Java Assessment

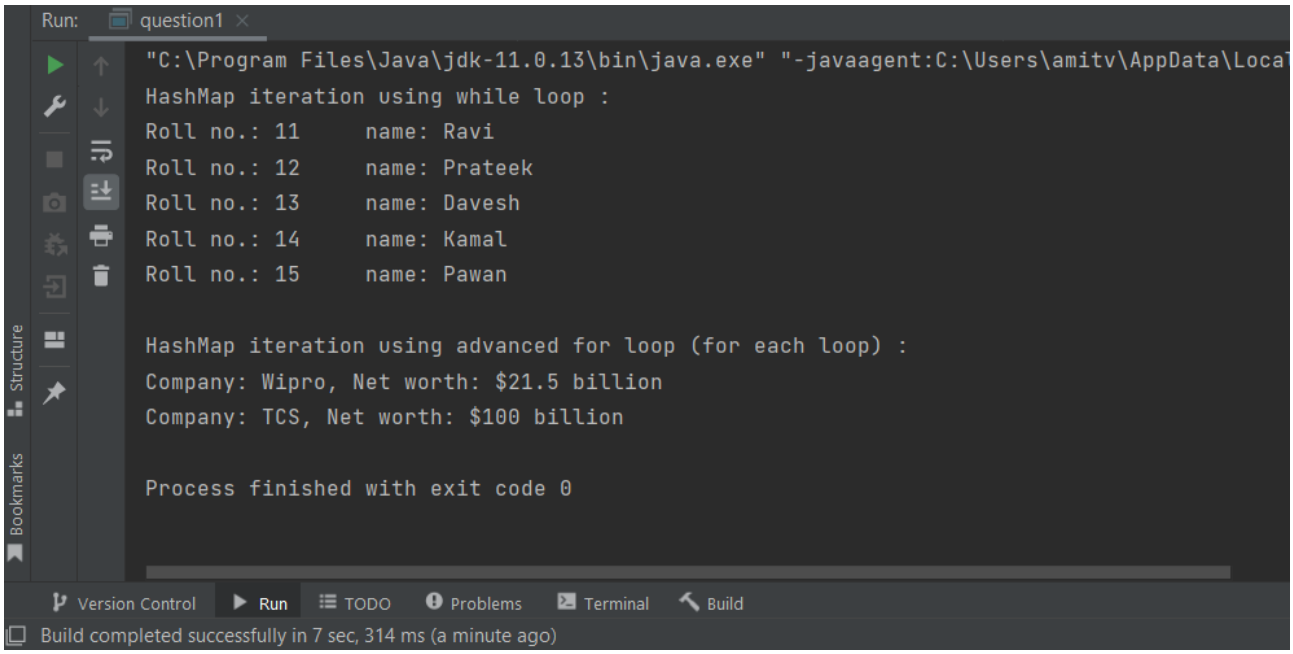
1. Write a Java Program to iterate HashMap using While and advance for loop.

Code : -

```
// Write a Java Program to iterate HashMap using While and advance for loop.

package com.company;
import java.util.Iterator;
import java.util.HashMap;

class HashMapIteration{

    private HashMap<Integer, String> hm = new HashMap<Integer, String>();
    private HashMap<String,String> map = new HashMap<String,String>();

    public void usingWhileLoop(){
        hm.put(11,"Ravi");
        hm.put(12,"Prateek");
        hm.put(13, "Davesh");
        hm.put(14, "Kamal");
        hm.put(15, "Pawan");
        Iterator <Integer> it = hm.keySet().iterator();       //keyset is a method
        while(it.hasNext())
        {
            int key=(int)it.next();
            System.out.println("Roll no.: "+key+"     name: "+hm.get(key));
        }
    }

    public void usingAdvanceForLoop(){
        map.put("TCS","$100 billion");
        map.put("Wipro","$21.5 billion");
        //iteration over map using forEach() method
        map.forEach((k,v) -> System.out.println("Company: " + k + ", Net worth: "+
v));
    }
}

public class question1 {
```

```
    public static void main(String[] arg)
    {
        HashMapIteration hmi = new HashMapIteration();
        System.out.println("HashMap iteration using while loop :");
        hmi.usingWhileLoop();
        System.out.println("\nHashMap iteration using advanced for loop (for each
loop) :");
        hmi.usingAdvanceForLoop();
    }
}
```

