**Swift Advance Exercise**

**Ques1. What is extension?**

**Ans.** Extension in Swift is a feature which is used to add the new fuctionality to an existing class, structure, enumeration or protocol type. It has the ability to extend types for which we do not have access to the original source code. The functionalities of Extension are:

1. Add computed instance properties and computed type properties
2. Define instance methods and type methods
3. Provide new initializers
4. Define subscripts
5. Make an existing type conform to a protocol

**Syntax:-** extension SomeType{

//New Functionality

}  
 Here **extension** is keyword and **SomeType** is either the existing structure, class, enum, or protocol.

Ques2. Create a class and write the delegate of UITextField in extension of that class

**Shown in playground**

Ques3. Write a protocol and create an extension of the protocol. In extension create a function

func sayHello() {

print(“Hello!”)

}

**Shown in playground**

Ques4.Write an enum and create an extension of the enum.

**Shown in playground**

5. What is Generic?

**Ans.** Generic code gives us the facility to write flexible and reusable functions and types. Generics are used to avoid duplication and to provide abstraction.

**For Example:-** Array and Dictionary both are Generic collections because array can be created of Int type , String type or any other type and same case is with dictionary as it can be of any type.

6. Explain generic with an example?

**Shown in playground**

7. Explain the difference between map and compactMap with an example.

**map:** map is an higher order function that is used to transform from one thing to another.This type of function loops over a collection and applies the same operation to each element in the collection.

**compactMap:** compactMap is also a higher order function which is similar to map but the only difference is that it is used to filter out nil value.

**Example:** Both the examples are shown in playground

8. Write an example of reduce function with initial value 1000.

**Shown in playground**

9. - 2 marks

struct Person {

var name : String

var age : Int

}

let person1 = Person(name: "Sam", age: 23)

let person2 = Person(name: "John", age: 30)

let person3 = Person(name: "Rob", age: 27)

let person4 = Person(name: "Luke", age: 20)

let personArray = [person1, person2, person3, person4]

Find all person whose age is more than 25 using filter function.

**Shown in playground**

10. Make a property wrapper @nonNegative and use it to make values to 0 if any negative value added to a variable

**//This topic is not explained during the session this topic will be explained later.**