

CS 1110 Assignment 3

1. Song Class

Name: `Song.java`

Create a Song class with the following items:

Attributes (private):

- Title
- Artist
- Duration

Constructors:

- No argument constructor (set the attributes to default values)
- Constructor that allows the attributes to be set

Operations (public):

- `toString()` to return a string representation of the Song in the format "Title" by Artist (duration) (e.g., "Fireflies" by Owl City (3:54))
- Getters for each attribute
- Setters for each attribute

Add a `main()` method to the Song class that creates two Song objects, one using the no arg constructor and one using the other constructor. Use the setters to set values for one of the Song objects. Display the result of both Song object's `toString()` methods.

Example Output:

```
song1: "Fireflies" by Owl City (3:54)
song2: "Sukiyaki" by Kyu Sakamoto (3:09)
```

2. Album Class

Name: Album.java

Create an Album class with the following items:

Attributes (private):

- Title
- Artist
- Year
- Tracks (an array of Song objects)

Constructors:

- No argument constructor (set the attributes to default values)
- Constructor that allows the attributes to be set

Operations (public):

- toString() to return a string representation of the Album in the format:

```
Album Title
Artist
Year
```

```
1. First Track Title (duration)
2. Second Track Title (duration)
3. Third TrackTitle (duration)
...
```

- Getters for each attribute
- Setters for each attribute

Add a main() method to the Album class that creates an Album object. Display the result of the Album object's toString() method.

Example Output (Abbreviated):

```
album1:

Discovery
Daft Punk
2001

1. One More Time (5:20)
2. Aerodynamic (3:32)

...
```

3. Email Class

Name: Email.java

Create a Email class with the following items:

Attributes (private):

- Local Part (part to the left of the @)
- Domain (part to the right of the @)

Constructors:

- No argument constructor (set the attributes to default values)
- Constructor that allows the attributes to be set

Operations (public):

- toString() to return a string representation of the Email in the format localPart@domain (e.g., johndoe@website.com)
- toAntiSpamString() to return a string representation of the Email in an anti-spam format (e.g., johndoe at website dot com)
- Getters for each attribute
- Setters for each attribute

Add a main() method to the Email class that creates two Email objects, one using the no arg constructor and one using the other constructor. Display the result of both Email object's toString() methods and toAntiSpaimString() methods.

Example Output:

```
email1:
johndoe@website.com
johndoe at website dot com

email2:
abc12340@ucmo.edu
abc12340 at ucmo dot edu
```

4. Student Class

Name: `Student.java`

Create a Student class with the following items:

Attributes (private):

- First Name
- Last Name
- ID (700 Number, nine digits including the 700 at the beginning)
- Email (Use the Email class. The format of an email address is first letter of first name, an x, first letter of last name, last four digits of ID, and 0, domain is ucmo.edu)
- Grades (array of char)

Constructors:

- Constructor that allows the names and grades to be set, generate the ID and then the email address using the name and ID

Operations (public):

- `toString()` to return a string representation of the Student object in the format:

```
Doe, John (700123456)
Email: jxd34560@ucmo.edu
GPA: 3.5
```

- `getGPA()`
- Getters for each attribute
- Setters for names and grades

Add a `main()` method to the Student class that creates a Student object. Display the result of the Student object's `toString()` method.

Example Output:

```
student1:

Doe, John (700123456)
Email: jxd34560@ucmo.edu
GPA: 3.5
```