A House having a single Picture and an array of Pictures

Important: The Mad Ph.D. video https://www.youtube.com/watch?v=1JJL0YszYik explains:

1. A class definition like
   ```java
   class House {
   }
   ```
   is a **blueprint** for making objects; ```java
class House {
   ...}
   ```
is NOT an object. (Such a class definition should be put in a file named House.java)

2. The `new House();` operation makes a different House instance each time the computer performs it. (The specific technical word for "make" in this context is **instantiate**.) (Commanding the `new` operation should be done in application code, often in the body of `main()`)

3. Reference variables declared (usually in the body of `main( )`) like ```java
   House PhDsHouse;
   House MiomotosHouse;
   ```
   and storing reference values (this is, addresses) like reference variables in them. Examples: ```java
   PhDsHouse = new House();
   MiomotosHouse = new House();
   ```
   These variables locate instances (that is, objects) for future computation on them.

We will add:

1. Enhance the blueprint to specify that each House shall have a String reference in it.
   ```java
   public class House
   {
      String owner; // Purpose: Each House must track its owner's name.
   }
   ```

2. Whenever a House is instantiated, immediately thereafter certain code is run:
   ```java
   public House(String param) {
      this.owner = param;
   }
   ```
   The `House(String param)` method we added is, technically, a **constructor**.

3. A method to return a String whenever `println` is used to print a House:
   ```java
   public String toString( ) {
      return ("House owned by " + this.owner);
   }
   ```

4. A main method to demonstrate and do a little testing of our House programming:
   ```java
   public static void main(String[] a)
   {
      House PhDsHouse = new House("Mad Ph.D.");
      House MiomotosHouse = new House("Miomoto");
      System.out.println( PhDsHouse );
      System.out.println( MiomotosHouse);
   }
   ```

Assignment for this Lab: Implement the above PLUS the idea that Houses can have Pictures in them! Your programming will support a story that the House owners are art collectors.

TAs and classmates should discuss and tell each other every detail they need to carry out the steps (on page 2) in the lab, with the objective to be able to code things like this in the future (like the final project); not just copying anything by rote just to get it done.

Before or during the programming, it would be wise to copy from `/usr/local/depts/cs/geintro/mediasources` (or from other places in your computer, or from the Web) 4 different small .jpg digital images so you can demonstrate your work.

For attendance credit, the TA will verify that his/her students have uploaded to Blackboard their work, or otherwise achieved the objective. All steps above will be demonstrated in Tues. lecture, so if you're a bit lost, finish the lab after that lecture.
1. AFTER getting your programming of Houses above to work, enhance the blueprint so each future House shall have
   - one favorite Picture reference variable, plus
   - one variable for an array of Picture references.
   Make up the two names for those two variables, so each name expresses what the variable is for, in other words, the purpose of the variable! (space for notes on how)

2. You will add code to the House constructor to instantiate that array so 3 Pictures can eventually be referenced. The code to add is below. Find out where to add it. Remember the idea for future use! this. the name you chose for the array = new Picture[3];

3. You will add to the House class a method to help an application add the favorite Picture to a House. (You must make up that method's name, give it a parameter variable, and code the body to store the parameter variable's value into the favorite Picture variable you named in step 1. (space for notes on how)

4. You will add to the House class a method that shows the favorite Picture.

5. You will add to the application: code to test and demonstrate your work on steps 1, 3 and 4 (Ignore the array for now, use and test it later.)

6. You will add to the House class a method for adding 3 additional Pictures, to be referenced by the 3 array elements. (YOU decide how to do this; discuss in lab. You can make it take 3 Pictures at a time, or one Picture at a time. Just make your testing and demonstration work.)

7. You will add to the House class a method for showing all 3 additional Pictures.

8. And of course, add to the application: code to test and demonstrate everything you programmed that you did not demonstrate and test before.