```java
public class Assistant {

    private String email;

    private String name;

    private static Assistant[] assistants = new Assistant[100];

    private static int numAssistants = 0;

        /**
         * The constructor for the Assistant class.
         * This is a class used for people related to the university (staff or students)
         * who are volunteering to perform COVID tests.
         *
         * @param email the university email of the assistant.
         * @param name the name of the assistant.
         */
        public Assistant(String email, String name){
            this.email = checkEmail(email);
            this.name = checkName(name);
            addAssistants(this);
            iterateNumAssistants();
        }

        /**
         * The email getter method.
         *
         * @return the university email of a specific assistant.
         */
        public String getEmail(){
            return email;
        }

        /**
         * The name getter method
         *
         * @return the name of a specific assistant.
         */
        public String getName(){
            return name;
        }

        /**
         * The array of the all assistants static getter method.
         *
         * @return the array of all assistants.
         */
        public static Assistant[] getAssistants(){
            return assistants;
        }

        /**
         * The number of assistants static getter method.
         *
         * @return the number of assistants.
         */
        public static int getnumAssistants(){
            return numAssistants;
        }

        /**
         * The email setter method.
         *
         * @param email the university email of a specific assistant.
         */
        public void setEmail(String email){
            this.email = checkEmail(email);
        }

        /**
         * The name setter method.
         *
         * @param name the name of a specific assistant.
         */
        public void setName(String name){
            this.name = checkName(name);
        }

        /**
         * The number of assistants iterator private static method.
         * Increases the number of assistants by 1 when called.
         */
        private static void iterateNumAssistants(){
            numAssistants += 1;
        }

        /**
         * The add assistants to the array of assistants static method
         *
         * @param assistant an instance of an assistant.
         */
        private static void addAssistants(Assistant assistant){
            assistants[numAssistants] = assistant;
        }

        /**
         * The name checker private method.
```

```java
     * Checks if the name of an assistant is not empty or blank spaces.
     *
     * @param name the name of a specific assistant.
     * @return the name of a specific assistant.
     */
    private String checkName(String name){
        if (name.trim().isEmpty()){
            throw new IllegalArgumentException("The name string should have at least one character that is not a space.");
        }
        return name;
    }

    /**
     * The email checker private method.
     * Checks if the email of an assistant ends with "@uok.ac.uk" and is unique.
     *
     * @param email the email of a specific assistant.
     * @return the email of a specific assistant.
     */
    private String checkEmail(String email){
        if (email.endsWith("@uok.ac.uk")){
            for (int i=0;i<numAssistants;i++){
                if (assistants[i].getEmail().equals(email)){
                    throw new IllegalArgumentException("The email should be unique.");
                }
            }
            return email;
        }
        throw new IllegalArgumentException("The email string should always end with @uok.ac.uk.");
    }

    /**
     * A method to return all the assistants as a string.
     *
     * @return a string of all the assistants.
     */
    public static String toStringAll(){
        String allAssistants = "Assistants-\n";
        for (int i=0;i<numAssistants;i++){
            allAssistants = allAssistants.concat((i+11)+". "+assistants[i].toString()+"\n");
        }
        return allAssistants;
    }

```

```
        /**
         * A method to return the string of an assistant
         *
         * @return a string of an assistant.
         */
        public String toString(){
            return "| "+name+" | "+email+" |";
        }
    }


```