

[Dart Programming Language] - Lecture [10]

[Exercises part 1]

1. Introduction

Q1. Write a dart code that extracts books that belong to a specific type (action, fantasy, crime, ...) and change the title of the books to uppercase.

```
class Book {  
  String title;  
  String author;  
  int year;  
  List<String> genres;  
  Book(this.title, this.author, this.year, this.genres);  
  @override  
  String toString() => '$title by $author ($year) - Genres: $genres';  
}  
  
void main() {  
  List<Book> library = [  
    Book('1984', 'George Orwell', 1949, ['Action', 'Science Fiction']),  
    Book('To Kill a Mockingbird', 'Harper Lee', 1960, ['Classic', 'Historical']),  
    Book('Brave New World', 'Aldous Huxley', 1932, ['Classic', 'Action']),  
  ];  
  
  List<Book> actionBooks = library.where((book) =>  
    book.genres.contains('Action')).toList();  
  
  List<String> transformedTitles = library.map((book) =>  
    book.title.toUpperCase()).toList();  
}
```

Q2. Write a dart code that describes an employee, then create a list of employees. Then make it possible to change their salary as we want. Change the department of an employee as we want. Extract employee with a specific department.

```
class Employee {
  String name;
  int id;
  double salary;
  String department;
  Employee(this.name, this.id, this.salary, this.department);
  @override
  String toString() =>
    'ID: $id , Name: $name , Salary: \${salary.toStringAsFixed(2)} , Department: $department \n \n';
}

List<Employee> calculateNewSalary(List<Employee> employees) {
  return employees.map((employee) {
    return Employee(
      employee.name, employee.id, employee.salary * 1.1, employee.department);
  }).toList();
}

List<Employee> employeesWithSpecificDepartment(
  List<Employee> employees, String targetDepartment) {
  return employees.where((employee) {
    return employee.department == targetDepartment;
  }).toList();
}
```

```

List<Employee> employeesWithSpecificDepartment(
    List<Employee> employees, String targetDepartment) {
    return employees.where((employee) {
        return employee.department == targetDepartment;
    }).toList();
}

void main() {
    List<Employee> employees = [
        Employee('Ahmed Ali', 101, 50000, 'HR'),
        Employee('Saeed Saleh', 102, 60000, 'IT'),
        Employee('Adel Emam', 103, 55000, 'IT'),
        Employee('Mona Ali', 104, 48000, 'Finance'),
    ];

    List<Employee> itDepartment =
        employeesWithSpecificDepartment(employees, 'IT');

    List<Employee> salaryIncrease = calculateNewSalary(employees);

    print('this is the original list of employees \n $employees \n \n');
    employees = changeEmployeeDepartment(employees, 'SS', employees[2]);
    print('list of IT employees \n $itDepartment \n \n');
    print('list of employees with their new salaries \n $salaryIncrease \n \n');
    print('the list of updated employees \n $employees \n \n');
}

```

The output of the code will be as follows:

```
this is the original list of employees

[ID: 101 , Name: Ahmed Ali , Salary: $50000.00 , Department: HR
, ID: 102 , Name: Saeed Saleh , Salary: $60000.00 , Department: IT
, ID: 103 , Name: Adel Emam , Salary: $55000.00 , Department: IT
, ID: 104 , Name: Mona Ali , Salary: $48000.00 , Department: Finance
]

list of IT employees

[ID: 102 , Name: Saeed Saleh , Salary: $60000.00 , Department: IT
, ID: 103 , Name: Adel Emam , Salary: $55000.00 , Department: IT
]

list of employees with their new salaries

[ID: 101 , Name: Ahmed Ali , Salary: $55000.00 , Department: HR
, ID: 102 , Name: Saeed Saleh , Salary: $66000.00 , Department: IT
, ID: 103 , Name: Adel Emam , Salary: $60500.00 , Department: IT
, ID: 104 , Name: Mona Ali , Salary: $52800.00 , Department: Finance
]

the list of updated employees

[ID: 101 , Name: Ahmed Ali , Salary: $50000.00 , Department: HR
, ID: 102 , Name: Saeed Saleh , Salary: $60000.00 , Department: IT
, ID: 103 , Name: Adel Emam , Salary: $55000.00 , Department: SS
, ID: 104 , Name: Mona Ali , Salary: $48000.00 , Department: Finance
]
```