**CMSC214 Project 4**

**Task for Project 4**

**Project Objective:**

* Effectively use Event-Driven Programming
* Create graphical user interface
* Effectively use I/O processed in Java
* Write a program that meets the following requirements(see below)

**Instructions:**

In addition to the goals of being able to write effective Java programs.

**Deliverables:**

In this programming project the student will design, develop, test and document a Java application.

**Projects submitted with evidence of plagiarism will be given a score of 0.**

**Description of a Project (Address book):**

* Supplement VI.B (see attachment) gives a case study of using random-access files for creating and manipulating an address book.
* Modify the case study by adding an *Update* button, as shown in Figure 1, to enable the user to modify the address that is being displayed.

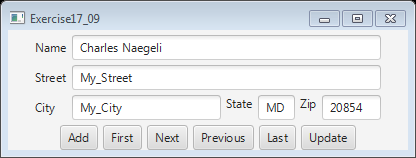


FIGURE 1 The application can store, retrieve, and update addresses from a file

**Code Documentation and Style Requirements**The documentation requirement for all programming projects is one block comment at the top of the program containing the course name, the project number, your name, the date and platform/compiler that you used to develop the project. In addition, there should be at least one comment for each class in the program describing what that class does. Additional comments should be provided as necessary to clarify the program.

Indentation must be consistent throughout the program. Variable and method names should be descriptive of the role of the variable or method. Single letter names should be avoided. All constants, except 0 and 1, should be named. Constant names should be all upper-case. Variable names should begin in lower-case, but subsequent words should be in title case (e.g., finalSpeed).

Separate compilation must be used in accordance with standard Java practice. Every class must be saved in a separate .java file

**Rubric for Project 4**

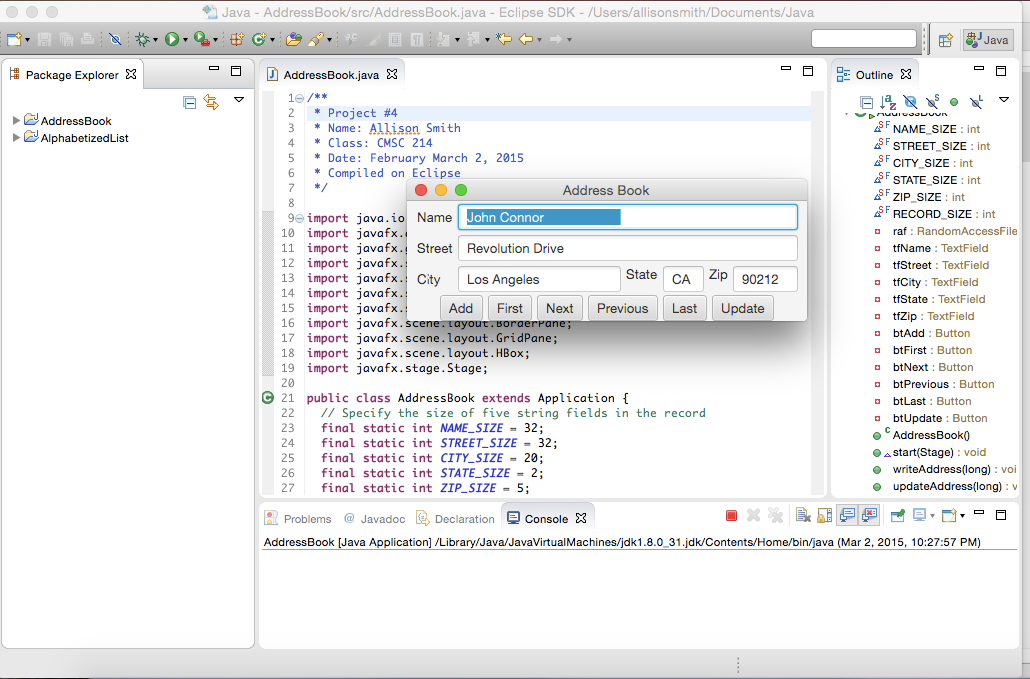
**Grading:**

This project has three parts, with **100 total points** possible.

  (40 points) Carefully declare all the variables with appropriate data types. Use comments effectively to make the program more readable. Implement the calculations correctly.

  (40 points) Compile and build the Java program. Test the program using the Test Plan as described above and make sure the program produces the desired answer.

  (20 points) Submit your source code, screenshots, UML diagram, and algorithm as attached files to Project 4 Folder

Screenshot

***Hints: use***

* ***Listing 17.9, AddressBook.java(see attachment)***
* ***SupplementAddressBook.pdf/doc***
* ***Solution exercise 17.8(see attachment)***