Output : -

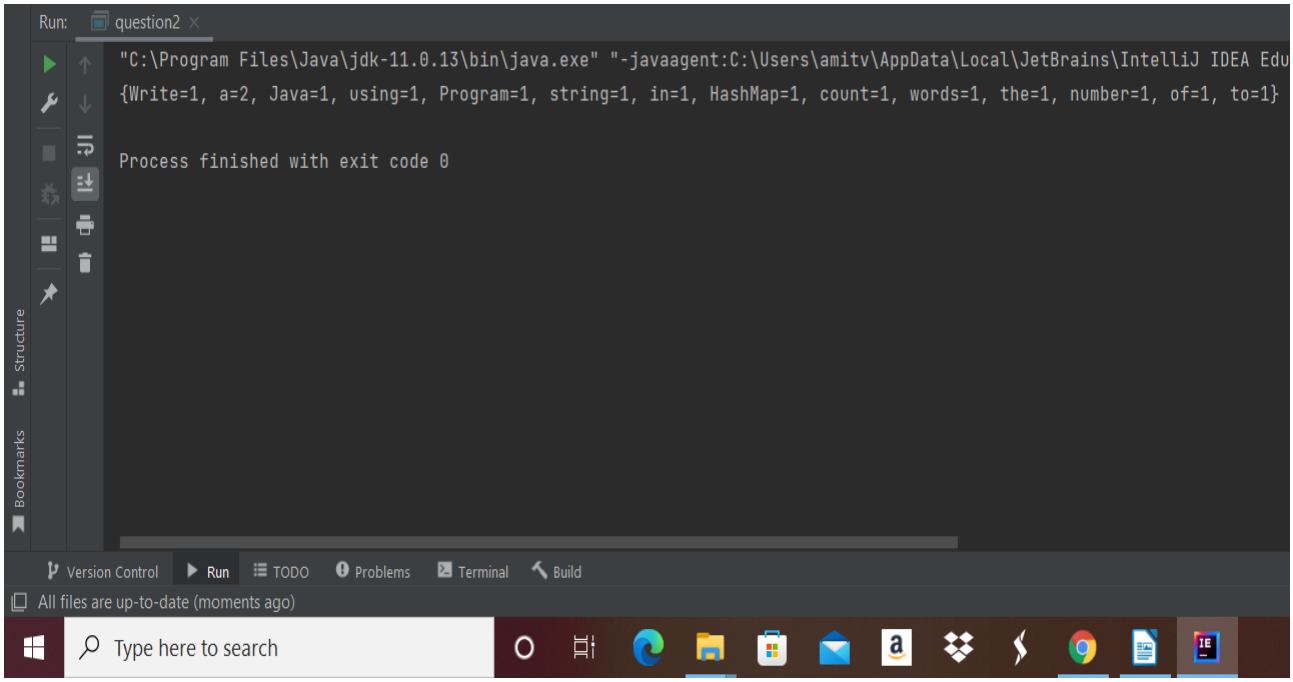


2. Write A Java Program to count the number of duplicate words in a string using HashMap

Code : -

```java
// Write A Java Program to count the number of duplicate words in a string using HashMap

package com.company;
import java.util.HashMap;

public class question2 {
    public static void main(String[] args) {
        String str = "Write a Java Program to count the number of words in a string using HashMap";
        HashMap<String, Integer> hashMap = new HashMap<>();
        String[] words = str.split(" ");

        for (String word : words) {

            // Asking whether the HashMap contains the key or not. Will return null if not.
            Integer integer = hashMap.get(word);
            if (integer == null)
                // Storing the word as key and its occurrence as value in the HashMap.
                hashMap.put(word, 1);
            else {
                // Incrementing the value if the word is already present in the HashMap.
                hashMap.put(word, integer + 1);
            }
        }
        System.out.println(hashMap);
    }
}
```

