1. Find out the count of male and female employees present in the organization?

public static void getCountOfMaleFemale(List<Employee> employeeList) {  
 Map<String, Long> noOfMaleAndFemaleEmployees=  
 employeeList.stream()  
 .collect(Collectors.groupingBy  
 (Employee::getGender, Collectors.counting()));   
 System.out.println(noOfMaleAndFemaleEmployees);  
 }  
  
 Output: {Male=11, Female=6}

2. Write a program to print the names of all departments in the organization.

public static void getDepartmentName(List<Employee> employeeList){  
 employeeList.stream()  
 .map(Employee::getDepartment)  
 .distinct()  
 .forEach(System.out::println);  
 }  
  
  
 Output:   
 HR  
 Sales And Marketing  
 Infrastructure  
 Product Development  
 Security And Transport  
 Account And Finance

3. Find the average age of Male and Female Employees.

public static void getGender(List<Employee> employeeList) {  
 Map<String, Double> avgAge = employeeList.stream()  
 .collect(Collectors.groupingBy  
 (Employee::getGender,   
 Collectors.averagingInt  
 (Employee::getAge)));  
 System.out.println(avgAge);  
 }  
  
 Output: {Male=30.181818181818183, Female=27.166666666666668}

4. Get the Names of employees who joined after 2015.

public static void getNameOfEmp(List<Employee> employeeList) {  
 employeeList.stream()  
 .filter(e -> e.getYearOfJoining() > 2015)  
 .map(Employee::getName)  
 .forEach(System.out::println);  
 }  
  
 Output:   
 Iqbal Hussain  
 Amelia Zoe  
 Nitin Joshi  
 Nicolus Den  
 Ali Baig

5. Count the number of employees in each department.

public static void countByDept(List<Employee> employeeList) {  
 Map<String, Long> countByDept = employeeList.stream()  
 .collect(Collectors.groupingBy  
 (Employee::getDepartment,   
 Collectors.counting()));  
 Set<Entry<String, Long>> entrySet = countByDept.entrySet();  
 for (Entry<String, Long> entry : entrySet)  
 {  
 System.out.println(entry.getKey()+" : "+entry.getValue());  
 }  
 }  
  
 Output:  
 Product Development : 5  
 Security And Transport : 2  
 Sales And Marketing : 3  
 Infrastructure : 3  
 HR : 2  
 Account And Finance : 2

6. Find out the average salary of each department.

public static void avgSalary(List<Employee> employeeList) {  
 Map<String, Double> avgSalary = employeeList.stream()  
 .collect(Collectors.groupingBy  
 (Employee::getDepartment,   
 Collectors.averagingDouble(Employee::getSalary)));  
 Set<Entry<String, Double>> entrySet = avgSalary.entrySet();  
 for (Entry<String, Double> entry : entrySet)   
 {  
 System.out.println(entry.getKey()+" : "+entry.getValue());  
 }  
 }  
  
 Output:  
 Product Development : 31960.0  
 Security And Transport : 10750.25  
 Sales And Marketing : 11900.166666666666  
 Infrastructure : 15466.666666666666  
 HR : 23850.0  
 Account And Finance : 24150.0

7. Find out the oldest employee, his/her age and department?

public static void oldestEmp(List<Employee> employeeList) {  
 Optional<Employee> oldestEmp = employeeList.stream()  
 .max(Comparator  
 .comparingInt(Employee::getAge)); Employee oldestEmployee = oldestEmp .get();  
  
 System.out.println("Name : "+oldestEmployee.getName());   
 System.out.println("Age : "+oldestEmployee.getAge());   
 System.out.println("Department : "+oldestEmployee.getDepartment());  
 }  
  
 Output:  
 Name : Iqbal Hussain  
 Age : 43  
 Department : Security And Transport

8. Find out the average and total salary of the organization.

public static void getEmpSalary(List<Employee> employeeList) {  
 DoubleSummaryStatistics empSalary = employeeList.stream()  
 .collect(Collectors  
 .summarizingDouble(Employee::getSalary));  
  
 System.out.println("Average Salary = "+empSalary.getAverage());  
 System.out.println("Total Salary = "+empSalary.getSum());  
 }  
  
 Output:  
 Average Salary = 21141.235294117647  
 Total Salary = 359401.0

9. List down the employees of each department.

public static void listDownDept(List<Employee> employeeList) {  
 Map<String, List<Employee>> empList = employeeList.stream()  
 .collect(Collectors  
 .groupingBy(Employee::getDepartment));  
  
 Set<Entry<String, List<Employee>>> entrySet = empList.entrySet();  
  
 for (Entry<String, List<Employee>> entry : entrySet)   
 {  
 System.out.println("--------------------------------------");   
 System.out.println("Employees In "+entry.getKey() + " : ");  
 System.out.println("--------------------------------------");  
  
 List<Employee> list = entry.getValue();  
 for (Employee e : list)   
 {  
 System.out.println(e.getName());  
 }  
 }  
 }  
  
 Output:  
 Employees In Product Development :  
 ————————————–  
 Murali Gowda  
 Wang Liu  
 Nitin Joshi  
 Sanvi Pandey  
 Anuj Chettiar  
 ————————————–  
 Employees In Security And Transport :  
 ————————————–  
 Iqbal Hussain  
 Jaden Dough  
 ————————————–  
 Employees In Sales And Marketing :  
 ————————————–  
 Paul Niksui  
 Amelia Zoe  
 Nicolus Den  
 ————————————–  
 Employees In Infrastructure :  
 ————————————–  
 Martin Theron  
 Jasna Kaur  
 Ali Baig  
 ————————————–  
 Employees In HR :  
 ————————————–  
 Jiya Brein  
 Nima Roy  
 ————————————–  
 Employees In Account And Finance :  
 ————————————–  
 Manu Sharma  
 Jyothi Reddy

10. Find out the height of experienced employees in the organization

public static void seniorEmp(List<Employee> employeeList) {  
 Optional<Employee> seniorEmp = employeeList.stream()  
 .sorted(Comparator  
 .comparingInt(Employee::getYearOfJoining)).findFirst();  
  
 Employee seniorMostEmployee = seniorEmp.get();  
  
 System.out.println("Senior Most Employee Details :");  
 System.out.println("----------------------------");  
 System.out.println("ID : "+seniorMostEmployee.getId());  
 System.out.println("Name : "+seniorMostEmployee.getName());  
 System.out.println("Age : "+seniorMostEmployee.getAge());  
 }  
  
Output:  
Senior Most Employee Details :  
—————————-  
ID : 177  
Name : Manu Sharma  
Age : 35