**This program is to be the sole effort of the individual turning it in. No collaborative efforts or code copying are permitted. You may not discuss this program with anyone except the Tas or instructor.**

**Cookie Time!**

Due: Tuesday, December 10, 11:59 p.m.

ALL METHODS AND CLASSES MUST BE NAMED EXACTLY AS SPECIFIED (INCLUDING UPPER AND LOWER CASE) AND MUST ACCEPT EXACTLY THE PARAMETERS SPECIFIED IN THE ORDER SPECIFIED)

**Objective:** To gain additional experience using arrays. **An important part of this program is to save information in appropriate types of variables and data structures. So, for instance if it is most appropriate to save values in an array, that is how they should be saved. If a variable is a constant value that should not change, it should be saved appropriately.**

**Program:** You are given the task of creating an object to keep track of cookie sales for Girl Scout troops. The class definition MUST be NAMED **CookieCounter**. Each CookieCounter object is to represent one troop and when an instance of a CookieCounter is created it will be passed an integer which is the troop number. The troops sell six different types of cookies, and each type is designated by a number as follows:

1. Thin Mints
2. Shortbread
3. Peanut Butter
4. Macaroons
5. Lemon Drop
6. Snickerdoodles

The class is to keep track of the total number of boxes of each type of cookie that has been sold by the troop represented by the object. In addition, the total number of each type of cookie sold by all the troops combined troop is also to be maintained.

The CookieCounter class is to provide the following public methods:

**countSale** - will take two integer parameters, the first is a number indicating the type of cookie sold and the second is the number of boxes of that type of cookie that were sold. The method should use this information to maintain a running total of sales for the type of cookie specified by the first parameter. If an invalid cookie type number has been passed to the method, an error message should be output that includes the invalid value.

**displayTroopSales** - this method can take either no parameters or one integer parameter. If no parameters are passed, the method should output the name of each type of cookie and the total sales of each type of cookie in a clear and easily readable format. If one parameter is passed, the number of boxes sold of the type of cookie indicated by the parameter should be output in a clear and easily readable format. Regardless of the number of parameters passed, the troop number should be included in the output. If an invalid cookie type number has been passed to the method, an error message should be output that includes the invalid value.

**displayAllSales** – this method should display the total sales of each type of cookie made by all of the troops combined in a clear and easily readable format. The name of the cookie is to be displayed along with its corresponding sales information.

**displayTroopBestseller** – this method should display the troop number and the cookie name and the number of boxes sold of the cookie that has currently the highest number of boxes sold. Note: there may be multiple best sellers. Consider for example if 100 boxes of all types of cookies have been sold but there were 140 boxes sold of both Thin Mint and Macaroons. Your method should output both Thin Mints and Macaroons.

The CookieCounter class should also provide the following private method, that will be used by other methods inside of he class for error checking:

**validCookieNumber** – This method takes one integer parameter and returns a boolean value (that is a value that can be true or false). It will return a value of true is the integer parameter represents a valid cookie type number and a false otherwise. The method should be invoked by other methods to test the validity of a cookie type numbers which have been passed as parameters.

You should include a separate class containing your main and sufficient method calls to test your program. We will be substituting our own main when we run your program for testing and a significant deduction will be taken for any errors that occur because your methods have not been named or accepted parameters as specified.

Drop both your Cookie Counter class and your main to moodle when complete.