Output : -


Run:
question2
"C:\Program Files\Java\jdk-11.0.13\bin\java.exe" "-javaagent:C:\Users\amitv\AppData\Local\JetBrains\IntelliJ IDEA Edu
{Write=1, a=2, Java=1, using=1, Program=1, string=1, in=1, HashMap=1, count=1, words=1, the=1, number=1, of=1, to=1}
Process finished with exit code 0
Structure
Bookmarks
Version Control
Run
TODO
Problems
Terminal
Build
All files are up-to-date (moments ago)
Type here to search

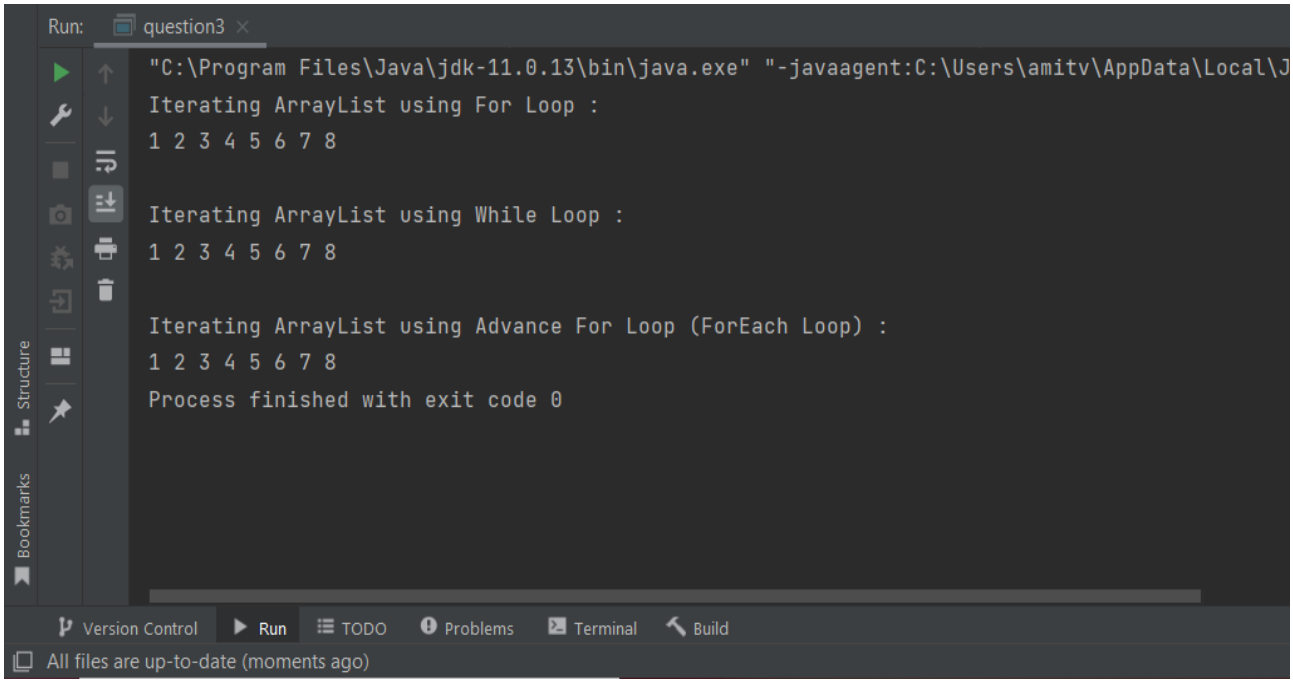
3. Write Java Program to iterate ArrayList using for-loop, while-loop, and advance for-loop.

