Objective:

This assignment has been designed for students to apply appropriate concurrent program methods in **implementing** a concurrent program from a program specification.

Learning Outcomes

ON COMPLETION OF THIS ASSIGNMENT, YOU SHOULD BE ABLE TO DEMONSTRATE THE FOLLOWING LEARNING OUTCOME(S):

No.	Learning Outcome	Assessment
1	Explain the fundamental concepts of concurrency and parallelism in the design of a concurrent system (C2, PLO1)	Exam
2	**Apply the concepts of concurrency and parallelism in the construction of a system using a suitable programming language. (C3, PLO2)**	**Individual Assignment (System)**
3	Explain the safety aspects of multi-threaded and parallel systems (A3, PLO6)	Individual Assignment (Report)

Programme Outcomes (PO):

PLO2 Cognitive Skills - This relates to thinking or intellectual capabilities and the ability to apply knowledge and skills. The capacity to develop levels of intellectual skills progressively begins from understanding, critical/creative thinking, assessment, and applying, analysing, problem solving as well as synthesizing to create new ideas, solutions, strategies, or new practices. Such intellectual skills enable the learner to search and comprehend new information from different fields of knowledge and practices.

Individual Assignment - System (25%):

Question No.	Topic	Question Vs Taxonomy						PLO
		Cognitive Level						
		1	2	3	4	5	6	
		SQ	SQ	SQ	SQ	SQ	SQ	
1	Appropriateness of coding techniques used to implement design with appropriate comment lines in source codes			**20%**				2
2	Appropriateness of the Java concurrent programming facilities used.			**20%**				2
3	Program runs appropriately with basic requirements			**20%**				2
4	Additional requirements met			**20%**				2
5	Explanations of concurrency concepts implemented with relevant code samples			**20%**				2
	Total			**100%**				

Submission Requirements: Part 2

Assignment Handout Date : 11th March 2022

Assignment Due Date : 3rd June 2022

Case Study

The Problem

This part of the assignment will require you to **implement the airport simulation**, as broken down in the sections below.

The management of an airport thinks that the way in which the airport is operated means that incoming flights must spend too much time waiting for landing clearance. To evaluate the situation a simulation of the airport has been commissioned. This simulation will simply run with text output describing the events as they occur in the airport and collect a minimal amount of statistical data.

Intention of assignment

Even if valuable to the owner, the simulation is *not* the main purpose of this assignment - indeed, if this was the case there are much better techniques for *simulating* than writing a concurrent program.

The requirement of this assignment is to **implement** a program in which synchronization and communication takes place between several concurrent processes. It is intended to force you to solve (and not simply avoid) a range of interesting synchronisation problems.

Asia Pacific Airport

You have been tasked to automate the task of Air Traffic Controller (ATC). ATCs have three main tasks as an aircraft approaches an airport, all of which must be carried out as quickly as possible:

**********Basic Requirements**********

- There is only 1 runway for all planes to land and depart.
- Ensure that the aircraft does not collide with another aircraft on the runway or gates
- Once an aircraft obtains permission to land, it should land on the runway, coast to the assigned gate, dock to the gate, allow passengers to disembark, refill supplies and fuel, receive new passengers, undock, coast to the assigned runway and take-off.
- **Each step should take some time.**
- A congested scenario should be simulated where planes are waiting to land while the 2 gates are occupied.
- As the airport is small, there is no waiting area on the ground for the planes to wait for a gate to become available.

