

Computer Science I

Droid Factory

CSCI-141

Homework 10

10/17/2020



1 Introduction

In this homework you will use a queue to implement an assembly line at a droid factory. Droids comprise four different parts: a pair of arms, a pair of legs, a body, and a head (the arms and legs each come in connected pairs that can be affixed to the body). An assembly line at the droid factory consists of a circular conveyor belt at the center of which stands a droid factory worker. As droid parts pass by on the conveyor, the worker uses them to assemble droids.

At the start of each day the droid parts are delivered to the droid factory; they are dumped in essentially random order onto the conveyor belt and slowly begin to travel in a circle around the droid factory worker. As each part arrives at the worker's station the worker removes it. If the droid that the worker is currently building is missing the part, the worker attaches it to the droid. If the part is not needed because the droid already has the same part installed, the worker places the part back on the conveyor belt so that it will travel around the circle once again. When all of the droids have been assembled, the worker heads home for the day for a few games of Sabacc.

2 Provided Data

Provided along with this homework document are the following files: `droid_parts_1.txt` (contains parts for one droid), `droid_parts_3.txt` (contains parts for 3 droids), and

`droid_parts_100.txt` (contains parts for 100 droids). Each file contains one droid part per line, and includes the precise number of droid parts needed to build a number of droids (there are no extra parts). These files will represent a "shipment" of droid parts to your factory. While you are encouraged to use these files when implementing your homework you should assume that the graders may use different files with parts to build a different number of droids (though none of the test files will include any extra parts).

3 Tasks

You will write a Python program that completes several small tasks.

3.1 Task 0: Downloads and Imports

Note that for this homework, the only data structure that you will use is the queue implementation provided to you with this week's lecture materials and linked from the course schedule. You should not implement your own queue, or use any Python list or queue. You will need to make sure that the following files are in the same directory/project as your Python program.

1. `node_types.py` - required by `cs_queue`.
2. `cs_queue.py` - provided queue implementation.

3.2 Task 1

Create a new Python program named `droid_factory.py` and define a dataclass that represents a droid. Remember that droids have four parts: a head, a body, a pair of arms, and a pair of legs; each part is either installed in the droid, or it is not. A droid is completed when all four parts have been installed. In addition, droids have a unique, 5-digit serial number that you will assign in sequence (e.g. 10000, 10001, 10002, ...).

3.3 Task 2

Write a function that, given a parts filename and a conveyor belt, unloads all of the droid parts from the file onto the belt. Remember that the file contains one droid part per line (arms, legs, body, or head). When the function returns, all of the parts should be loaded onto the conveyor belt. You will use a queue to represent the conveyor belt.

3.4 Task 3

Write a function that, given a serial number and a conveyor belt, builds a new droid. The function should do at least the following:

1. Create a new (empty) droid with the specified serial number.
2. Remove parts from the conveyor belt one at a time. If the droid is missing the part, it should be added to the droid. Otherwise, it should be placed back onto the belt so that it will (eventually) come back around the circle to be used in the same or a future droid. A message should be printed to indicate which action was taken. See the output example below.

3. When all four parts have been installed, the droid is complete. Print a message including the droid's serial number.

3.5 Task 4

Write another function that, given a conveyor belt, uses all of the parts that have been loaded onto the belt to build as many droids as possible. This function should call the function that you wrote in the previous task to build one droid at a time until all of the parts have been used.

3.6 Task 5

Write a main function that does at least the following:

1. Prompts the user for the name of the droid parts file.
2. Parses the file and loads all of the droid parts onto the conveyor belt.
3. Calls the function that you wrote in the previous task to build all of the droids.

You are not limited to writing only the functions that are described above. You are encouraged to write several small helper functions. You will also probably find it useful to test each function individually, either with small input files or hard-coded information, before testing the complete program.

4 General Requirements and Example Output

Your Python file must be named: `droid_factory.py`

The example output below illustrates an execution of this program. Your output does not need to exactly match this, but should provide at least the same information, capability, and readability.

```
Enter parts filename: droid_parts_3.txt
Starting a shift at the droid factory!
Building a new droid with serial number 10001
  attaching arms...
  attaching legs...
  attaching body...
  placing unneeded part back on belt: arms
  placing unneeded part back on belt: body
  attaching head...
Droid 10001 has been assembled!
Building a new droid with serial number 10002
  attaching head...
  attaching legs...
  placing unneeded part back on belt: legs
  attaching body...
  placing unneeded part back on belt: head
  attaching arms...
Droid 10002 has been assembled!
Building a new droid with serial number 10003
  attaching arms...
  attaching body...
  attaching legs...
  attaching head...
Droid 10003 has been assembled!
All of the droids have been assembled! Time to clock out and play Sabacc...
```

5 Submission Instructions

Zip your `droid_factory.py` file into a file named `hw10.zip` and submit it to the appropriate MyCourses assignment by the deadline specified.

6 Grading

- 10%: Task 1: dataclasses
- 10%: Task 2: file processing and data structure creation
- 40%: Task 3: building a droid using the conveyor
- 25%: Task 4: running the factory
- 15%: Task 5: main function
- Up to -10%: For poor style, documentation, or incorrect submission technique/format.