* **JS Advanced Exam – Sample Problem**
* **Problem 2. Garden**

class Garden {

//TODO Implement this class

}

Write a class **Garden**, which implements the following functionality:

**Functionality**

**Constructor**

Should have these **3** properties:

* **spaceAvailable –** Number
* **plants –** Array (empty)
* **storage –** Array (empty)

**At the initialization of the Garden class,** the **constructor** accepts the **spaceAvailable.**

**addPlant (plantName, spaceRequired)**

The **plantName** is of type **string**, while the **spaceRequired** is of type **number**.

* If there is **not enough space in the garden** for the new plant, **throw an Error**:

"**Not enough space in the garden.**"

* Otherwise, this function should add the plant with the properties: **plantName**, **spaceRequired**, **ripe**: **default false, quantity: default 0** to the plants array, reduce the space available with the space required by the plant, and **return**:

"**The {plantName} has been successfully planted in the garden.**"

**NOTE: Plant names will be unique.**

**ripenPlant(plantName, quantity)**

The **quantity** is of type **number**.

* If the plant is not found, **throw an Error**:

"**There is no {plantName} in the garden.**"

* If the plant is already ripe, **throw an Error**:

"**The {plantName} is already ripe.**"

* If the received quantity is less than or equal to 0, **throw an Error**:

"**The quantity cannot be zero or negative.**"

* Otherwise, this function should set the ripe property of the particular plant to true
* and add the quantity to the quantity property of the plant.
* If the quantity passed as a parameter is 1, **return:**

"**{quantity} {plantName} has successfully ripened.**"

If the quantity parameter is greater than 1, **return:**

"**{quantity} {plantName}s have successfully ripened.**"

**harvestPlant(plantName)**

* If the plant is **not found**, throw **an Error:**

"**There is no {plantName} in the garden.**"

* If the plant **is not ripe**, throw **an Error**:

"**The {plantName} cannot be harvested before it is ripe.**"

* Otherwise, this function should **remove** the plant from the plants array,
* add it to storage with properties **plantName** and **quantity,**
* free up the total space that the plant required, and **return**:

"**The {plantName} has been successfully harvested.**"

**generateReport()**

This method should **return** the complete information about the garden:

* On the first line:

"**The garden has { spaceAvailable } free space left.**"

* On the second line list all plants that are in the garden **ordered alphabetically by plant name ascending** in the format:

"**Plants in the garden: {plant1Name}, {plant2Name}, {…}**"

* On the third line add:
* If there are no plants in the storage, print:

"**Plants in storage: The storage is empty.**"

If there are plants in the storage list them in the format:

"**Plants in storage: {plant1Name} ({plant1Quantity}), {plant2Name}, ({plant2Quantity}), {…}**"

* **Examples**

|  |
| --- |
| **Input 1** |
| **const *myGarden* = new** Garden**(250)**  ***console*.log(*myGarden*.addPlant('apple', 20));**  ***console*.log(*myGarden*.addPlant('orange', 200));**  ***console*.log(*myGarden*.addPlant('olive', 50));** |

|  |
| --- |
| **Output 1** |
| The apple has been successfully planted in the garden.  The orange has been successfully planted in the garden.  Uncaught Error Error: Not enough space in the garden. |

|  |
| --- |
| **Input 2** |
| **const *myGarden* = new** Garden**(250)**  ***console*.log(*myGarden*.addPlant('apple', 20));**  ***console*.log(*myGarden*.addPlant('orange', 100));**  ***console*.log(*myGarden*.addPlant('cucumber', 30));**  ***console*.log(*myGarden*.ripenPlant('apple', 10));**  ***console*.log(*myGarden*.ripenPlant('orange', 1));**  ***console*.log(*myGarden*.ripenPlant('orange', 4));** |

|  |
| --- |
| **Output 2** |
| The apple has been successfully planted in the garden.  The orange has been successfully planted in the garden.  The cucumber has been successfully planted in the garden.  10 apples have successfully ripened.  1 orange has successfully ripened.  Uncaught Error Error: The orange is already ripe. |

