Q) **Group Employee by using Department and find highest salary of each department?**


```java
import java.util.Arrays;
import java.util.Comparator;
import java.util.List;
import java.util.Map;
import java.util.Optional;
import java.util.stream.Collectors;
/**
 * Group Employee by using Department and find highest salary of each
   department?
 * @author Sankar Karra
 *Output: {hr=Employee [id=103, name=kiran, dept=hr, salary=50000.0],
            it=Employee [id=102, name=ram, dept=it, salary=90000.0]}
 */
class Employee{
        Integer id;
        String name;
        String dept;
        Double salary;
        public Employee() {

        }
        public Employee(Integer id, String name, String dept, Double salary) {
                super();
                this.id = id;
                this.name = name;
                this.dept = dept;
                this.salary = salary;
        }
        public Integer getId() {
                return id;
        }
        public void setId(Integer id) {
                this.id = id;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public String getDept() {
                return dept;
        }
        public void setDept(String dept) {
```

```java
                this.dept = dept;
        }
        public Double getSalary() {
                return salary;
        }
        public void setSalary(Double salary) {
                this.salary = salary;
        }
        @Override
        public String toString() {
                return "Employee [id=" + id + ", name=" + name + ", dept=" + dept + ",
salary=" + salary + "]";
        }


}

public class Test{

        public static void main(String[] args) {
    List<Employee> list= Arrays.asList(new Employee(101, "sankar", "hr", 20000D),
                        new Employee(102, "ravi", "it", 30000D),
                        new Employee(102, "ram", "it", 70000D),
                        new Employee(102, "durga", "it", 90000D),
                        new Employee(103, "kiran", "hr", 40000D),
                        new Employee(104, "sam", "hr", 50000D)
                        );


    Map<String, Optional<Employee>> map
=list.stream().collect(Collectors.groupingBy(Employee::getDept,
                Collectors.maxBy(Comparator.comparingDouble(Employee::getSalary))));

    System.out.println(map);


        }
}
```