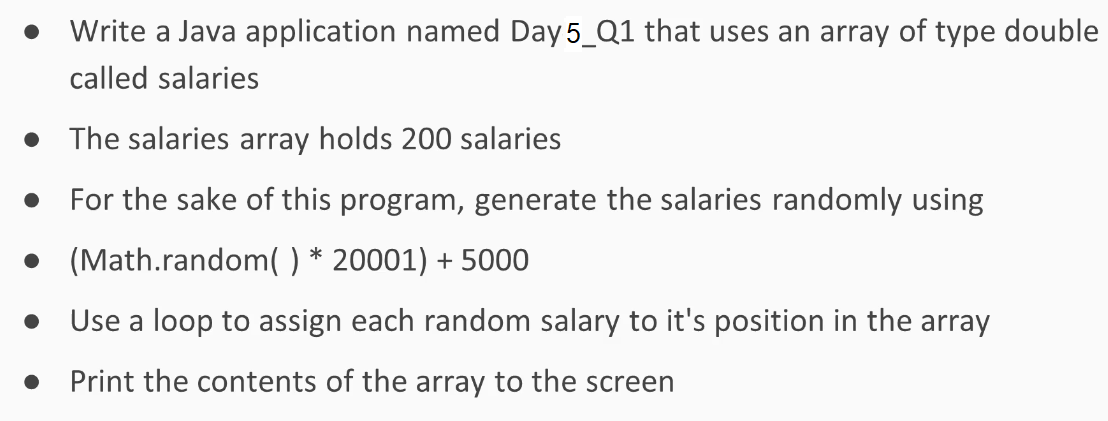
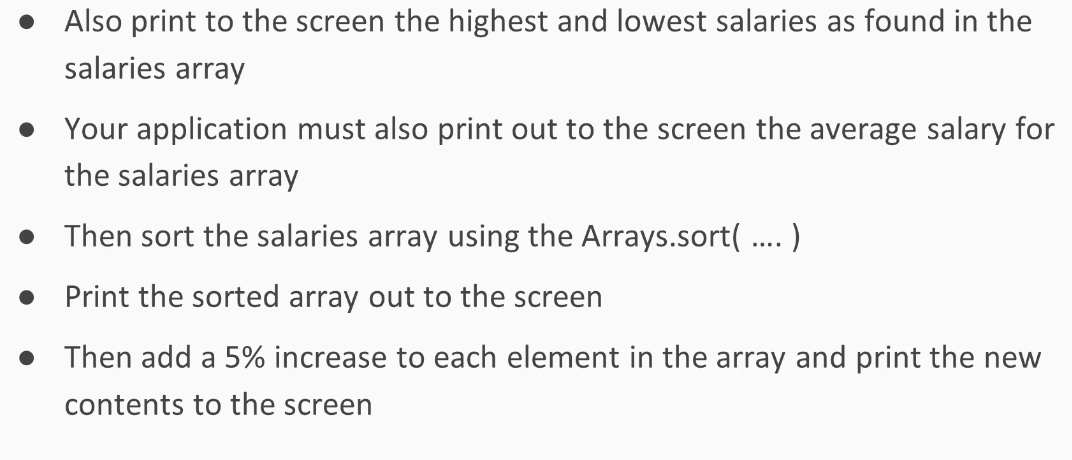
Hands On Exercises - Collections

**Setup Instructions:**

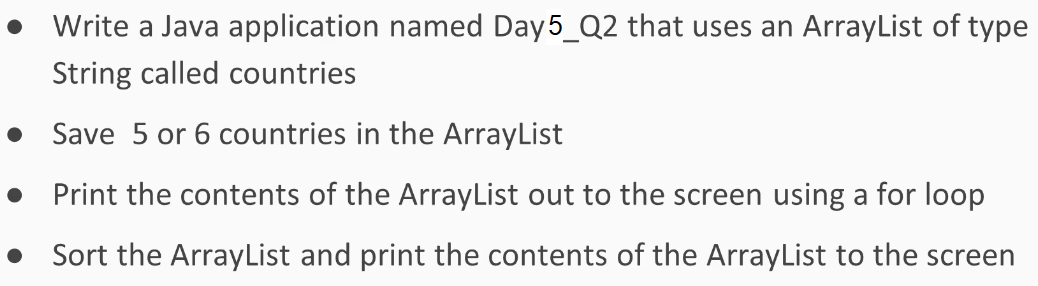
1. Create a new Java Project called “FullStackDay5”.
2. Create your classes in the package “collections”.
3. Define your classes as given below for each assignment.

