1. **Create abstract class Order (id, description). Derive two classes Purchase Order & Sales Order with members Vendor and Customer. Create object of each Purchase Order and Sales Order. Display the details of each account.**

// Abstract class Order

abstract class Order(val id: Int, val description: String) {

// Abstract method to display order details

def displayDetails(): Unit

}

// Class for Purchase Order with Vendor as a member

class PurchaseOrder(id: Int, description: String, val vendor: String) extends Order(id, description) {

// Override method to display purchase order details

override def displayDetails(): Unit = {

println(s"Purchase Order ID: $id, Description: $description, Vendor: $vendor")

}

}

// Class for Sales Order with Customer as a member

class SalesOrder(id: Int, description: String, val customer: String) extends Order(id, description) {

// Override method to display sales order details

override def displayDetails(): Unit = {

println(s"Sales Order ID: $id, Description: $description, Customer: $customer")

}

}

// Main object to test the program

object OrderTest {

def main(args: Array[String]): Unit = {

// Create an object of PurchaseOrder

val purchaseOrder = new PurchaseOrder(1, "Office Supplies", "XYZ Corp")

// Create an object of SalesOrder

val salesOrder = new SalesOrder(2, "Laptop Sale", "ABC Inc")

// Display the details of each order

purchaseOrder.displayDetails()

salesOrder.displayDetails()

}

1. **Create abstract class Shape with abstract Functions volume () and display (). Extend two classes Cube and Cylinder from it. Calculate volume of each and display it**

// Define the abstract class Shape

abstract class Shape {

// Abstract methods

def volume(): Double

def display(): Unit

}

// Define the Cube class extending Shape

class Cube(side: Double) extends Shape {

// Implement the volume method

override def volume(): Double = {

// function in Scala (from the scala.math package)

math.pow(side, 3) // Volume of a cube = side^3

}

// Implement the display method

override def display(): Unit = {

println(s"Cube with side length $side has volume: ${volume()}")

}

}

// Define the Cylinder class extending Shape

class Cylinder(radius: Double, height: Double) extends Shape {

// Implement the volume method

override def volume(): Double = {

math.Pi \* math.pow(radius, 2) \* height // Volume of a cylinder = π \* r^2 \* h

}

// Implement the display method

override def display(): Unit = {

println(s"Cylinder with radius $radius and height $height has volume: ${volume()}")

}

}

// Main object to test the functionality

object Main extends App {

// Create a Cube object and display its volume

val cube = new Cube(3)

cube.display()

// Create a Cylinder object and display its volume

val cylinder = new Cylinder(2, 5)

cylinder.display()

}