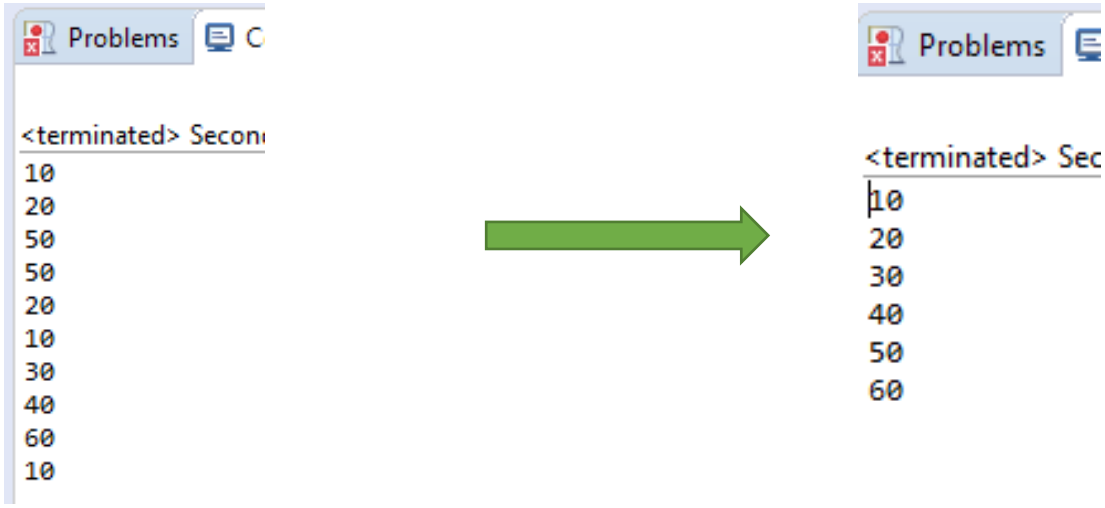
BSc (Hons) in Information Technology
Year 02

Lab Exercise – Collections and Generics

IT2030 - Object Oriented Programming (OOP) **Semester 1, 2018**

Exercise 01

Implement the correct Collection to **remove duplicates** and **store values in ascending order** of the below program.

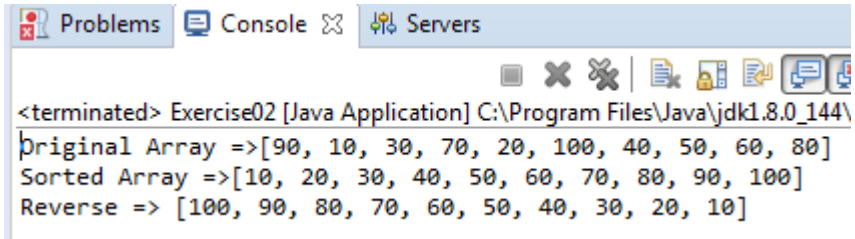


Exercise 02

Convert the below Integer array into **Collection ArrayList<Integer>** and **sort** it to the **reverse order.**

```
Integer [] array = new Integer[]{90, 10, 30, 70, 20, 100, 40, 50, 60, 80};
```

Your output should be as follows.