Assignment 01

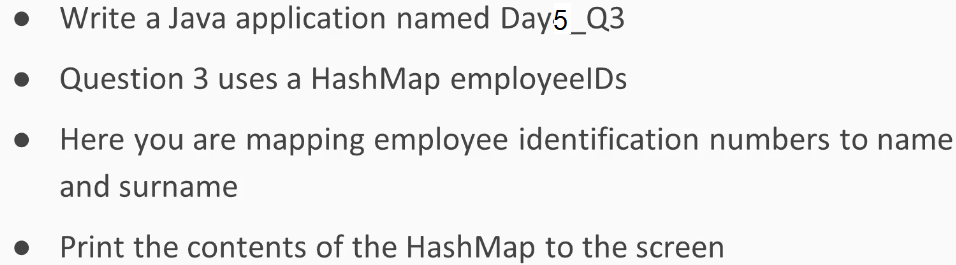


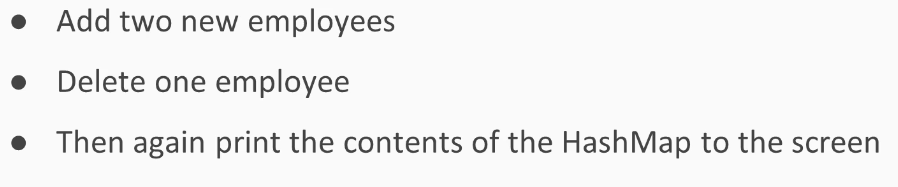


Assignment 02



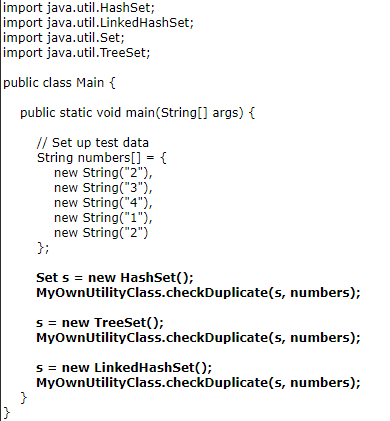
Assignment 03



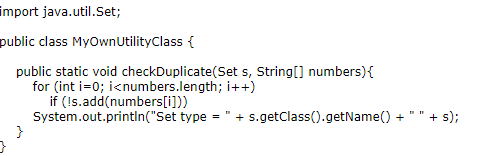


Assignment 04

* Write a Java application as follows.
* “setpolymorphism” is the sub-package name under package “collections”.
* Create a class ”Main” with a main method as follows:



* Write MyOwnUtilityClass.java.



