**// Initializers**

1. Implement the parameterised initialisation with class or struct.

**Showed in playground Swift Basic 2 Assignment**

1. Write all the Rules of initialiser in Inheritance

Ans. **There are three rules of initialiser in Inheritance:-**

**(a)**A designated initializer must call a designated initializer from its immediate superclass.

**(b)**A convenience initializer must call another initializer from the *same* class.

**(c)**A convenience initializer must ultimately call a designated initializer.

1. Using convenience **Initializers**, write-down the **Initializers** for MOVIE class having basic attributes like title, author, publish\_date, etc.

**Showed in playground Swift Basic 2 Assignment**

1. Declare a structure which can demonstrate the throwable Initializer

**Showed in playground Swift Basic 2 Assignment**

**// Array**

1. Create an array containing the 5 different integer values. Write are at least 4 ways to do this.

**Showed in playground Swift Basic 2 Assignment**

1. Create an immutable array containing 5 city names.

**Showed in playground Swift Basic 2 Assignment**

1. Create an array with city 5 city names. Later add other names like Canada, Switzerland, Spain to the end of the array in at least 2 possible ways.

**Showed in playground Swift Basic 2 Assignment**

1. Create an array with values 14, 18, 15, 16, 23, 52, 95. Replace the values 24 & 48 at 2nd & 4th index of array

**Showed in playground Swift Basic 2 Assignment**

**//Set**

1. Given the following sets:

let houseAnimals: Set = ["🐶", "🐱"]

let farmAnimals: Set = ["🐮", "🐔", "🐑", "🐶", "🐱"]

let cityAnimals: Set = ["🐦", "🐭"]

**Use set operations to...**

1. Determine whether the set of house animals is a subset of farm animals.
2. Determine whether the set of farm animals is a superset of house animals.
3. Determine if the set of farm animals is disjoint with city animals.
4. Create a set that only contains farm animals that are not also house animals.
5. Create a set that contains all the animals from all sets.

**All are** **Showed in playground Swift Basic 2 Assignment**

**// Dictionary**

1. Create an empty dictionary with keys of type String and values of type Int and assign it to a variable in as many ways as you can think of (there's at least 4 ways).
2. Create a mutable dictionary named secretIdentities where the key value pairs are "Hulk" -> "Bruce Banner", "Batman" -> "Bruce Wayne", and "Superman" -> "Clark Kent".
3. Create a nesters structure of Key-value pair.
4. Print all the keys in the dic

**All are** **Showed in playground Swift Basic 2 Assignment**

**Subscript :**

1. What is subscript ? Write down the declaration syntax.

Ans. Classes, structures, and enumerations can define subscripts, which are shortcuts for accessing the member elements of a collection, list, or sequence. We use subscripts to set and retrieve values by index without needing separate methods for setting and retrieval. We can define multiple subscript for a single type.

**Syntax:-** subscript(index: Int) -> Int {

get {

// Return an appropriate subscript value here

}

set(newValue) {

// Perform a suitable setting action here

}

* }

1. Create a simple subscript that outputs true if a string contains a substring and false otherwise.

**Showed in playground Swift Basic 2 Assignment**