Teller Machine

1. Create a new class that represents a *Teller Machine*.
2. Add a *manufacturer, deposits, withdrawals, and balance* attribute/property to the Teller Machine class:
    - `manufacturer`: indicates the manufacturer name for the teller machine.
    - `deposits`: indicates the total amount that has been deposited into the machine.
    - `withdrawals`: indicates the total amount that has been withdrawn from the machine.
    - `balance`: indicates the net difference between `deposits` and `withdrawals`.
3. Create a constructor that accepts `manufacturer`, `deposits`, and `withdrawals`.
4. Instantiate an object (or objects) in *main()* or *Main()* and use the object(s) to test your methods.
5. Create a method that checks to see if a string, `cardNumber`, is a valid card. The method only returns `true` under the following conditions:
    - if the `cardNumber` begins with a `5` and has 16 digits
    - if the `cardNumber` begins with a `4` and has 13 or 16 digits
    - if the `cardNumber` begins with a `3` and is followed by a `4` or a `7`.
6. Override the *ToString*()/*toString()* method and have it return `"ATM – {manufacturer} – {balance}"` where `{manufacturer}` and `{balance}`. The {} are placeholders for the actual values. i.e. the values from the object should be shown in the string where the {} are indicated.
7. Implement unit tests to validate the functionality of:
    - the balance calculation
    - the valid card number method
8. In the main program class, within the main method, read in the provided csv file `TellerInput.csv` and use it to populate a list of *Teller Machine* objects.
9. Add up the total balance for all of the teller machines in the list and print it to the screen.