Code : -

```
// Write Java Program to iterate ArrayList using for-loop, while-loop, and advance
for-loop.

package com.company;
import java.util.*;

class ArrayListIteration {

    List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8);

    public void usingForLoop() {
        System.out.println("Iterating ArrayList using For Loop :");
        for (int i = 0; i < numbers.size(); i++)
            // Printing and display the elements in ArrayList
            System.out.print(numbers.get(i) + " ");
    }

    public void usingWhileLoop() {
        System.out.println("\n\nIterating ArrayList using While Loop :");
        int val = 0;
        while (numbers.size() > val) {
            System.out.print(numbers.get(val) + " ");
            val++ ;
        }
    }

    public void usingAdvanceForLoop() {
        System.out.println("\n\nIterating ArrayList using Advance For Loop (ForEach
Loop) :");
        for(int i:numbers) System.out.print(i+" ");
    }
}

public class question3 {
    public static void main(String[] args) {
        ArrayListIteration ali = new ArrayListIteration();
        ali.usingForLoop();
```

```
        ali.usingWhileLoop();
        ali.usingAdvanceForLoop();
    }
}
```

Output : -

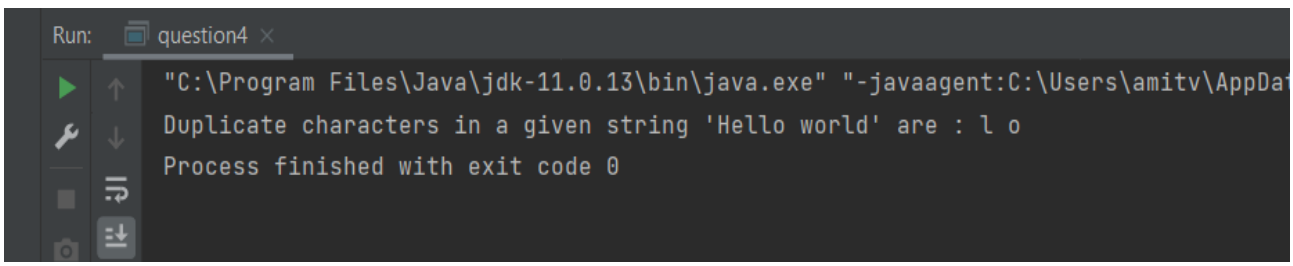


4. Write a Java Program to find the duplicate characters in a string.

Code : -

```java
// Write a Java Program to find the duplicate characters in a string.

package com.company;

public class question4 {
    public static void main(String[] args) {
        String str = "Hello world";
        int count;

        //Converts given string into character array
        char[] str1 = str.toCharArray();

        System.out.print("Duplicate characters in a given string '"+str+"' are : ");
        //Counts each character present in the string
        for(int i = 0; i <str1.length; i++) {
            count = 1;
            for(int j = i+1; j <str1.length; j++) {
                if(str1[i] == str1[j] && str1[i] != ' ') {
                    count++;
                    //Set string1[j] to 0 to avoid printing visited character
                    str1[j] = '0';
                }
            }
            //A character is considered as duplicate if count is greater than 1
            if(count > 1 && str1[i] != '0')
                System.out.print(str1[i]+" ");
        }
    }
}
```