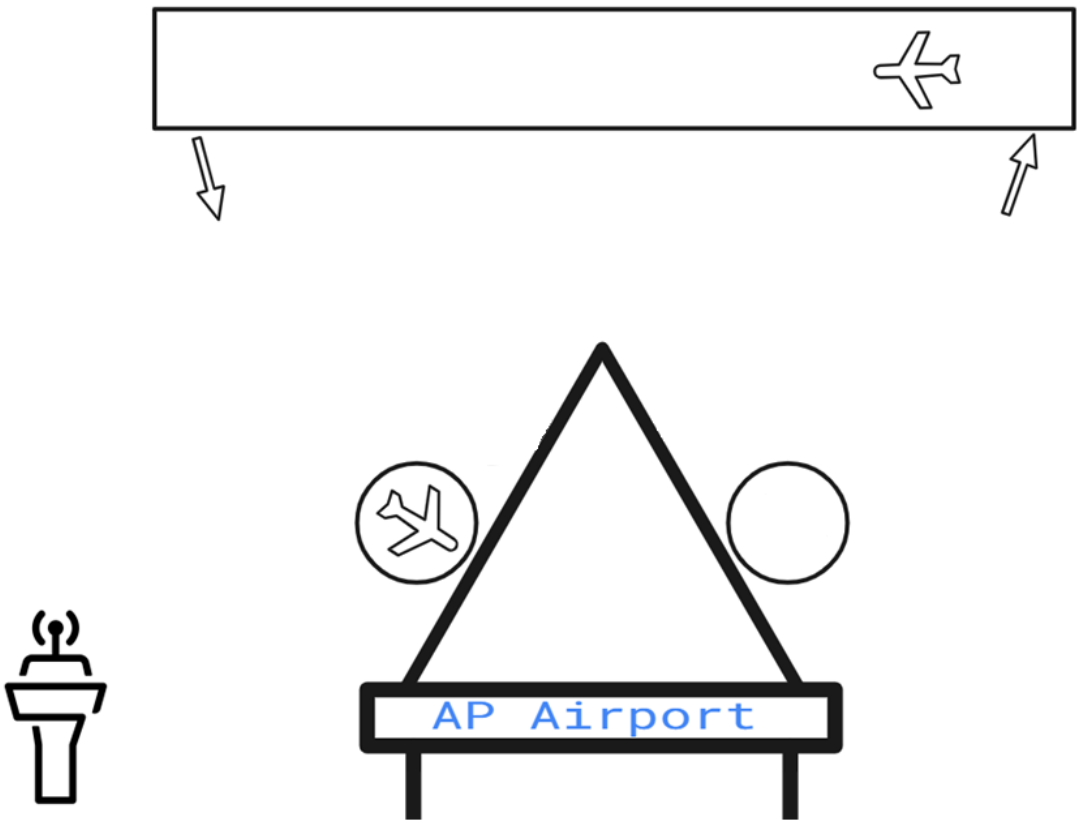

Figure 1: Layout of the Asia Pacific Airport (Not to scale)

**********Additional Requirements**********

These events should happen concurrently:

- Passengers disembarking/embarking

- Refill supplies and cleaning of aircraft

As there is only 1 refuelling truck, this event should happen exclusively:

- Refuelling of aircraft

State your assumptions and how you will implement them.

The Statistics

At the end of the simulation, i.e., when all planes have left the airport, the ATC manager should do some sanity checks of the airport and print out some statistics on the run. The result of the sanity checks must be printed. You must

- Check that all gates are indeed empty.
- Print out statistics on

  - Maximum/Average/Minimum waiting time for a plane.

  - Number of planes served/Passengers boarded.

Deliverables:

For this exercise, you are to model the ATC scenario and design a Java program to simulate activity for the airport:

- Altogether, 6 planes should try to land at the airport.
- Use a random number generator, so a new airplane arrives every 0, 1, 2, or 3 seconds. (This *might* be accomplished by an appropriate statement sleep (rand.nextInt(3000));
- Assume each plane can accommodate maximum 50 passengers.
- Assume passengers are always ready to embark and disembark the terminal (i.e., no capacity issues inside the passenger terminals)

Sample Output

In order to see what is happening dynamically you must have output from the passengers, the air traffic controller, and the pilots reporting all their major events.

Add information about which process/thread is doing the output. This way you can see if a process/thread acts for another, which is strictly forbidden, but is a common error for Java solutions (objects are not processes!). An example of such incorrect behaviour is

Thread-ATC : Plane 5: Requesting permission to land!
MainThread : ATC: Please wait and join the circle queue.
Thread-Passenger-8 : I'm boarding Plane 2 now.

Where you can see that not only the ATC thread is acting for Plane 5, but also the main thread is acting for the ATC.

You *must not*

- Kill a thread or process. You may not use any of the following primitives in Java:
  - Thread.stop
  - Thread.resume
  - Thread.suspend
  - Thread.interrupt

  You may not use the destroy or stop(0) primitives in - except to take care of temporary resources like simple timers.
- Solve the last orders problem in a manner forbidden in the description above.
- Resolve communication with an all-purpose one-channel solution.

Implementation

You should implement your simulation in Java.

Each simulation run should not take more than **60 seconds** to simulate.

Documentation for System (Week 12)

The documentation should detail the system implementation and testing.

1. Basic requirements met:
   - List of requirements met.
     - Code snippet of the Java concurrent programming facilities implemented.
     - Explanations of concurrency concepts (atomic statements, synchronization, etc) implemented if not stated in Part 1.

2. Additional requirements met:
   - List of requirements met.
     - Code snippet of the Java concurrent programming facilities implemented.
     - Explanations of concurrency concepts (atomic statements, synchronization, etc) implemented if not stated in Part 1.

3. Requirements which were **NOT** met:
   - List of basic requirements.
   - List of additional requirements

No more than 1000 words excluding references/appendix/coding.

Submission for System (Week 12)

- *Documentation for system*
  - *The documentation file should be named TP0XXXXX CCP Document.docx*
- *Java files required to run the simulation.*
  - *Zipped into a single zip file named TP0XXXXX CCP.zip*
- *Video of the simulation running. Maximum 5 minutes per person.*
  - *Simulate scenarios as stated above.*
  - *Show which requirements are met in the output and corresponding code.*
  - *The video file should be named TP0XXXXX CCP Video.zip*

Marking Scheme (NOT for student's documentation guidance)

Criteria	Total marks	Marks awarded
Appropriateness of coding techniques used to implement design with appropriate comment lines in source codes *	20	
Appropriateness of the Java concurrent programming facilities used.	20	
Program runs appropriately with basic requirements *	20	
Additional requirements met *	20	
Explanations of concurrency concepts implemented with relevant code samples *	20	
TOTAL MARKS	100	

*Based on video presentation of the simulation.