## Spoopify

You are tasked to create **three classes**: **a Song** class, **an Album** class, and **a Band** class.



The **Song** class should receive a **name** (string), **length** (float) and **single** (bool) upon initialization. It has **one** method:

* **get\_info()**
  + Returns the information of the song in this format: **"{song\_name} - {song\_length}"**

The **Album** class should receive a **name** (string) and **songs(one, many or none)** **as arguments** upon initialization. It also has an **instance** **attribute published** (**False** by default). It has **four** methods:

* **add\_song(song: Song)**
  + Adds the **song to the album**. Return **"Song {song\_name} has been added to the album {name}."**
  + If the song is **single**, return **"Cannot add {song\_name}. It's a single"**
  + If the album is **published**, return **"Cannot add songs. Album is published."**
  + If the song is **already added**, return **"Song is already in the album."**
* **remove\_song(song\_name: str)**
  + Removes the song with the given name and return **"Removed song {song\_name} from album {album\_name}."**
  + If the song is not in the album, return **"Song is not in the album."**
  + If the album is published, return **"Cannot remove songs. Album is published."**
* **publish()**
  + Publish the album (set to **True**) and return **"Album {name} has been published."**
  + If the album is published, return **"Album {name} is already published."**
* **details()**
  + Returns the information of the album, with the songs in it, in this format:

**"Album {name}**

**== {first\_song\_info}**

**== {second\_song\_info}**

**…**

**== {n\_song\_info}"**

The **Band** class should receive a **name** (string) upon initialization. It also has an **attribute albums** (**empty list**).

The class has **three** methods:

* **add\_album(album: Album)**
  + Adds an **album to the collection** and returns **"Band {band\_name} has added their newest album {album\_name}."**
  + If the album **is already added**, return **"Band {band\_name} already has {album\_name} in their library."**
* **remove\_album(album\_name: str)**
  + Removes the album from the collection and returns **"Album {name} has been removed."**
  + If the album is **published**, return **"Album has been published. It cannot be removed."**
  + If the album is **not in the collection**, return **"Album {name} is not found."**
* **details()**
  + Returns the information of the band, with their albums, in this format:

**"Band {name}**

**{album details}**

**...**

**{album details}"**

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| song = Song("Running in the 90s", 3.45, False)  print(song.get\_info())  album = Album("Initial D", song)  second\_song = Song("Around the World", 2.34, False)  print(album.add\_song(second\_song))  print(album.details())  print(album.publish())  band = Band("Manuel")  print(band.add\_album(album))  print(band.remove\_album("Initial D"))  print(band.details()) | Running in the 90s - 3.45  Song Around the World has been added to the album Initial D.  Album Initial D  == Running in the 90s - 3.45  == Around the World - 2.34  Album Initial D has been published.  Band Manuel has added their newest album Initial D.  Album has been published. It cannot be removed.  Band Manuel  Album Initial D  == Running in the 90s - 3.45  == Around the World - 2.34 |