

CST8132: Object-Oriented Programming

Midterm Test: Winter 2008: Part 2: Programming

Name: _____

On separate sheets of paper, write a program and draw a memory map to perform the following. When you submit your answer, I will staple it to this paper.

Program Coding (20 marks)

Passenger Class Hierarchy

Create a hierarchical class structure to handle passengers for an airline company. The class organization must use *inheritance* and *virtual functions* where appropriate. The base class will be called **Passenger**. There will be two derived classes: **Economy** and **FullFare**. All passengers will have the following data members.

Data Members Common to all Passenger Types

- **szLastName**: up to 15 characters in the last name. (Note: To simplify your work during this test, you can make several assumptions. There will be no intervening spaces in strings, thus you can use the simplest possible input routines – the ones that stop input at whitespace. You can also assume that no input will exceed the 15 character limit.)
- **dBasePrice**: floating point number.
- **dFinalPrice**: floating point number.

Data Member Specific to *FullFare* Passengers

- **IPointsEarned**: long number: **FullFare** passengers are also entitled to frequent flyer points.

Member Functions

- You will implement *virtual functions* to support *reading data from the file* and to support the *display of objects to the screen*. There are several correct approaches to implement these functions. I'll let you decide which approach to use.
- To keep this program simple (because of time constraints), no other functions are needed in the **Passenger** class hierarchy (no constructors, no destructors etc.).

Flight Class

The **Flight** class is a container that will hold a list of passengers.

Data Members

- **szFlight**: up to 8 characters in the flight identifier.
- **nNumPassengers**: a count of the current number of passengers on this flight.
- pointer to hold a dynamically allocated array of pointers to **Passenger** objects.

Member Functions

You will implement four functions:

- **Constructor**:
- **Destructor**:
- **LoadFlightData**:
- **Display**:

File:
FlightData.txt

```
AC747 5
fSmith 288.88 594.35 1111
eUnger 199.99 535.30
fJones 883.75 984.73 21192
eTyler 332.22 636.73
fCrane 456.78 688.45 7428
```

Main Routine

The function **main()** will be very simple. I've supplied it here.

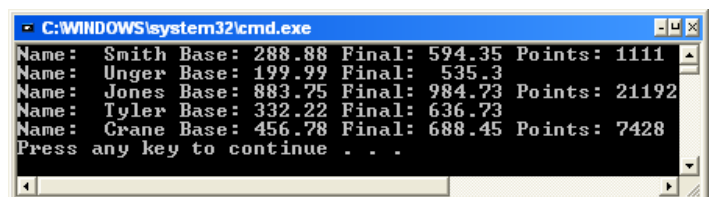
```
void main()
{
    Flight fAirCanada123;
    fAirCanada123.LoadFlightData("FlightData.txt");
    fAirCanada123.Display();
}
```

Memory Map (2 marks)

Draw a memory map that shows the organization of memory after the **LoadFlightData()** function has been called.

Sample Output

The following screen shot shows my output. Yours does not need to look exactly the same, but you may find it useful.



```
C:\WINDOWS\system32\cmd.exe
Name: Smith Base: 288.88 Final: 594.35 Points: 1111
Name: Unger Base: 199.99 Final: 535.3
Name: Jones Base: 883.75 Final: 984.73 Points: 21192
Name: Tyler Base: 332.22 Final: 636.73
Name: Crane Base: 456.78 Final: 688.45 Points: 7428
Press any key to continue . . .
```