JAVA CODING EXERCISE

We are passionate about code and hope you are too! We'd love to see how you implement a common business requirement - so please choose one requirement from the list below, code up a solution and return it to us so we can have a look at it.

Please see below for more detailed instructions.

Good luck!

/The Expedia Software Development Team, Montreal

This test has been designed to evaluate your coding style and capabilities. We do not recommend investing more than **4 hours** but if you do, it does not guarantee that you will go to the next step in the interview process. The expectation is not that the entire problem is solved perfectly and completely. What we expect, is that when you stop, you document all limitations of your work and the reasoning behind each of them. The premise is also that the code you will do will be used in a **production environment**, which means to use the best approach to demonstrate that.

Instructions

1. Choose one of the Business Requirements below
2. Develop a solution that solves the Business Requirement and make sure that any assumption is documented and that your code is production-ready.
3. Use **Java** as your main language but be original. We want to see how you code so, avoid “copy/paste” from Google and try to solve the problem your own way. If you use “copy/paste code” when solving the main problem, please make sure to document the reference and why you did it.
4. Once the code is to your satisfaction - follow the steps below (we have put a lot of effort to make original exercises, please **do not distribute your results publicly** to give everyone a fair chance):
    a. Put all your code in a folder named with your first and last name (no space, e.g. JohnDoe).
    b. Zip your folder into a single file (e.g. JohnDoe.zip).
    c. Upload your file using https://uploadfiles.io/.
    d. Send an email to mtlrecruiting@expedia.com with:
        i. Subject set to: "*Java Coding Exercise from <insert name here>*"
        ii. In the email’s body: the name of the agency representing you if are a contractor, or mentioning “This is for an FTE position” if not.

Requirement #1: Flight Search

Develop an application that will give you available flights for a given search query. You have an inventory of the following flights:


```
{
	"flights": [
		{
			"flight": "Air Canada 8099",
			"departure": "7:30AM"
		},
		{
			"flight": "United Airline 6115",
			"departure": "10:30AM"
		},
		{
			"flight": "WestJet 6456",
			"departure": "12:30PM"
		},
		{
			"flight": "Delta 3833",
			"departure": "3:00PM"
		}
	]
}
```


To search a flight you simply enter the time of the departure. Your flight search algorithm will only display flights at a distance (plus or minus) of 5 hours from the time you select for your departure. For example if you search for a 6AM flight, you will see both the 7:30AM and the 10:30AM flights in the results.

Please implement a REST service to display the flights for different queries with what you consider would be the best implementation for this scenario.

Requirement #2: Credit Card Validator

Develop an application that will validate credit cards based on the following requirements:

- Only Visa and MasterCard are accepted.
- All card numbers have 16 digits.
- Card numbers can optionally include spaces every 4 digits.
- Visa card numbers start with a 4.
- MasterCard card numbers start with the numbers 51 through 55.
- The final digit of the card (verification digit) must be validated against the Luhn formula.
- Expiration dates must use the format MM/YY (e.g. 01/19 represents January 2019).
- Expiration dates must be in the future.
- Credit cards must not be present on the blacklist (see below).

Here is a blacklist of cards that have been detected as high risk by our fraud prevention team. These cards must not pass validation:

```
{
   "blacklist":[
      "4788 3845 3855 2446",
      "5144 3854 3852 3845"
   ]
}
```

Please implement a REST service that will take as input credit cards numbers and their expiration date and provide a response if the validation passes or not. Also if the credit card has been blacklisted, the service must respond with an appropriate response. The details of the implementation is up to you, considering what would be the best way to solve this problem.