

Lab Exercise 2 – Conditional Statements

So far, we have written programs that are linear – Java runs each line of your code in sequential order. However, there are situations in programming where you need to run particular pieces of code depending on rules and situations required of the application.

For this lab exercise, you will write an extension to your “Mobile Phone” program (Lab 1). Your program should be able to perform the functions described in Lab 1 (including text messaging) and be able to enforce constraints on the supplied data.

1. Open up BlueJ and create a *MobilePhone* class supports the following methods (refer to Lab 1 for more details):

- `void load(double pesos)`
- `void call(int minutes)`
- `void sendTextMessage()`
- `double getLoadLeft()`
- `int getTotalMinutesCalled()`
- `int getNumTextMessages()`
- `void changeRate(double newCallRate)`
- `void printSummary()`

You may also implement the following method (optional):

- `void passLoadTo(MobilePhone dest, double amount) // see #3`

2. Use if statements to implement the following constraints:

Rule 1: Calls and text messages that would cost more than your remaining credits are forbidden

- Optional: allow the call but only up to what the remaining load could handle

Rule 2: Call rates cannot be set to more than P10.00/minute

Rule 3: There is a P25.00 minimum when loading credits

Rule 4: Operations with negative numbers (loading, calling, changing call rates) are forbidden

It is good practice that you do not print error messages from within the methods of *MobilePhone*. Rather, you should return values to the class/method that uses *MobilePhone* and let that procedure handle the error. This task is optional, described in #4.

3. *Optional:* When the *passLoadTo* method is called, credits are transferred from one mobile phone to the other. There is a P1.00 service charge for this service. For example:

```
MobilePhone alice = new MobilePhone();
MobilePhone bob = new MobilePhone();

alice.load( 300.00 ); // alice - P300.00; bob - P0.00
alice.passLoadTo( bob, 10.00 ); // alice - P289.00; bob - P10.00

bob.passLoadTo( alice, 10.00 ); // forbidden - total cost is
// greater than credits
```

Note that attempting to use this service is forbidden if the total cost (*amount to be passed + service charge*) is greater than the remaining credits of the source mobile phone, or the amount to be passed is less than P1.00 ($P1 \leq \text{amount} \leq \text{total cost}$).

4. *Optional:* You can also change the return value of your methods to indicate whether the

operation was successful or forbidden. As a simple exercise, change the following method signatures in your `MobilePhone` class from *void* to *boolean*:

- `boolean load(double pesos)`
- `boolean call(int minutes)`
- `boolean sendTextMessage()`
- `boolean changeRate(double newCallRate)`

When a method call violates one of the rules indicated in #2, your method should return the boolean value *false*. Otherwise, it should return *true*. For example, if you attempt to load P10.00, it should return *false* since it would violate rule 3.

Printing peso amounts should be prefixed with a 'P' and should be accurate to exactly 2 decimal places.

To test your program, download the `Driver.java` program from <http://curry.ateneo.net/~jpv/> and place this to your BlueJ project folder. There are two `Driver.java` programs available, one with error message printing (as described in #4) and one without.

Archive your BlueJ project directory and name it as `[Surname]-[ID Number].zip` and submit it through Moodle (or as prescribed by your instructor).