Graphical user interface, application, table

Description automatically generated

# SENG2200 Programming Languages & Paradigms Assignment 2 Report

## Introduction

The goal in this project report is to describe the relationship between each class and answering all the requirements for the assignment 3 specification.  
 This Assignment is aims to design a simulator Production Line use Discrete Event Simulation (DES). This java program is to simulate the production of “items” on a production line, the items processing time is vary due to random factors. The user will be asked to input an average processing time of an item in a stage and range of processing time in a stage as well as the capacity of a storage.

## UML class diagram



# Use of Inheritance and Polymorphism

This Java program’s “S0” class, “normalStage” class, “twoTimeStage”class and ”endStage” class are the childs class which inheritance and Polymorphism form “AbstractStages”. Cause “S0” stages is only to generate the items for the production line and its having value of M, N, “normalStage” is having value of M, N processing speed and “twoTimeStage” is having value of 2\*M and 2\*N, “endstage” class which is the final stage of the production line and having value of value of M, N.

# Alter the program

It is easy to increase or decrease the number of stages in this production line design. To increase the number stations for the production line, we just have to increase the number of the normalStage (which having value of M,N). If we wan to increase as S2a/b/c, we just need to increase the number of the “twoTimeStage” (which having value of 2\*M, 2\*N).

# Alter the program in more complicated than the “Straight line”

Because of the stages and inner storage that this java program has designed is only allow simulating a particular “item” type. For the following example given by the assignment:  
Diagram

Description automatically generated

To cater for a production line that will involves taking two different types of items and assembling them to make a new type of item which will not be suitable for this designed java program. If I had known that this might be a possibility, I would make “item” class become an generics class “item <T >” which T can be referring to different type of items and allow variety item type as input.

If we want to achieve such production line, The blocking and starving method will remain the same but we need to create two “S0” stage as R0 and S0 given above and that will be generating item<b> and item <a> using “item<T>” class. The “normalStage” class then have to be considerate two different type of items to be completed productions then store the created “item<c>” into its next inner storage queue. As we want to assemble two different types of items to make a new type of item, this java program might need some extra condition for the stages to follow such as: must wait till both item<a> and item<b> both finished its own production in order to pass to its next stage.   
Conclusion

The SENG2200 Programming Languages & Paradigms Assignment 3 was first written on 15/05/2021 and finished on 02/06/2021. The designed production line which will be recording the item’s history such as “T1” (time when item enter an stage or queue), “T2” (time when item leave an stage or queue). average Time Calculation on the item stayed in each inner storage queue, each stage’s work, starve time, block time as well as number of item passing the “twoTimeStage” such as S2a/b.