**Project 1B**

Simulation is the use of one system to imitate the behavior of another system. Simulations are often used when it is expensive or dangerous to experiment with the real system. There are physical simulations, such as wind tunnels used to experiment with designs for car bodies. Flight simulators are used to train airplane pilots. Mathematical simulations are systems of equations used describe some system, and computer simulations use the steps of a program to imitate the behavior of a system under study.

In a computer simulation, the objects being studied are usually represented as data, often as classes whose methods describe behavior of the objects and whose instance variables describe the properties of the objects. Most importantly, actions being studied are represented as methods of the classes, and the rules describing these actions are translated into computer algorithms. By changing the values of the data or by modifying these algorithms we can observe changes in computer simulation, then we can draw worthwhile conclusions concerning the behavior of the actual system.

This project is a simulation of an airport. As a specific example, we consider a small but busy airport with only one runway. In each unit of time, one plane can land or one plane can take off, but not both. Planes arrive ready to land or take off at random times, so at any given moment of time, the runway may be idle or a plane may be landing or taking off, and there may be several planes waiting either to land or take off.

In this simulation we need the following classes:

**Plane class:**

You need to create a class Plane whose objects represent individual planes.

The class needs to maintain data about particular *Plane* objects.

This data must include a flight number, time of arrival at the airport system, and a plane status as either arriving or departing.

Since we do not wish clients of this program to modify this data, we shall keep it in private data members.
What methods should the class have? Think about what a Plane can do. In this simulation a Plane only has a few things that it needs to do.

2) **class Runway** :

The aim of this simulation is to gather data about likely airport use. The class Runway is used to keep the statistics such as the number of planes processed, the average time spent waiting, the number of planes refused service etc.

The class Runway maintains two ArrayLists of type Planes.: A list called **landing** which keeps the info about landing planes, and a list called **takeoff** for planes that are ready to take off.

Some rules to be implemented by the class Runway.

a. It is better to keep a plane waiting on the ground than in the air. So in a small airport a plane can take off only if there are no planes waiting to land.

b. The Runway method activity, will service the first plane in the ArrayList of planes waiting to land, and only when this List is empty will it allow a **plane** take off.

Use a specialized type of class known as an enum to describe the three activities of the runway: idle, land, takeoff. (See pages 533-542 of the textbook for a description of how to set up an enum.)

The class Runway should have the following instance variables:

```
ArrayList<Planes> landing
ArrayList<Plane> takeoff
int listLimit   // to restrict the maximum # of planes in the landing/takeoff list
int numLandingRequests //number of planes asking to land
int numTakeoffrequests //number of planes asking to takeoff
int numLandings //number of planes that have landed
int numTakeoffs //number of planes that have taken off
```
int numLandAccepted //number of planes lined up to land
int numTakeoffAccepted //number of planes lined up to takeoff
int numLandRefused  //number of landing planes refused
int numTakeoffRefused //number of departing planes refused
int landWait // total time of planes waiting to land
int takeoffWait //total time of planes waiting to take off
int idleTime  // total time runway is idle.

A plane can only be added to the landing list if the list length is less than the list length limit. Otherwise, the plane is diverted to a different airport and service is refused.

The same applies to planes asking to takeoff. If the length of the takoff list is less than the listLimit, then only a plane is added to the takeoff list. Otherwise, the plane is asked to takeoff at a different time.

Think carefully about constructors. What should instance variables be set to before runway use? What activities take place on the runway?

3) class Simulation:

This class will provide a main method, keep track of the runway and planes in the simulation and will count of intervals representing units of time.

Things to remember:

- In any unit of time, one plane can land or one plane can takeoff, but not both.
- Planes arrive ready to land or to take off at random times, so at any given moment of time, the runway may be idle or a plane may be landing or taking off.
- More than one plane might arrive at the airport or ask to leave the airport in any given interval of time.
- There may be several planes waiting either to land or take off.
• Use Random class to generate requests from planes.