

Activity 01 from Lab 03 Slides.

Activity 02:

Create a class called Time that has separate int member data for hours, minutes, and seconds. Provide the following member functions for this class:

- a. A no-argument constructor to initialize hour, minutes, and seconds to 0.
- b. A 3-argument constructor to initialize the members to values sent from the calling function at the time of creation of an object. Make sure that valid values are provided for all the data members. In case of an invalid value, set the variable to 0.
- c. A member function show to display time in 11:59:59 format.
- d. A function AddTime for addition of two Time objects. Each time unit of one object must add into the corresponding time unit of the other object. Keep in view the fact that minutes and seconds of resultant should not exceed the maximum limit (60). If any of them do exceed, subtract 60 from the corresponding unit and add a 1 to the next higher unit.

NOTE: Define all the member functions outside the class.

A main() programs should create two initialized Time objects and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third Time variable. Finally it should display the value of this third variable.

Activity 03

Create a class called Distance containing two members feet and inches. This class represents distance measured in feet and inches. For this class, provide the following functions:

- a) A no-argument constructor that initializes the data members to some fixed values.
- b) A 2-argument constructor to initialize the values of feet and inches to the values sent from the calling function at the time of creation of an object of type Distance.

- c) AddDistance function to add two distances: Feet and inches of both objects should add in their corresponding members. 12 inches constitute one foot. Make sure that the result of addition doesn't violate this rule.
- d) isGreater: should return a variable of type bool to indicate whether 1st distance is greater than 2nd or not.
- e) isSmaller: should return a variable of type bool to indicate whether 1st distance is smaller than 2nd or not.
- f) isGreaterEqual: should return a variable of type bool to indicate whether 1st distance is greater than or equal to 2nd or not.
- g) isSmallerEqual: should return a variable of type bool to indicate whether 1st distance is smaller than or equal to 2nd or not.
- h) isEqual: should return a variable of type bool to indicate whether 1st distance is equal to the 2nd distance or not.
- i) isNotEqual: should return a true value if both the distances are not equal and return a false if both are equal.

NOTE: Define all the member functions outside the class.

Activity 04

Create a class called Matrix that represents a 3x3 matrix. This matrix contains a two-dimensional integer array of size 3x3. Provide the following member functions for this class:

- a) a no-argument constructor for initializing the matrix with 0 values.
- b) A one-argument constructor to initialize the member matrix to the matrix sent as an argument from the calling function.
- c) An AddMatrix function for addition of two matrices
- d) A MultiplyMatrix method for finding the product of the two matrices.
- e) An isEqual function for checking the equality of two matrices

Note: Define all the member functions outside the class.