**Assignment - 6**

**(Class, Inheritance, Collection, Loops)**

Name : Om Kulkarni Date : 17/02/25

Class : TY CSE

PRN : 22510034 Batch : T-3

Source Codes :

// Q1

class Car {

var brand: String

var model: String

var year: Int

init(brand: String, model: String, year: Int) {

self.brand = brand

self.model = model

self.year = year

}

}

let myCar = Car(brand: "Toyota", model: "Camry", year: 2020)

print("Car Details: \(myCar.brand) \(myCar.model), Year: \(myCar.year)")

// Q2

class Car {

var brand: String

var model: String

var year: Int

init(brand: String, model: String, year: Int) {

self.brand = brand

self.model = model

self.year = year

}

func displayDetails() {

print("Car Details: \(brand) \(model), Year: \(year)")

}

var age: Int {

let currentYear = Calendar.current.component(.year, from: Date())

return currentYear - year

}

}

let myCar = Car(brand: "Toyota", model: "Camry", year: 2020)

myCar.displayDetails()

print("Car Age: \(myCar.age) years")

// Q3

class ElectricCar: Car {

var batteryCapacity: Int

init(brand: String, model: String, year: Int, batteryCapacity: Int) {

self.batteryCapacity = batteryCapacity

super.init(brand: brand, model: model, year: year)

}

override func displayDetails() {

super.displayDetails()

print("Battery Capacity: \(batteryCapacity) kWh")

}

}

let myElectricCar = ElectricCar(brand: "Tesla", model: "Model S", year: 2022, batteryCapacity: 100)

myElectricCar.displayDetails()

// Q4

class Animal {

var name: String

var age: Int

init(name: String, age: Int) {

self.name = name

self.age = age

}

}

class Dog: Animal {

var breed: String

init(name: String, age: Int, breed: String) {

self.breed = breed

super.init(name: name, age: age)

}

}

let myDog = Dog(name: "Buddy", age: 3, breed: "Labrador")

print("Dog Name: \(myDog.name), Age: \(myDog.age), Breed: \(myDog.breed)")

// Q5

class Laptop {

var brand: String

var processor: String

var ramSize: Int

init(brand: String, processor: String, ramSize: Int) {

self.brand = brand

self.processor = processor

self.ramSize = ramSize

}

}

class GamingLaptop: Laptop {

var graphicsCard: String

init(brand: String, processor: String, graphicsCard: String, ramSize: Int = 16) {

self.graphicsCard = graphicsCard

super.init(brand: brand, processor: processor, ramSize: ramSize)

}

}

let myGamingLaptop = GamingLaptop(brand: "ASUS", processor: "Intel i7", graphicsCard: "NVIDIA RTX 3060")

print("Laptop Details: \(myGamingLaptop.brand), \(myGamingLaptop.processor), RAM: \(myGamingLaptop.ramSize)GB, GPU: \(myGamingLaptop.graphicsCard)")

// Q6

var numbers = [1, -3, 50, 72, -95, 115]

print(numbers.contains(72)) // true

print(numbers.contains(95)) // false

// Q7

var zeroArray = Array(repeating: 0, count: 20)

print(zeroArray)

// Q8

var fruits = ["Apple", "Banana", "Cherry"]

print(fruits.isEmpty ? "Array is empty" : "Array has \(fruits.count) elements")

// Q9

var cricketers = ["Sachin", "Rahul", "Rohit", "Virat"]

cricketers[0] = "Yuvraj"

cricketers.append("Shubman")

cricketers += ["Ravindra", "Hardik"]

cricketers.insert("Suresh", at: 5)

cricketers.remove(at: 4)

cricketers.removeLast()

print(cricketers)

// Q10

var nameAndAge = [["Sachin", "Rahul", "Rohit"], [40, 38, 35]]

print("First Cricketer: \(nameAndAge[0][0]), Age: \(nameAndAge[1][0])")

// Q11

var playerScores = ["Sachin": 50000, "Hardik": 4000, "Ravindra": 8000]

print(playerScores)

// Q12

let oldScore = playerScores["Hardik"]

playerScores["Virat"] = 25000

playerScores["Hardik"] = 9000

print("Old Score of Hardik: \(oldScore!)")

print(playerScores)

// Q13

if let removedScore = playerScores.removeValue(forKey: "Sachin") {

print("Removed Sachin with score: \(removedScore)")

}

print(playerScores)

// Q14

let players = Array(playerScores.keys)

let scores = Array(playerScores.values)

print("Players: \(players)")

print("Scores: \(scores)")

// Q15

let cities = ["New York", "London", "Paris", "Tokyo", "Sydney"]

for city in cities {

print(city)

}

// Q16

let number = 5

var factorial = 1

for i in 1...number {

factorial \*= i

}

print("Factorial of \(number) is \(factorial)")

// Q17

let vehicles = ["unicycle": 1, "bicycle": 2, "tricycle": 3, "quad bike": 4]

for (vehicle, wheels) in vehicles {

print("\(vehicle) has \(wheels) wheels")

}

// Q18

var numberToCheck = 17

var isPrime = numberToCheck > 1

var divisor = 2

while divisor \* divisor <= numberToCheck {

if numberToCheck % divisor == 0 {

isPrime = false

break

}

divisor += 1

}

print(isPrime ? "Prime" : "Not Prime")

// Q19

var originalNumber = 1234

var reversedNumber = 0

while originalNumber > 0 {

let digit = originalNumber % 10

reversedNumber = reversedNumber \* 10 + digit

originalNumber /= 10

}

print("Reversed number: \(reversedNumber)")

// Q20

var numberToCheckArmstrong = 153

var tempNumber = numberToCheckArmstrong

var sumOfDigits = 0

let numberOfDigits = String(numberToCheckArmstrong).count

while tempNumber > 0 {

let digit = tempNumber % 10

sumOfDigits += Int(pow(Double(digit), Double(numberOfDigits)))

tempNumber /= 10

}

print(sumOfDigits == numberToCheckArmstrong ? "Armstrong Number" : "Not an Armstrong Number")

Output :



