# Programming Assignment 1

PA1 Fixed Audio

Powered by Panopto

## Programming Assignment 1

This page gives details on the first programming assignment. There is a **discussion board available (https://uah.instructure.com/courses/52725/discussion_topics/304549)** if you have any questions about the assignment.

## Important Dates

### Date Posted: January 24

### Preliminary Class Diagram: February 14

### Class Outline: February 21

### Functionality Outline: February 28

### Final Project Due: March 14

## Background Information

Simulations of imagined or real situations are frequently created to provide an experimental environment for studies as wide ranging as astronomical events, battlefield encounters, and the clustering mentality of insects and birds. These types of software are most efficient when designed following an object oriented paradigm.

# Your Assignment

You are to create a program that simulates the flights of a number of airplanes. The basic requirements for this assignment are listed below.

1. The simulation scenario shall consist of a set of cities joined by flight paths and a set of defined airline flights between those cities.
2. When the simulation is running each one second of real time shall be one minute of simulation time. The simulation shall be capable of being run in 1X, 2X, or 3X time. This shall be user selectable at the start of the simulation.
3. Data on cities shall consist of the city name, state, latitude, longitude, and the distance in miles from each city to every other city in the scenario. The data shall be provided in a special data file. Source code for a data parser to read, parse, and provide the data for use in the scenario shall be provided by the instructor.
4. Data on all aircraft which could appear in the scenario shall consist of the aircraft make and model, cruise speed, cruise altitude, rate of climb, and size consisting of the wingspan and the fuselage length. The data shall be provided in a special data file. Source code for a data parser to read, parse, and provide the data for use in the scenario shall be provided by the instructor.
5. Data on all flights shall consist of the airline, aircraft type, flight number, departure city, destination city, and departure time. The data shall be provided in a special data file. Source code for a data parser to read, parse, and provide the data for use in the scenario shall be provided by the instructor.
6. When started the application shall ask the user to input the name of a text file giving the names of the two data files (City and Aircraft/Flight).
7. Whenever the clock time of the scenario indicates time for a flight to be departing or arriving a message shall be printed on the screen giving the current scenario clock time, the airline, flight number, type of aircraft, the departure city and state, the arrival city and state, and the estimated time of arrival.
8. At 5 second intervals a report shall be printed on the screen giving the current scenario clock time and information on all flights currently in route. Flight information shall include the airline, flight number, type of aircraft, the departure city symbol, latitude and longitude, the time of departure, the arrival city symbol, latitude and longitude, the estimated time of arrival, the current location of the aircraft in latitude and longitude, the number of miles from the departure city, the number of miles to the destination city, the cruise speed of the aircraft, and the current altitude.
9. An example implementation, including example simulation data files, city data files, and flight data files, **is available here (https://uah.instructure.com/courses/52725/files/4650093?wrap=1)** . This zip file also includes the source code to parse the simulation data files, city data files, and flight data files. **You should use those parsers in your program. You do not need to write your own parsers.**

# Deliverables

These products as specified below shall be delivered electronically via Canvas. More details about each of these deliverables is available from the **Programming Assignment General Information (https://uah.instructure.com/courses/52725/pages/programming-assignment-general-information)** page.

## Preliminary Class Diagram

The class diagram shall be drawn using standard UML notation and shall show all of the classes to be implemented in the software and their relationships (dependencies, associations, generalizations, realizations, etc.) The preliminary class diagram should be submitted as a PDF document via Canvas. **Here (https://uah.instructure.com/courses/52725/assignments/507096)** is a link to the Canvas submission page for the preliminary class diagram.

## Class Outline

The class outline shall list all proposed variables and functions in each proposed class with a brief description of what each does. The class outline should be submitted as a PDF document via Canvas. **Here (https://uah.instructure.com/courses/52725/assignments/507093)** is a link to the Canvas submission page for the class outline.

## Functionality Outline

The functionality outline shall be an outline which will show the step-by-step functionality of the program. This should be taken out to a fair amount of detail. The functionality outline should be submitted as a PDF document via Canvas. **Here (https://uah.instructure.com/courses/52725/assignments/507095)** is a link to the Canvas submission page for the functionality outline.

## Final Project

The entire software project (compatible with Microsoft Visual Studio 2017) shall be compressed into a zip, 7z, or rar file and submitted via Canvas. **Just turning in your source files is not acceptable.** **Here (https://uah.instructure.com/courses/52725/assignments/507094)** is a link to the Canvas submission page for the final assignment.