|  |
| --- |
| **const *myGarden* = new** Garden**(250)**  ***console*.log(*myGarden*.addPlant('apple', 20));**  ***console*.log(*myGarden*.addPlant('orange', 100));**  ***console*.log(*myGarden*.addPlant('cucumber', 30));**  ***console*.log(*myGarden*.ripenPlant('apple', 10));**  ***console*.log(*myGarden*.ripenPlant('orange', 1));**  ***console*.log(*myGarden*.ripenPlant('olive', 30));** |

|  |
| --- |
| **Output 3** |
| The apple has been successfully planted in the garden.  The orange has been successfully planted in the garden.  The cucumber has been successfully planted in the garden.  10 apples have successfully ripened.  1 orange has successfully ripened.  Uncaught Error Error: There is no olive in the garden. |

|  |
| --- |
| **Input 4** |
| **const *myGarden* = new** Garden**(250)**  ***console*.log(*myGarden*.addPlant('apple', 20));**  ***console*.log(*myGarden*.addPlant('orange', 100));**  ***console*.log(*myGarden*.addPlant('cucumber', 30));**  ***console*.log(*myGarden*.ripenPlant('apple', 10));**  ***console*.log(*myGarden*.ripenPlant('orange', 1));**  ***console*.log(*myGarden*.ripenPlant('cucumber', -5));** |

|  |
| --- |
| **Output 4** |
| The apple has been successfully planted in the garden.  The orange has been successfully planted in the garden.  The cucumber has been successfully planted in the garden.  10 apples have successfully ripened.  1 orange has successfully ripened.  Uncaught Error Error: The quantity cannot be zero or negative. |

|  |
| --- |
| **Input 5** |
| **const *myGarden* = new** Garden**(250)**  ***console*.log(*myGarden*.addPlant('apple', 20));**  ***console*.log(*myGarden*.addPlant('orange', 200));**  ***console*.log(*myGarden*.addPlant('raspberry', 10));**  ***console*.log(*myGarden*.ripenPlant('apple', 10));**  ***console*.log(*myGarden*.ripenPlant('orange', 1));**  ***console*.log(*myGarden*.harvestPlant('apple'));**  ***console*.log(*myGarden*.harvestPlant('olive'));** |

|  |
| --- |
| **Output 5** |
| The apple has been successfully planted in the garden.  The orange has been successfully planted in the garden.  The raspberry has been successfully planted in the garden.  10 apples have successfully ripened.  1 orange has successfully ripened.  The apple has been successfully harvested.  Uncaught Error Error: There is no olive in the garden. |
| **const *myGarden* = new** Garden**(250)**  ***console*.log(*myGarden*.addPlant('apple', 20));**  ***console*.log(*myGarden*.addPlant('orange', 200));**  ***console*.log(*myGarden*.addPlant('raspberry', 10));**  ***console*.log(*myGarden*.ripenPlant('apple', 10));**  ***console*.log(*myGarden*.ripenPlant('orange', 1));**  ***console*.log(*myGarden*.harvestPlant('apple'));**  ***console*.log(*myGarden*.harvestPlant('raspberry'));** |

|  |
| --- |
| **Output 6** |
| The apple has been successfully planted in the garden.  The orange has been successfully planted in the garden.  The raspberry has been successfully planted in the garden.  10 apples have successfully ripened.  1 orange has successfully ripened.  The apple has been successfully harvested.  Uncaught Error Error: The raspberry cannot be harvested before it is ripe. |

|  |
| --- |
| **Input 6** |
| **const *myGarden* = new** Garden**(250)**  ***console*.log(*myGarden*.addPlant('apple', 20));**  ***console*.log(*myGarden*.addPlant('orange', 200));**  ***console*.log(*myGarden*.addPlant('raspberry', 10));**  ***console*.log(*myGarden*.ripenPlant('apple', 10));**  ***console*.log(*myGarden*.ripenPlant('orange', 1));**  ***console*.log(*myGarden*.harvestPlant('orange'));**  ***console*.log(*myGarden*.generateReport());** |

|  |
| --- |
| **Output 6** |
| The apple has been successfully planted in the garden.  The orange has been successfully planted in the garden.  The raspberry has been successfully planted in the garden.  10 apples have successfully ripened.  1 orange has successfully ripened.  The orange has been successfully harvested.  The garden has 220 free space left.  Plants in the garden: apple, raspberry  Plants in storage: orange (1) |