Garden Project (Part I)

Directions: You will be completing this program on your own – using your notes, prior projects, and online resources. Limit the number of times you ask the teacher for help – try to debug your program on your own. As a suggestion, make sure that you complete each step fully (meaning the program compiles correctly) before moving on to the next step. This means that you complete step one before you move onto step two and you complete step two before your move onto step three, etc.
Mrs. Moffat will not help you with your program if you complete these steps out of sequence.

> **Hint: This program should `compile` after _every_ step.**
> **If it does not `compile`, then you made a mistake and**
> **you should most definitely fix it before moving on. Good luck!** ☺

Part A: Download

1. Download and unzip the compressed "`GardenStarter`" **File.**

Part B: Flower Class

1. **Field Variables:**
   a. _**SHARED**_ `final` **literal** `String` **array:** `names`
      i. **This array stores the names of possible** `Flowers`
      ii. **Store:** "`Daisy`", "`Orchid`", "`SunFlower`", "`Tulip`"
      iii. **Note: Be careful of the spelling!**

   b. _**SHARED**_ `final` **literal** `int` **array:** `fadingAge`
      i. **This array stores the age when a** `Flower` **starts to fade.**
      ii. **Store:** `500, 100, 2000, 600`
      iii. **This array is associated with the** `names[]` **in that a** `Daisy` **takes 500 before it starts to fade, the** `Orchid` **takes 100 before it starts to fade, etc.**

   c. _**SHARED**_ `final` **literal** `double` **array:** `spreadChance`
      i. **This array stores the decimal percentage of a** `Flower` **spreading its seeds in the** `Garden`.
      ii. **Store:** `0.002, 0.0015, 0.001, 0.0022`
      iii. **This array is associated with the** `names[]` **in that a** `Daisy` **has a 0.2% chance of spreading, an** `Orchid` **has a 0.15% chance of spreading, etc.**

   d. _**SHARED**_ `final int:` `spreadDistance`
      i. **Set equal to** `101`;

   e. `int:` `age, transparency, type`

2. **Constructor:**
   a. **Build a** `Flower` **with a** `int type` **parameter.**
      i. **Set** `type` **equal to the field variable**
      ii. **Set** `age` **equal to 0.**
      iii. **Set** `transparency` **equal to 255;**
      iv. **Set image to** `String Array.png`
         1. **Example:** `names[type] + ".png"`

3. Methods:
   a. `age()`
      i. Add one to the `age` variable.
      ii. Check to see if the `age` is greater than the `fadingAge` array of the flower `type`. If so…
         1. Get the image and set the **transparency method** to the `transparency` field variable.
            a. HINT: You may have to do a little research in order to find this method's name.
         2. Subtract one from the `transparency` field variable.
      iii. Check to see if the `transparency` is no longer positive. If so, remove object.

   b. `spread()`
      i. Check to see if the `age` is greater than the `fadingAge[type]/2` and if `Math.random()` is less than the `spreadChance[type]`. If so…
         1. Build a new `Flower` of the same type.
         2. Add the object at a semi-random x-coordiante and a semi-random y-coordinate. Use the code below:

```
int x = getX() + (int)( Math.random() * spreadDistance ) - spreadDistance / 2;
int y = getY() + (int)( Math.random() * spreadDistance ) - spreadDistance / 2;
```

   c. `act()`
      i. This method should call the following methods:
         1. `spread()`
         2. `age()`

Part C: World Class

1. Methods:
   a. `getRandomX()`
      i. No parameters
      ii. Returns a random x-coordiante from 0 to the width
      iii. Methods can be used any time you need them.

   b. `getRandomY()`
      i. No parameters
      ii. Returns a random y-coordinate from 0 to the height
      iii. Methods can be used any time you need them.

   c. `buildWildFlowers()`
      i. Parameter indicating the `amount` of `Flowers` to build.
      ii. Create a loop to build that `amount` of `Flowers`.
         1. HINT: When you build a `Flower`, you will have to send over a parameter to determine which type of `Flower` you will be building. Each `Flower` should be different or *random*.
         2. REMEMBER: There are 4 different types of `Flowers` you can build.
      iii. Add the `Flowers` to the screen using a random x-coordiante and random y-coordiante.
         1. NOTE: Don't reuse Math.random(), use your method.

2. Constructor:
   a. In the given constructor, call the `buildWildFlowers()` and initialize it to 20.