**CSCI 2251  
Programming Assignment 2 – RentalDue**

This assignment has the following objectives:

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

**Problem Description**

A real estate investor in Albuquerque, New Mexico started their residential rental business since 2016. Their business expanded from a few single-family houses to a couple of dozens of single-family houses plus a few of apartment complexes. The information about their rental properties are listed in a database file, rentalDB, which include monthly rent. The monthly rent varies from $600.00 to $1,400.00 based on the rental unit configurations, such as property type and number of bedrooms.

Due to the increase of business tax and other expenses, the company decided to increase the rent of single-family house by 4% and the rent of apartment unit by 8%. You are requested to write a program to compute the rent for their rentals, update the database file and output the new monthly rent summary on the screen. In your program, you need to have at least the followings:

1. An interface, Payment, to calculate the rent of the two rental types.
2. A super class, RentalProperty, for the two different rental properties. It **implements** the interface Payment.
3. Two subclasses: SingleFaimlyRental and ApartmentRental for the two different rental properties. They should **extends** the RentalProperty.
4. Apply polymorphism (such as RentalProperty[] rentals = mew 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 the records. It also output the new monthly rent summary on the screen.

Hint: identify how many rental properties (how many records), numProperty, in the database file first.

You need to open the database file (plain text file), rentalDB, and get the current rental information for your program. In the database file, a line represents a data record, six fields per records, all fields are right justified. The format of rentalDB 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 characters rental property identification (first letter is the rental type, next three letters is the rental location, next three digits is the rental ID of the location)

Field 4 (3 characters): number of bedrooms – it means how many bedrooms of the rental.

Field 5 (10 characters): monthly rent

Example of the database file (the content of rentalDB.txt) as follow:

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 interpretations of the above records are:

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

. . .

Note: you need to use text editor (example notepad++) to verify the position of each field

**Assume after your program acquire data records from the above database file, it needs to update the database file, rentalDB, as follows. Note that the updated database file must have the same record format and the sequence of original records has to be maintained:**

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 the 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 as described above. You need to have a RentalDueTest class (file name RentalDueTest.java) for the driver class that contains the main code can be simply just invoking other methods based on your design.
* Three classes: RentalProperty, SingleFaimlyRental and ApartmentRental. Need to implement the interface Payment.
* Comments at the top with short description of your program, your name, e-mail, date and course title.
* We expect programming assignments to be implemented using Java 1.8. Your code will be tested on the machines, Windows or Linux, installed the same Java 1.8 version (Java compiler and Java Virtual Machine). Make sure your code runs on at least one of those machines.
* Two part submissions (Blackboard):
  + Part I: RentalProperty.doc – the file contains a description on how to solve this problem, the UML class diagrams (Word document, Open Office, or others) showing the class hierarchy, control flow, and related charts/illustrations (if you have any).
  + Part II: java source code file(s) – Create a Rental.jar file for all your java source (the .java) files, submit the Rental.jar only.

Example of create Rental.jar (assume all java files are in the same folder/directory):

jar cvf Rental.jar \*.java

* Command line only, do not use graphic user interface (GUI). I will not accept any GUI program.
* No late, refer to the syllabus for the late policy.

I will test your program as follows (after download your Rental.jar in a new folder/directory):

jar xvf Rental.jar

javac RentalDueTest.java

java RentalDueTest