**Code documentation**

Required packages and software

TODO

General overview

TODO

Proposed solution for each task

2.1 Production Line

Task 1:   
We've designed the following classes for the task: Batch, Buffer, Task, Unit, and ProductionLine.

Batch:

* Attributes:
  + size: Represents the number of wafers in a batch.
  + id: Used for printing the process.
* Initialization: All attributes are set using arguments.

Buffer:

* Attributes:
  + capacity: Represents the maximum wafers it can hold.
  + content: List of batches currently in the buffer.
  + id: Used for printing the process.
* Functions:
  + add\_batch: Insert a batch into the buffer.
  + remove\_batch: Remove a batch from the buffer.
  + get\_total\_wafers: Calculate the total wafers in the buffer
* Initialization: id and capacity are set using arguments.

Task:

* Attributes:
  + inputbuffer: where the task receives batches.
  + outputbuffer: where the task sends processed batches.
  + time\_per\_wafer: time required to process a single wafer.
  + active\_batch: batch being processed by the task.
  + id: used for printing the process.
* Functions:
  + load: insert a batch into the task.
  + unload: remove a batch from the task.
  + check\_if\_output\_buffer\_space: verify if there's enough space in the output buffer to load a batch.
* Initialization: inputbuffer, outputbuffer and time\_per\_wafer are set using arguments.

Unit:

* Attributes:
  + tasks: a list of the task in prioritized order.
  + time\_unitil\_finished: is a timestamp into the future on how long until the unit is free and can take a new task.
* Functions:
  + load: This function will call the the load function of the next task if the current time is bigger than the time\_until\_finished attribute.
  + Unload: This function will unload the finished task in the unit.
  + choose\_next\_task: Finds the first next task with a batch in the inputbuffer in the list of tasks.
* Initialization: id and tasks are set using arguments.

ProductionLine:

* Attributes:
  + We have an attribute for each buffer, each task, and each unit.
* Functions:
  + The only function we have is the initialization function that initializes the production line.

Attributes:

tasks: list of tasks within the unit

timestamp: represents the task completion time

id: used for printing the process

Initialization: attributes are set using arguments

Functions:

load unit: call the loading function of the active task

unload unit: call the unloading function of the active task

select next task: choose the next task based on the input buffer's content (heuristic: first task with non-empty input buffer)

ProductionLine:

Initialization function: creates the entire production line as described in the assignment

Buffer capacities:

start buffer: infinity

end buffer: infinity

intermediate buffers: 120

Task initialization: tasks are initialized with the appropriate input/output buffers

Task prioritization: dynamically created based on a dictionary and an argument list for desired prioritization

Task 2:

2.2 Simulator

Task 3:

Task 4:

2.3 Optimization

Task 5

Task 6

Task 7