iOS Hands-On 2

Swift

Task 1

- Create a Calculator Application with the following specs:
  - Create calculator class with the following
    - var param1
    - var param2
    - String operation
    - getResult method
  - Create 4 objects of calculator to illustrate +,-,*,/

Task 2

- Create a friend list application with the following elements:
  - Class Friend: first name, last name, phone, email, age, closeness value [1-10]
  - Array of 5 friends
  - Printer Class with method print(friend) to print data in a good readable way
  - Use the above elements to print friends ordered by age, name and closeness value

Task 2 - cont.

- Create a class FriendStats with the following functions
  - sameName: prints friends with same names
  - relatives: prints friends of the same family

Task 3

- Create an Application with the following specs:
  - Shape class with
    - func getArea
  - Circle class child of Shape with
    - var radius
    - func get area

Task 3 - cont.

- Square class child of Shape with
  - var length
  - func get area
- Rectangle child of Shape with
  - var length, var width
  - func getArea

Task 3 - cont.

- Use the previous classes as following:
  - Create two objects of each class
  - Set parameters with suitable values
  - Print values of their areas

Task 3

- Update previous task as follows:
  - Create class MeterJudge with
  - var shapes as array of shapes
  - func getLargest
  - func getSmallest

Task 4

- Create an object of MeterJudge
- Pass the previous created objects
- Use largest and smallest methods to print the min and max area values

Extra

Task 5

- Create an app with the following classes
  - Class Employee
  - Class Manager as sub class of Employee
  - Class HRManager as sub class of Manager
  - Class Company
  - Class DayOffRequest

Extra

Task 5 - cont.

- Class Employee has the following :
  - attribute : name
  - attribute : email
  - attribute: salary
  - attribute : manager
  - methods : setters and getters for above attributes

Extra

Task 5 - cont.

- Class Manager extends Employee
  - methods : send request

Extra

Task 5 - cont.

- Class HRManager extends Employee
  - methods : override send request

Extra

Task 5 - cont.

- Class Company has the following
    - Array of employees
    - Array of managers
    - HR Manager

Extra

Task 5 - cont.

- Class DayOffRequest has the following
  - attribute : requester employee
  - attribute : status
  - attribute : type [can be casual or emergency]

Extra

Task 5 - cont.

- In main function the following needs to be created:
  - An object of company with the following:
    - Two employees with two different managers
    - An HR Manager

Extra

Task 5 - cont.

- Main Function should simulate the following scenario:
  - employee1 requests an emergency day off from his manager who will forward the request to HR manager who would approve it
  - employee2 requests a casual day off from his manager who will forward the request to HR manager who would decline it
  - print both request summaries