* Build and run the project

**Output:**



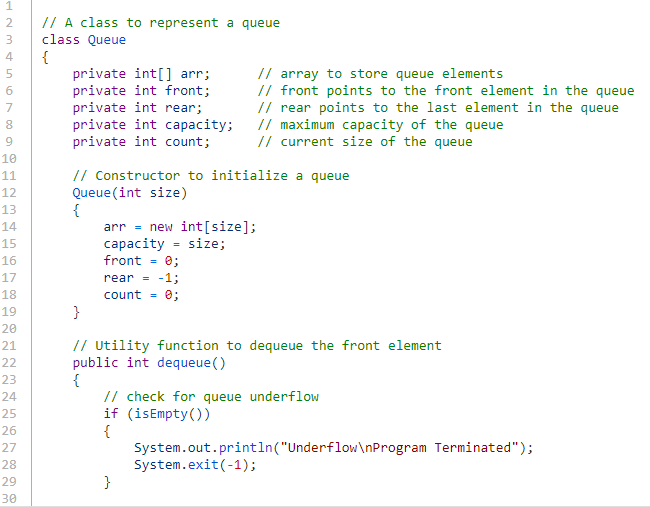
Assignment 05

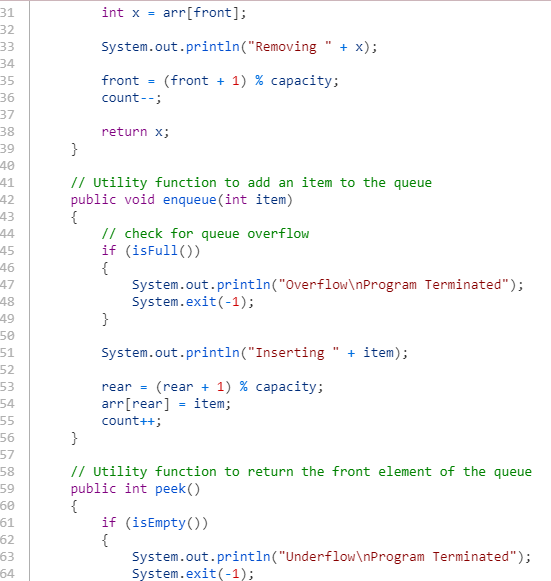
The next two assignments cover queue implementation in Java. A queue is a linear data structure that follows the FIFO (First–In, First–Out) principle. That means the object inserted first will be the first one out, followed by the object inserted next.

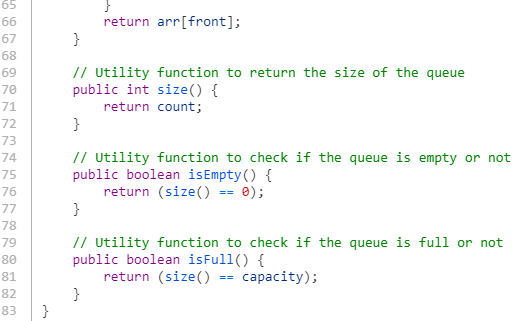
The queue supports the following core operations:

1. Enqueue: Inserts an item at the rear of the queue.
2. Dequeue: Removes the object from the front of the queue and returns it, thereby decrementing queue size by one.
3. Peek: Returns the object at the front of the queue without removing it.
4. IsEmpty: Tests if the queue is empty or not.
5. Size: Returns the total number of elements present in the queue.

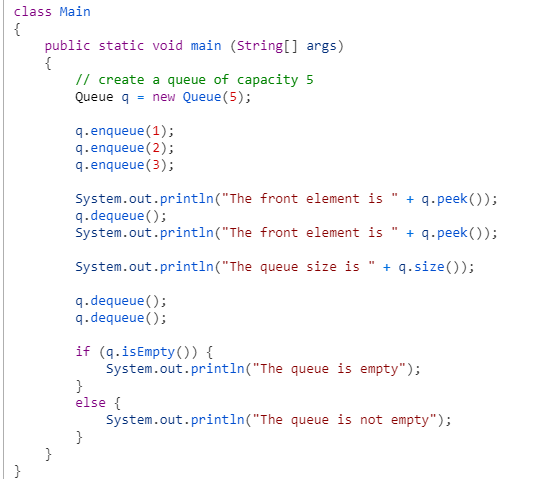
* Write a Java application named Queue.
* “queuewitharray” is the sub-package name under package “collections”.
* Create the following class :

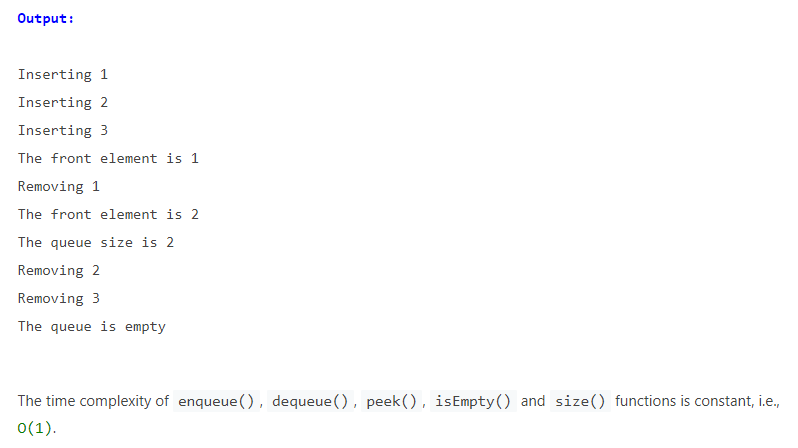






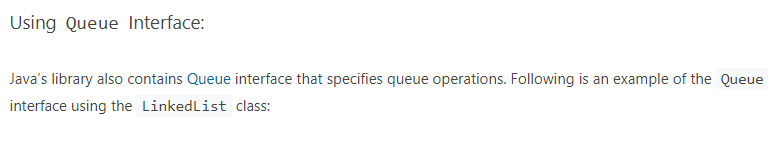
* Create a class ”Main” with a main method as follows:



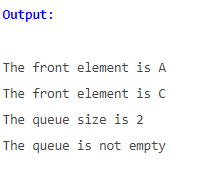


Assignment 06

* Write a Java application as follows.
* “queuewithlinkedlist” is the sub-package name under package “collections”.
* Create a class ”Main” with a main method as follows:







Assignment 07

* Do the Homework Assignment on Slide # 8 of *“5. Collection for Insertions - Queues”* presentation.

