

Practice Examples for Lab: Set 14

• 1

Define a `struct` for storing complex numbers. Define functions for arithmetic on complex numbers.

- Write test cases in the main program to test out addition, subtraction, multiplication, and division of two complex numbers input from the user.

• 2

Define a structure for representing axis parallel rectangles, i.e. rectangles whose sides are parallel to the axes. An axis parallel rectangle can be represented by the coordinates of the diagonally opposite points. Write a function that takes a rectangle (axis parallel) as the first argument and a point as the second argument, and determines whether the point lies inside the rectangle. Write a function which takes a rectangle and double values `dx`, `dy` and returns a rectangle shifted by `dx`, `dy` in the `x` and `y` directions respectively.

Practice Examples for Lab: Set 14

- 3

Define a structure class for storing information about a book for use in a program dealing with a library. The class should store the name, author, price, a library accession number for the book, and the identification number of a library patron (if any) who has borrowed the book. This field, patron identification number could be 0 to indicate that the book is not borrowed.

Read information about books from a file into an array of `book` objects. Then you should enable patrons to issue and return books. When a patron issues/returns a book, the patron identification number of the book should be changed. Write functions for doing this. The functions should check that the operations are valid, e.g. a book that is already recorded as borrowed is not being borrowed without first being returned.

Practice Examples for Lab: Set 14

- 4

Define a **struct** for storing dates. Define a function which checks whether a given date is valid, i.e. the month is in the range 1 to 12, and the day is a valid number depending upon the month and the year.

- 5

Write a program to answer queries about ancestry. Your program should read in a file that contains lines giving the name of a person (single word) followed by the name of the father (single word). Assume that there are at most 100 lines, i.e. 200 names. After that, your program should receive a name from the keyboard, and print all ancestors of the person, in the order father, grandfather, great grandfather and so on as known.