

---

# Coursework Specification

Software Testing (INFR10057) 2018/19 Spring

## NEW SPECIFICATIONS FOR TASK 3 - TEST DRIVEN DEVELOPMENT

Options may also take a list of integers as its value. In order to retrieve value types that are an integer list, please implement the following function as a method of the Parser class:

```
List<Integer> getIntegerList(String option)
```

Example:

```
Parser parser = new Parser();
parser.add("list", "l", Parser.STRING);
parser.parse("--list=1,2,4-7,10");
List l = parser.getIntegerList("list");
```

In this example, the returned list contains list values {1, 2, 4, 5, 6, 7, 10}.

Parameter list:

**String option:** the full name or shortcut of an option.

Return value:

This method returns the list of all integer values retrieved from the value of the option in an ascending order. It returns an empty list if the input is not valid or the parser fails.

Specifications:

1. The order of search is full name of options first and then shortcut. For example, if “o” exists as a full name for an option and a shortcut for another option, this function returns the value of the first option.
2. If the option is not provided a value, an empty list is returned.
3. Non-number characters except hyphen (-) can be used as separators including commas, dots, spaces etc. For example, the following values can produce identical lists:

**1,2 4**

**1,2.4**

**[]1<2>4{}**

4. A hyphens (-) indicates an inclusive range of numbers. For example, **4-7** and **7-4** both represents integers 4, 5, 6, 7.

- 
5. The unary prefix hyphen indicates a negative value. For example, **-7** is a negative value, **-7-5** includes integers -7, -6, -5 and **-2-1** includes integers -2, -1, 0, 1.
  6. Hyphens cannot be used as a suffix. **3-**, for instance, is invalid and an empty list should be returned.

Partial marks are given to implementations that do not support all above specifications.