Output : -

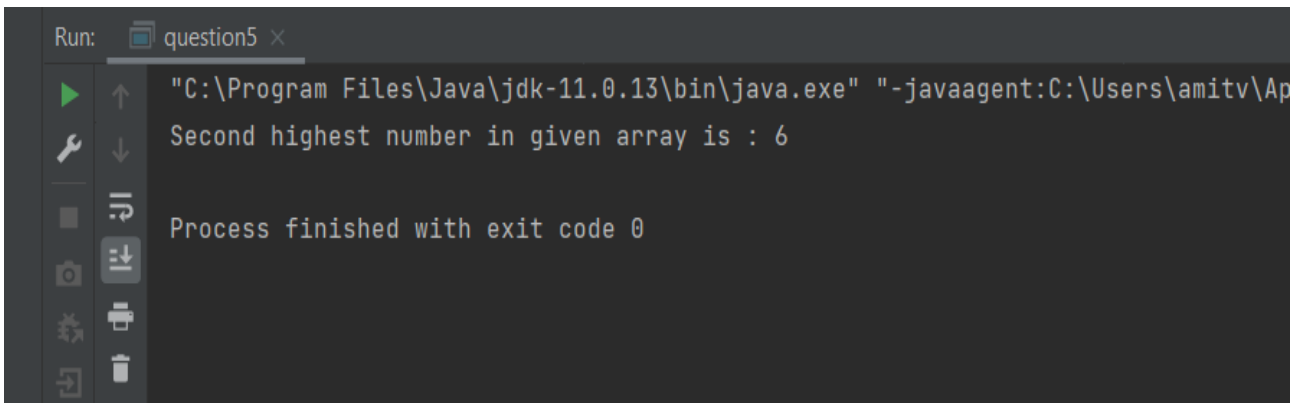


5. Write a Java Program to find the second-highest number in an array.

Code : -

```java
// Write a Java Program to find the second-highest number in an array.

package com.company;
import java.util.*;

public class question5 {
    public static int getSecondLargest(Integer[] x, int total){
        List<Integer> list=Arrays.asList(x);
        Collections.sort(list);
        int element=list.get(total-2);
        return element;
    }
    public static void main(String args[]){
        Integer a[]={8,6,5,2,0};
        System.out.println("Second highest number in given array is :
"+getSecondLargest(a,a.length));
    }
}
```

Output : -

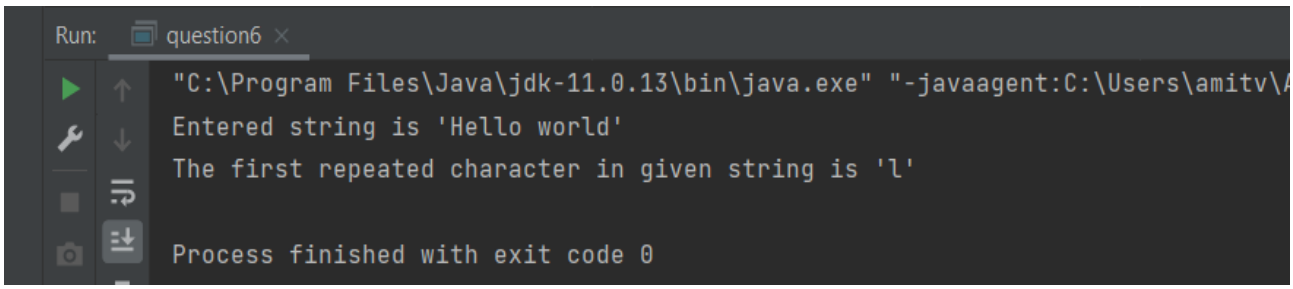


6. Given a String, find the first repeated character in it using Stream functions.

Code : -

```
// Given a String, find the first repeated character in it using Stream functions.

package com.company;

import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Optional;
import java.util.function.Function;
import java.util.stream.Collectors;

public class question6 {
    public static void main(String[] args) {
        String str = "Hello world";
        System.out.println("Entered string is '"+str+"'");
        Map<Character, Long> collect =  str.chars()
                .mapToObj(i -> (char)i)
                .collect(Collectors.groupingBy(Function.identity(), LinkedHashMap::new,
Collectors.counting()));

        Optional<Character> firstRepeat = collect.entrySet()
                .stream().
                filter( (e) -> e.getValue() > 1).
                map(e -> e.getKey()).findFirst();

        System.out.println("The first repeated character in given string is '" +
firstRepeat.orElse(null)+"'");
    }
}
```

Output : -



7. Given a String, find the first non-repeated character in it using Stream functions.

Code : -

```
// Given a String, find the first non-repeated character in it using Stream functions.

package com.company;

import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Optional;
import java.util.function.Function;
import java.util.stream.Collectors;

public class question7 {
    public static void main(String[] args) {
        String str = "Hello world";
        System.out.println("Entered string is '"+str+"'");
        Map<Character, Long> collect =  str.chars()
                .mapToObj(i -> (char)i)
                .collect(Collectors.groupingBy(Function.identity(), LinkedHashMap::new,
Collectors.counting()));

        Optional<Character> firstNonRepeat = collect.entrySet()
                .stream()
                .filter( (e) -> e.getValue() == 1)
                .map(e -> e.getKey()).findFirst();

        System.out.println("The first non-repeated character in given string is '" +
firstNonRepeat.orElse(null)+"'");
    }
}
```

Output : -