# Testing Mini Project

**Task**: Implement the following in TypeScript, focusing on a plant growth system. Create one interface – Plant, and two classes – a parent GardenPlant and a child Flower. Also, create two functions that interact with these classes.

**Plant Interface**

In the file Plant.ts, create a Plant interface and export it.

* Properties:
  + growthRate (a number)
  + currentHeight (a number)
* Methods:
  + water(): Has no parameters. It returns nothing.
* Jest Tests: N/A

**GardenPlant Class**

In the file GardenPlant.ts, create a GardenPlant class and export it.

* Implements the Plant interface.
* Properties:
  + growthRate (a number)
  + currentHeight (a number) - Always starts at 0
* The constructor parameters:
  + growthRate (a number) - sets the growth rate property
* Methods:
  + water(): Takes no parameters. Increases the currentHeight by adding the growthRate.
  + prune(): Takes no parameters. Reduces the currentHeight by 0.5. Can't go below 1.0
* Jest Tests: In GardenPlant.test.ts, write the following test cases:
  + Confirm that a new instance of GardenPlant has growthRate set from the constructor and that currentHeight is set to 0.
  + Calling water() increases the currentHeight property by adding the growthRate.
  + Calling prune() on a GardenPlant with currentHeight at 10 reduces the currentHeight to 9.5 (HINT: Remember to use [toBeCloseTo](https://docs.google.com/presentation/d/e/2PACX-1vSTGsJ6PWmWYCYgBR8PPiYL1wFMB9FV2mAqOR14s3a2Kjnt2daXX07S0IdLB9R604z1u8h30v-FoikX/pub?start=false&loop=false&delayms=3000&slide=id.ge51e52e251_0_0) for floating point tests.)
  + Calling prune() on a GardenPlant with currentHeight set to 1.0 doesn't reduce the plant's currentHeight any further.

**Flower Class**

In the file Flower.ts, create a Flower class and export it.

* Flower is a subclass of GardenPlant.
* Additional properties:
  + hasThorns (a boolean)
* The constructor parameters:
  + growthRate (a number) - sets the growthRate property (Hint: use super constructor.)
  + hasThorns (a boolean) - sets the hasThorns property. Defaults to false.
* Methods:
  + This class overrides the prune() method by reducing the currentHeight 2 times the normal amount if hasThorns is true. If hasThorns is false it resorts to the normal prune method.
* Jest Tests: In Flower.test.ts, write the following test cases:
  + Confirm that a new instance of Flower has growthRate and hasThorns set from the constructor parameters and currentHeight is set to 0.
  + Confirm that hasThorns is set to false if not provided.
  + Calling water() increases the currentHeight property by adding the growthRate.
  + Calling prune() on a Flower with currentHeight set to 10 and hasThorns set to false reduces the currentHeight to 9.5
  + Calling prune() on a Flower with currentHeight set to 10 and hasThorns set to true reduces the currentHeight to 9.0
  + Calling prune() on a Flower with currentHeight set to 1 doesn't reduce the plant's currentHeight any further.

**assessGrowth Function**

In the file functions.ts, create an assessGrowth function and export it.

* Parameters
  + plants (an array of Plant)
* Returns: a number
* Functionality: return the sum of the currentHeight of all plants in the array.
* Jest Tests: In functions.test.ts, write the following test case:
  + Create an array of with a mix of GardenPlant and Flower instances and call the assessGrowth function with your array. The function should accurately calculate the sum of all the plant's currentHeights.

**waterPlants Function**

In the file functions.ts, create a waterPlants function and export it.

* Parameters:
  + plants (an array of Plant)
* Returns: nothing
* Functionality: loop through the array of plants and call the water method on each of them.
* Jest Tests: In functions.test.ts, write the following test case:
  + Create an array of Plant instances and call the waterPlants function with your array. The function should water all the plants, which affects their currentHeights.