**CSCI 2251  
Programming Assignment 2 – RentalDue**

This assignment has the following objectives:

1. to apply inheritance to objects
2. to implement an interface
3. to apply polymorphism to the class hierarchy
4. to read data from file and update the database file
5. to generate a UML class diagram with object-oriented design

**Problem Description**

A real estate investor started a residential rental business. They own a few single-family houses and a few apartment complexes. The information about their rental properties are listed in a database file, rentalDB.txt.

The company decides to increase the rent of single-family houses by 4% and the rent of apartment units by 8%. You are tasked with writing a program to compute the new rent amounts, update the database file, and output the new monthly rent summary on the screen. In your program, you need to have the following:

1. An interface, Payment, with two abstract methods: one to update the rent (either a 4% or 8% increase) and one to get the rent. The first method modifies the rent instance variable and returns void. The second method modifies nothing and returns a double.
2. A super class, RentalProperty, for the two different rental properties. This class **implements** the interface Payment.
3. Two subclasses: SingleFamilyRental and ApartmentRental for the two different rental properties. These should **extend** the RentalProperty and override the update rent method.
4. Apply polymorphism (for example, you should keep only one array of rental properties RentalProperty[] rentals = new RentalProperty[numProperty]) to update the database file, rentalDB via the method updateRent(). The updated database file should retain its original record format as well as sequence of records.
5. Lastly, output the new monthly rent summary on the screen.

Hint: identify how many rental properties are in the database file before initializing the array.

You need to open the database file, rentalDB.txt, and get the current rental information for your program. In the database file each line represents a data record with six fields per record. All fields are right justified. The format of rentalDB.txt is as follows:

Field 1 (6 characters): Unique sequence number.

Field 2 (4 characters): rental type – S means single-family rental, A means apartment rental

Field 3 (10 characters): rental ID – a 7 character rental property identification (first letter is the rental type, next three letters are the rental location, next three digits are the rental ID of the location)

Field 4 (3 characters): number of bedrooms.

Field 5 (10 characters): monthly rent

Example contents of the database file:

1 S SABQ138 3 1400.00

2 A AABQ205 2 900.00

3 S SABQ127 1 900.00

4 S SABQ126 2 1200.00

5 A AABQ302 2 850.00

6 A AABQ201 1 600.00

The interpretation of the above records is:

1, Single-family rental, ID number SABQ138, three bedrooms, monthly rent $1,400.00

2, Apartment rental, ID number AABQ205, two bedrooms, monthly rent $900.00

3, Single-family rental, ID number SABQ127, one bedroom, monthly rent $900.00

. . .

**After your program acquires the data from the file, it needs to update the database file as follows. Note that the updated database file must have the same record format and the sequence as the original records:**

1 S SABQ138 3 1456.00

2 A AABQ205 2 972.00

3 S SABQ127 1 936.00

4 S SABQ126 2 1248.00

5 A AABQ302 2 918.00

6 A AABQ201 1 648.00

**Also, your program needs to output rental summary on the screen like the following (sorted by field 3: rental ID):**

Single-Family Rental Summary:

House ID Number Rental Due

=============== ==========

SABQ126 $1,248.00

SABQ127 $936.00

SABQ138 $1,456.00

Apartment rental Summary:

Apartment ID No. Rental Due

================ ==========

AABQ201 $648.00

AABQ205 $972.00

AABQ302 $918.00

**Specifications**

Your program must meet the following specifications:

* Work on your own
* The name of the source code file must be the same as the class names described above. You need to have a RentalDueTest class (file name RentalDueTest.java) for the driver class that contains the main code.
* Three classes: RentalProperty, SingleFamilyRental and ApartmentRental.
* Comments at the top of RentalDueTest with a short description of your program, your name, e-mail, date and course title.
* Two part submissions:
  + Part I: RentalProperty.doc – the file contains a description on how to solve this problem, the UML class diagrams showing the class hierarchy, control flow, and related charts/illustrations (if you have any).
  + Part II: java source code file(s) – Create a Rental.zip file for all your java source files.
* Command line only, do not use graphic user interface (GUI).

I will test your program as follows:

javac RentalDueTest.java

java RentalDueTest