higher than number x? Is it equal to a stored password?).

Imagine we're writing code for point-of-sale systems. Let's write some new code that implements some basic logic that warns the user of any final sale amount that is over a certain threshold amount, such as purchase limit on a credit card designed for youth learning to manage their money.

**Program specification:** Create a program that stores an internal purchase limit set by the programmer. The user is then asked for a purchase amount. Finally, the program outputs a prompt to buy something else if the purchase amount is above the limit. If the user's purchase amount is below the limit, display a confirmation that the purchase is allowed. and its explanation.

To achieve this specification, we need to think carefully about what each part of code must do. Here's a sample flow chart of how the above specification might be planned out. Please grab a sheet of paper and as you study this diagram, create it for yourself on real paper:

**Figure 1: Program logic flow chart**



Notice that each task in our program is summarized in a **rounded rectangle**. Any decisions that our program must make are declared in a **diamond**. The flow coming from this diamond corresponds to each of the possible outcomes of the test. In our case, the if statement can work with outcomes of true and false.

Create a class called IfControl and try coding up the above flow chart in your main method.

## Possible Java Implementation

The defined specification can be coded in many ways. The following Java code is an example implementation of the logic in Figure 1.

Access the source on GitHub:

[https://github.com/edarsow/javaCIT111/blob/master/cit111NetBeans/src/cis\\_program\\_assessment/IfControl.java](https://github.com/edarsow/javaCIT111/blob/master/cit111NetBeans/src/cis_program_assessment/IfControl.java)

```
import java.util.Scanner;

/**
 * Compares a user-entered value against a pre-set purchase limit to
 * model a banking training program for you learning to manage money.
 * @author Eric Darsow
 */
public class IfControl {

    // this is the entry point for our program
    public static void main(String[] args) {

        // create a purchase limit as a final variable since it won't change
        final double PURCHASE_LIMIT = 200.00;
        // declare a double-type variable to store the user's purchase amount
        double purchaseAmount;
        // create a Scanner object to get values from the keyboard
        Scanner userInputScanner = new Scanner(System.in);

        // prompt the user for a purchase amount
        System.out.println("Enter the amount of the purchase (no $) and press
enter:");
        // read whatever the user typed on the keyboard into the variable
        purchaseAmount
        purchaseAmount = userInputScanner.nextDouble();

        // compare the user-entered purchase amount against the purchase limit
        if(purchaseAmount > PURCHASE_LIMIT){
            // code to run if the the comparison evaluates to true (purchase is
too high)
            System.out.println("Ah Ah ah--That's too expensive! Buy something
else");

        } else {
            // code to run if the comparison evaluates to false
            System.out.println("Okay, you can purchase this item for: $" +
purchaseAmount);
        } // close if/else block

    } // close main method
} // close class IfControl
```