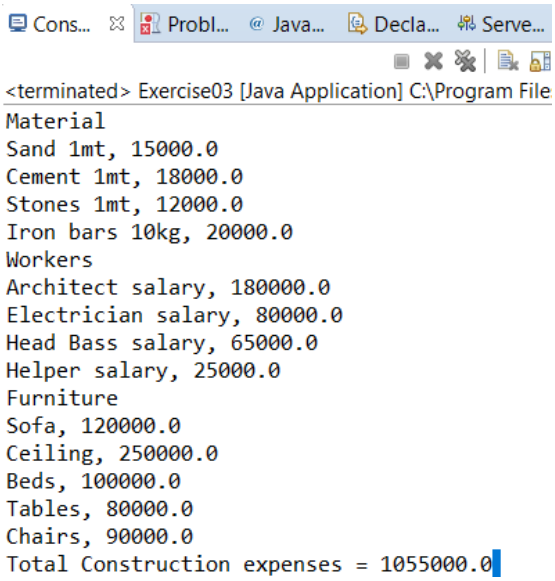
Exercise 03

Implement the **Expenses.jave** class and implement the `Map<String, LinkedHashMap<String, Double>> getConstructionExpenses()` method to construct the map and return the map as follows.

```
public static Map<String, LinkedHashMap<String, Double>> getConstructionExpenses(){

    LinkedHashMap<String, Double> materialExpensesMap = new LinkedHashMap<String, Double>();
    materialExpensesMap.put("Sand 1mt", 15000.0);
    materialExpensesMap.put("Cement 1mt", 18000.0);
    materialExpensesMap.put("Stones 1mt", 12000.0);
    materialExpensesMap.put("Iron bars 10kg", 20000.0);

    LinkedHashMap<String, Double> workerExpensesMap = new LinkedHashMap<String, Double>();
    workerExpensesMap.put("Architect salary", 180000.0);
    workerExpensesMap.put("Electrician salary", 80000.0);
    workerExpensesMap.put("Head Bass salary", 65000.0);
    workerExpensesMap.put("Helper salary", 25000.0);

    LinkedHashMap<String, Double> furnitureExpenses = new LinkedHashMap<String, Double>();
    furnitureExpenses.put("Sofa", 120000.0);
    furnitureExpenses.put("Ceiling", 250000.0);
    furnitureExpenses.put("Beds", 100000.0);
    furnitureExpenses.put("Tables", 80000.0);
    furnitureExpenses.put("Chairs", 90000.0);

    Map<String, LinkedHashMap<String, Double>> constructionExpenses =
            new LinkedHashMap<String, LinkedHashMap<String, Double>>();
    constructionExpenses.put("Material", materialExpensesMap);
    constructionExpenses.put("Workers", workerExpensesMap);
    constructionExpenses.put("Furniture", furnitureExpenses);

    return constructionExpenses;
}
```

Exercise 04

Now implement a method to display all above expenses and **print the Total expenses** as well to get the below output.



Exercise 05

Now convert the below method into **Generic Method**

```
public static void printMap(Map<String, LinkedHashMap<String, Double>> expensesMap){

      for (Map.Entry<String, LinkedHashMap<String, Double>> expenses : expensesMap.entrySet()) {
          System.out.println(expenses.getKey());
          for (Map.Entry<String, Double> expense : expenses.getValue().entrySet()) {
              System.out.println(expense.getKey() + ", " + expense.getValue());
          }
      }
  }
```

Exercise 06

Now implement following Subjects.java class and implement the following **public static Map<String, LinkedHashMap<String, Integer>> getStudentMarks()** method.

```
public static Map<String, LinkedHashMap<String, Integer>> getStudentMarks() {

    LinkedHashMap<String, Integer> oopMarksMap = new LinkedHashMap<String, Integer>();
    oopMarksMap.put("IT1011233", 80);
    oopMarksMap.put("IT1011244", 85);
    oopMarksMap.put("IT1011255", 75);
    oopMarksMap.put("IT1011266", 65);
    oopMarksMap.put("IT1011277", 55);

    LinkedHashMap<String, Integer> seMarksMap = new LinkedHashMap<String, Integer>();
    seMarksMap.put("IT1011233", 80);
    seMarksMap.put("IT1011244", 85);
    seMarksMap.put("IT1011255", 75);
    seMarksMap.put("IT1011266", 65);
    seMarksMap.put("IT1011277", 55);

    LinkedHashMap<String, Integer> dbmsMarksMap = new LinkedHashMap<String, Integer>();
    dbmsMarksMap.put("IT1011233", 80);
    dbmsMarksMap.put("IT1011244", 85);
    dbmsMarksMap.put("IT1011255", 75);
    dbmsMarksMap.put("IT1011266", 65);
    dbmsMarksMap.put("IT1011277", 55);

    Map<String, LinkedHashMap<String, Integer>> subjectsMap =
            new LinkedHashMap<String, LinkedHashMap<String, Integer>>();
    subjectsMap.put("OOP", oopMarksMap);
    subjectsMap.put("SE", seMarksMap);
    subjectsMap.put("DBMS", dbmsMarksMap);

    return subjectsMap;
}
```

Now refer the implemented **Generic Method** in **Exercise 05** to insert the following map as well. The Generic method should support both maps and print responses as expected.

`Map<String, LinkedHashMap<String, Integer>>`

`Map<String, LinkedHashMap<String, Double>>`

```
public static void main(String[] args) {

    genericMapDisplay(Expenses.getConstructionExpenses());
    genericMapDisplay(Subjects.getStudentMarks());
}
```

Output should be as follows when you run the **Exercise 06**

```
<terminated> Exercise05 [Java Application]
Material
Sand 1mt, 15000.0
Cement 1mt, 18000.0
Stones 1mt, 12000.0
Iron bars 10kg, 20000.0
Workers
Architect salary, 180000.0
Electrician salary, 80000.0
Head Bass salary, 65000.0
Helper salary, 25000.0
Furniture
Sofa, 120000.0
Ceiling, 250000.0
Beds, 100000.0
Tables, 80000.0
Chairs, 90000.0
OOP
IT1011233, 80
IT1011244, 85
IT1011255, 75
IT1011266, 65
IT1011277, 55
SE
IT1011233, 80
IT1011244, 85
IT1011255, 75
IT1011266, 65
IT1011277, 55
DBMS
IT1011233, 80
IT1011244, 85
IT1011255, 75
IT1011266, 65
IT1011277, 55
```

Exercise 07

Now modify the above **Exercise 06** with including **Generic class** as follows to display the same output in **Exercise 06**

```
class GenericMap<K, V> {

        ≪Implement the rest of the code≫

}
```

----------The End of the Lab----------