

- Add comment block at top:

```
/*  
 * Java program name  
 *  
 * Project description  
 *  
 * Your name  
 * The version date  
 * The course number and section  
 */
```

1. **Class name of MoonSamples**

2. **Import java Scanner**

3. **Create new method:** `public static String[] getElements(String InputElementString) {}`

- a. Separate a string into a string array using `<String>.split(",")`

4. **Create new method:** `public static double[][] getSamples(String inputSamplesString) {}`

- a. Split into separate arrays based on:

`<String>.split("<>")`

- i. Within that split separate the elements,

`<String>.split(",")`

5. `public static int[] searchForLife(double [][] samples){}`
 - a. Take in sample values data, plug and chug into following formula:

formula for life = (8·carbon dioxide) + (2·magnesium) + sodium + (4·potassium) + chloride + (5·water)

- b. If (formula for life >= 300) {
sample number is returned in an int array.
If (first row of data attains the score of 300 or more) {
the number 1 will be inserted into the first position in the int array.
}
}

Since not all samples may meet the criteria, the array returned should be resized so that it only contains the values that meet the criteria with no extra data.

6. `public static String searchHighestElements(double [][] samples, String[] elements, int sampleNum){}`
 - a. The fourth method is `searchHighestElements()`. This method will take in a String array of elements, a 2D double array of sample values, and a sample to search. It will search that sample and will return a string containing the two elements that are found in the highest amount for that sample in the order of highest first, then second highest element.

7. `public static int searchHighestSample(double [][] samples, String[] elements, String element){}`
 - a. Use a loop to find the single element's location in the elements string. That tells you the row of the samples array.
 - b. Use a loop to search for highest value (from project 4)
 - c. Return that value

9. In main method, prompt user to input two strings:

- a. The elements
 - i. Sample:
carbon-dioxide,magnesium,sodium,potassium,chloride,water
- b. The samples
 - i. 8.3,4.5,6.7,2.3,12.5,4.5<>3.9,1.8,34.7,23.5,1.2,14.3<>6.7,7.4,1.5,1
8.4,7.2,23.7<>23.4,5.6,2.9,18.5,39.5,18.2<>15.4,5.3,27.4,9.8,3.8,2
7.4
- c. Pass first string into `getElements()`.

- d. Pass second string into getSamples().
 - e. Print searchforLife(), argument is getSamples().
 - i. **"The samples that contain life are: "**
 - f. Print searchHighestElements(), passing in getSamples() and getElements()
 - i. **The highest elements for sample 1 are chloride and carbon-dioxide**
 - g. Print searchHighestSample(), passing in the results of getSamples() and getElements() and print the result
 - i. **The sample with the highest value of the element water is 5**
- Add algorithm comments throughout
 - Submit to Gradescope, 4 attempts, use Junit tests first
 - Submit two screenshots one of the code running and one of the Junit tests passing

Sample run:

```
The samples that contain life are: 4 5
The highest elements for sample 1 are chloride and
carbon-dioxide
The sample with the highest value of the element
water is 5
```

Rubric for Programming Project 5

Criteria	Points
Algorithm submitted in class on time	15
Comments used appropriately (including comment header information and fully documented algorithm)	15
Appropriate choice of variable names	5
Correctly populates elements array	10
Correctly populates samples array	10
Separate methods are used as specified and using good program design	20
Output is correct	10
Program layout and appearance (Coding style is clear and easily understood)	5
Screenshot of JUnit tests passing & MoonSamples.java running	10
TOTAL	100