main.scala

```scala
object OrganizationApp extends App {  new *

  val organizationTree = Department("Organization", List(
    Department("Finance", List(
      Department("Payments", List(Employee(1, "Ravi", "Chennai"), Employee(2, "Anand", "Sahebganj")))
    )),
    Department("Sales", List(
      Department("Marketing", List(Employee(3, "Rohan", "Kolkata"), Employee(4, "Rakesh", "Mumbai"))),
      Employee(5, "Ravi", "Mumbai"),
      Department("Advertisement", List(
        Employee(6, "Ricky", "Chennai")
      )),
      Department("SalesManagement", List())
    )
  ));
```

11 2

Run OrganizationApp

```
/Users/abhishekanand/Library/Java/JavaVirtualMachines/openjdk-22.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=58244:/Applications/IntelliJ IDEA CE.app/C
Organization
      └─Finance
         └─Payments
            │  ├─ (1, Ravi, Chennai)
            │  ├─ (2, Anand, Sahebganj)
      └─Sales
         └─Marketing
            │  ├─ (3, Rohan, Kolkata)
            │  ├─ (4, Rakesh, Mumbai)
         │  ├─ (5, Ravi, Mumbai)
         └─Advertisement
            │  ├─ (6, Ricky, Chennai)
         └─SalesManagement
------ ORGANIZATIONAL CHART ------
Enter the employee details (sno., name, city) or type 'exit' to quit:
```

main.scala

```scala
object OrganizationApp extends App {  new *

  val organizationTree = Department("Organization", List(
    Department("Finance", List(
      Department("Payments", List(Employee(1, "Ravi", "Chennai"), Employee(2, "Anand", "Sahebganj")))
    )),
    Department("Sales", List(
      Department("Marketing", List(Employee(3, "Rohan", "Kolkata"), Employee(4, "Rakesh", "Mumbai"))),
      Employee(5, "Ravi", "Mumbai"),
      Department("Advertisement", List(
        Employee(6, "Ricky", "Chennai")
      )),
      Department("SalesManagement", List())
    ))
  ));
  printOrganizationChart(organizationTree, "    ")
  println("------ ORGANIZATIONAL CHART ------")
  var breakloop: Boolean = false;
  while (!breakloop) {
    print("Enter the employee details (sno., name, city) or type 'exit' to quit: ")
    val inputString = scala.io.StdIn.readLine();
    if (inputString == "" || inputString == "exit") {
      printOrganizationChart(organizationTree, "    ")
      breakloop = true;
    } else {
      println("input string " + inputString)
      val Array(serialNo, name, city, department) = inputString.split(",").map(_.trim)
      println(serialNo + "serial no")
      println(name + "name")
      println(city + "city")
```

Run OrganizationApp

```
    └──Sales
       └──Marketing
          │  ├── (3, Rohan, Kolkata)
          │  ├── (4, Rakesh, Mumbai)
       │  ├── (5, Ravi, Mumbai)
       └──Advertisement
          │  ├── (6, Ricky, Chennai)
       └──SalesManagement
------ ORGANIZATIONAL CHART ------
```

main.scala

```scala
object OrganizationApp extends App {  new *
        val employee = Employee(serialNo.toInt, name, city);
        addRecord(organizationTree, department, employee)
      }
    }
}

case class Employee (sno: Int, name: String, city: String) extends Organization {  new *
  override def toString: String = {  new *
    val str: String = s"($sno, $name, $city)"
    str
  }
}

class Organization  new *

case class Department (name: String, children: List[Organization]) extends Organization {  new *
  override def toString: String = name;  new *
}

def printOrganizationChart(org: Organization, indent: String = "    "): Unit = {  new *

  org match {
    case Department(dept, children) => {
      if (dept == "Organization") println(dept) else println(indent + "└──" + dept)
      children.foreach(child => printOrganizationChart(child, indent + "    " ))
    }
    case Employee(sno, name, city) => println(s"$indent  ├── ($sno, $name, $city)")
  }
}
```

Run OrganizationApp

```
    └──Sales
       └──Marketing
          │  ├── (3, Rohan, Kolkata)
          │  ├── (4, Rakesh, Mumbai)
       │  ├── (5, Ravi, Mumbai)
       └──Advertisement
          │  ├── (6, Ricky, Chennai)
       └──SalesManagement
------ ORGANIZATIONAL CHART ------
```

main.scala

```scala
def printOrganizationChart(org: Organization, indent: String = "    "): Unit = {  new *

  org match {
    case Department(dept, children) => {
      if (dept == "Organization") println(dept) else println(indent + "└──" + dept)
      children.foreach(child => printOrganizationChart(child, indent + "    " ))
    }
    case Employee(sno, name, city) => println(s"$indent|   ├── ($sno, $name, $city)")
  }
}

def addRecord(org: Organization, department: String, employee: Employee): Organization = {  new *
  def addDepartment(dept: Department): Department = {
    println("adding a department " + dept.name);
    if (dept.name == department) {
      Department(dept.name, dept.children :+ employee)
    } else {
      Department(dept.name, dept.children.map {
        case n: Department => addDepartment(n)
        case l: Employee => l
      })
    }
  }

  org match {
    case dept: Department => addDepartment(dept)
    case _ => org
  }
}
```

Run OrganizationApp

```
    └──Sales
       └──Marketing
          |  ├── (3, Rohan, Kolkata)
          |  ├── (4, Rakesh, Mumbai)
       |  |  ├── (5, Ravi, Mumbai)
       └──Advertisement
          |  ├── (6, Ricky, Chennai)
       └──SalesManagement
------ ORGANIZATIONAL CHART ------
```