Class

- Create a swift program to demonstrate swift class. Which has two variables inside class body. Access (Set its values and get its value.) these variables by creating one instance of class.

- Create a swift program to demonstrate usage of swift class for usage of multiple initializers. Create one initializer with two parameters your name and college and another initializer with your name and department. You can be able to access these properties using instance of class.

- Create a swift class without initializers and access (**write, read**) its properties using instance of class.

- Create a swift class which is having code to return square of given number and access this code using class instance.

- Create a swift class example to show use of swift inheritance. Your base class has some property access this property. In two different child classes and show its usage using the instances of both child classes.

- Create a swift class example to show use of swift base class which have some implementation inside any method. Now demonstrate usage of overriding that method implementation.

- Create a swift class with an initializer in a manner that you can create this class using initializing value. And one function which takes int input and returns the power of class instance.

example: let instance = Example(value: 5)
let result = instance.doPower(power: 3)
then it should return 5's 3 power (125)

- Create a swift class example which has a parent class for vehicles, and child classes for two different vehicles e.g. (bike & car). You need to use common properties and method in the parent class and inside child class, there will be some different property which is not common.

- Create a person swift class with person name initializer and create one function to greet that person.

Example: let person1 = Person(name: Joe)
Person1.greet() // This should print “Hello Joe”

- Create a swift class with some property. And then set its value using initializer of class, then perform below actions.

let example1 = Example()
Example1.a = “Hello”
let example2 = example1

example2.a = “Hi”
print(example1.a)
print(example2.a)
// Consider ‘a’ as string property of example class.
--> Check these two print outputs and try to find out the reason for that output.

Structure

- ***Create one structure of type Work, which have some properties like work location, work hours and one string array of project names. And create a function inside structure which returns all project names for the person.***

- ***.Create a structure example which demonstrates use of initializer in structures***

- ***Create a structure program which can be initialized with parameters and it also .should be able to initialized without parameters***

- ***Create one structure which have initializer which takes array of int as input returns .two arrays one of all even numbers and another is all odd numbers***
Input: [1,3,5,6,8,10,9,7,8,12]
Output: even numbers are: [6,8,10,8,12]
Odd numbers are: [1,3,5,9,7]

- ***Create one swift structure program for person that contains basic details like name, age, gender. Then create one array of person with all details. Print all details of that .array***
.Consider two persons array for Joe and Harry. But in your code Its size can be vary //
Output be like: Name: Joe, Gender: Male, Age: 27
Name: Harry, Gender: Male, Age: 21

- ***Perform same example given in class question no 10 and check output using structure. Try to find difference between these two outputs and try to figure out the .reason for that output***

- ***Try to figure out basic difference between class and structures and give a demo for .same***

Enumerations

- ***Write swift program using enumerations to check week day from given number. Consider Monday as 1 and Sunday as 7. We can be able to get day name as string in short form (sun, mon,...) and same full name (Sunday, Monday...) by given number .of the day***
.Example. It should return Monday or mon if I pass parameter 1 for get day name

.Create one enumeration program to return number of days in a month (2

.Write a swift program using enumerations to demonstrate Int enums (3

.Write a swift program to demonstrate string type enum (4

.Write a swift program for enum with raw values (5

Write a swift program using enumerations to get all cases as an array with use of (6
.CaseIterable(Optional) and without use of CaseIterable(Mandatory)

Write a swift program using enumerations for learn and demonsrate enum cases (7
.with parameters (Enum with associated values)

Create an enum with its rawValues of type String and show usage of case to print (8
.value of case

Properties Subscripts Methods & Inheritance

- ***Create an example using swift to demonstrate use of get only properties. Create a class (1***
  .with a property which value cannot be set outside of class

Output: for example, your class Demo has one property of string type then you can be able to set it
.within a class. If I try set it outside of class it must give an error

- ***Create a swift program to demonstrate usage of computed properties using getter and (2***
  .setter
- ***.Create a swift program to show usage of stored properties (3***
- ***Create a swift program which has private properties which cannot directly be accessed. (4***
  .Means we cannot directly read or write it
- ***Create one swift class which have two properties name and id, the class must have one (5***
  .initializer to set that properties. And create one array of that class types
- ***Output: My class is Person and properties are id and name, then create an array of person.***
  .And print all elements of that array
- ***.Create one example of usage of willSet and didSet (6***
- ***.Create one lazy stored property in a class and show usage of it (7***
- ***Create one class as Base type Person which has common properties like name, (8***
  occupation, etc. Create two child classes from the person which are Student, Employee and

this two-child classes must have base properties and some other properties also. Example, student have college, but Employee have company. And demonstrate the usage of inheritance.

- Create one structure to show usage of mutating function in swift (9.
- Create one class inheritance demo to show usage of method overriding (10.
- Create one swift class to show usage of type methods (11.
- Create one swift class which is having class method and static method. Then in one child class try to override that methods and check the output/ error. (you will learn difference between class and static) (12
- Create one example of subscript using array. Create one array of weekdays and one subscript func which takes int as argument and returns day of week (13.
- Output: WeekDays.subscript(day: 1) // will return Sunday Note: Handle invalid input.
- Create a swift program to show usage of subscript in string. I pass int argument and it returns the character at given position (14.
- Create one swift subscript program which takes range as input and returns the string between the ranges (15.
- I have one integer array and create one function which takes range as input and returns all elements between the range (16.
- I have one key value pair array. Create one function using subscript which takes key as input and returns it's value (17.
- Output: let array = [[1: "Hello"], [2: "Hi.."]] "Now call function subscript(at: 1) it should print "Hello
- Create one array of type Person and create one subscript function which takes person name as input and returns person info like name, age, birthdate etc (18.
- Create one base class of type Song and create subclasses of music types (Hip-Hop, classical) and show usage of inheritance. // Here Music class have property singer, composer (19
- Create a swift class with properties which can be read-write outside of class (20.

~~Error Handling~~

- Create a password validation program. If length of password is less than 8 characters, it throws "Password too short" exception and if password is empty, throw "Empty password" exception.
- Create a program for shopping cart. If desired quantity for an item is not available, throw exception.

~~Extensions~~

- Create following string manipulation functions using string extensions:
  - Add a character in a string at 5^{th} position
  - Replace one character with other in a string

- ***Remove white spaces from string***
- ***Get number of words in a string***