Assignment 08

\*\*\* This is a doSelect Test Problem.

##### **Mobile Shop**

 Coding

##### Description

Your task here is to implement ****Java**** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider ****default visibility**** of classes, data fields, and methods unless mentioned.

## Specifications

## 

## Task

****Class**** ****Mobile****

****-define the object of HashMap<String, ArrayList<String>>**** with variable name ****mobiles****.

* The ****String****defines the ****name of the company****and the ****Arraylist<String>****will have list of models.

****Implement the below methods for this class:****

****-String addMobile(String company, String model):****

* Write a code to add ****a company**** and its ****model****in ****mobiles****map as given below
* If the ****company**** does not exist in the map already, add the ****company**** and its ****model**** into the map. (Note: Add ****model**** into a new ****ArrayList<String>****and add this list into map as value)
* If the ****company**** already exist in the map, append the given ****model**** into the corresponding model list.
* Return "****model successfully added****" after performing the above operations

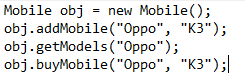
****-ArrayList<String> getModel(String company):****

* Write a code to get the Model list for the given company from Map ****mobiles****.
* Return ****null**** if the given company doesn't exist or doesn't have any model, else return the****List<String>**** of all the models.

****-String buyMobile(String company, String model):****

* Write a code to buy a mobile.
* Remove the mobile ****model**** from the list according to the ****compnay**** and ****model**** given. In case there are two same models then remove one and return the message "mobile sold successfully"
* Return a message "****item not available" i****f the ****company**** or corresponding ****model**** is not present in the Map

****Sample Input****



****Sample Output****



## NOTE:

* You can make suitable function calls and use ****RUN CODE**** button to check your ****main()**** method output.

##### Execution time limit

10 seconds

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

class Mobile{

// Write your code here..

}

public class Source {

public static void main(String args[] ) throws Exception {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

}

}

Assignment 09

\*\*\* This is a doSelect Test Problem.

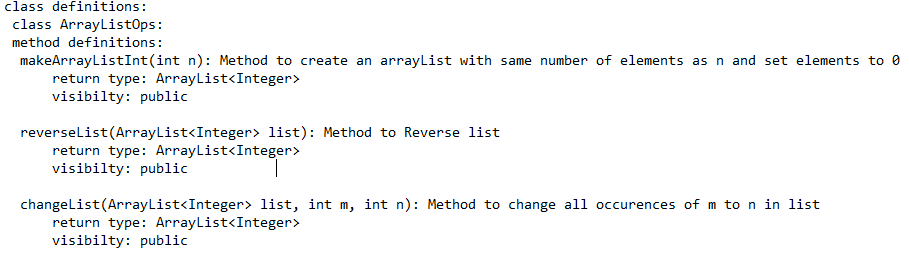
##### **List of Operations**

 Coding

##### Description

Your task here is to implement a ****Java**** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.

****Specifications:****



****Task:****

Your task is to create a class ****ArrayListOps**** and implement the following:

****1.**** ****makeArrayListInt(int n):****Method to create an A****rrayList**** with number of elements as n and **set**elements to ****0****.

* If number of elements ****n**** is 4 , then the method should return a list containing ****[0,0,0,0]****

****2.**** ****reverseList(ArrayList<Integer> list):**** Method to **Reverse**list****

****3. changeList(ArrayList<Integer> list, int m, int n)****: Method to change all ****occurences**** of ****m**** to ****n**** in ****list****

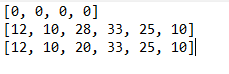
****Important:****

* To check your program, you can use the ****main()**** method (in Source class) given in the stub. You can make suitable function calls and use ****RUN CODE**** button to check your main() function output.

****Sample Input****



****Sample Output****



****NOTE:****

* The above ****Sample Output**** is only for demonstration purposes and will be obtained if you implement the ****main()**** method with all method calls accordingly.
* Upon implementation of ****main()**** method, you can use the ****RUN CODE**** button to pass input data in the method calls and arrive at the ****Sample Output****.

##### Execution time limit

10 seconds

import java.util.\*;

class ArrayListOps {

}

public class Source{

public static void main(String[] args) {

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*