

# Day 10 Assignment

## Part 2

1. Create a program that declares an array of integers, initializes it with consecutive numbers, and prints the array in reverse order.

```
public class Part_2_Assignment_1 {  
    public static void main(String[] args) {  
        int n = 10;  
        int[] arr = new int[n];  
  
        for (int i = 0; i < n; i++) {  
            arr[i] = i + 1;  
        }  
  
        //Printing in Reverse using loop  
        for (int i = n - 1; i >= 0; i--) {  
            System.out.println(arr[i]);  
        }  
    }  
}
```

## Output

```
C:\Users\coolr\.jdk\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBr  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
  
Process finished with exit code 0
```

---

**2. Implement a method that takes a List as an argument and removes every second element from the list, then prints the resulting list.**

```
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;

/*
    Implement a method that takes a List as an argument and removes every second
    element from the list,
    then prints the resulting list.
*/
public class Part_2_Assignment_2 {

    static List<Integer> removeEveryElement(List<Integer> ls) {
        List<Integer> li = new LinkedList<Integer>();

        for (int i = 0; i < ls.size(); i++) {
            if (i % 2 == 1) {
                continue;
            }
            li.add(ls.get(i));
        }

        return li;
    }

    public static void main(String[] args) {
        List<Integer> ls = new LinkedList<Integer>();
        ls = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
        System.out.println("Before function call");
        System.out.println(ls);
        ls = Part_2_Assignment_2.removeEveryElement(ls);
        System.out.println("After function call");
        System.out.println(ls);
    }
}
```

**Output**

```
C:\Users\coolr\.jdk\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Users\coolr\.jdk\openjdk-22.0.1\bin\javaagent.jar"
Before function call
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
After function call
[1, 3, 5, 7, 9]

Process finished with exit code 0
```

- 
3. Write a program that reads words from a String variable into a Set and prints out the number of unique words, demonstrating the unique property of sets.

```
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

/*
 * Write a program that reads words from a String variable into a Set
 * and prints out the number of unique words, demonstrating the unique property
 * of sets.
 */
public class Part_2_Assignment_3 {

    public static void main(String[] args) {
        System.out.println("Enter a sentence, avoid putting any punctuations.");
        String s = new Scanner(System.in).nextLine();

        Set<String> set = new HashSet<>();

        for (String x : s.split(" ")) {
            set.add(x);
        }

        System.out.println("Unique words in the string are: ");
        for (String x : set) {
            System.out.println(x);
        }
    }
}
```

```
}
```

## Output

```
Enter a sentence, avoid putting any punctuations.  
My Name is Sayan and I have a Pet dog named Teddy I love programming and Data Structure and Algorithms This Program will find all those unqiue words  
Unique words in the string are:  
love  
named  
Sayan  
Teddy  
I  
My  
Name  
and  
find  
have  
This  
Structure  
programming  
those  
all  
a  
Program  
will  
words  
is  
Data  
unqiue  
Algorithms  
dog  
Pet
```

### 4. Create a Java class that uses a Map to store the frequency of each word that appears in a given string.

```
import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
import java.util.Set;  
  
class WordCounter {  
    private Map<String, Integer> mp;  
  
    WordCounter(String string) {  
        this.mp = new HashMap<String, Integer>();  
        this.insert(string);  
    }  
  
    public void insert(String string) {  
        for (String x : string.split(" ")) {  
            if (mp.containsKey(x)) {  
                mp.put(x, mp.get(x) + 1);  
            } else {  
                mp.put(x, 1);  
            }  
        }  
    }  
}
```

```

    }

    }

}

public Set<String> keySet() {
    return this.mp.keySet();
}

@Override
public String toString() {
    return mp.toString();
}
}

public class Part_2_Assignment_4 {
    public static void main(String[] args) {
        System.out.println("Enter a sentence, avoid putting any punctuations.");
        String s = new Scanner(System.in).nextLine();

        WordCounter wc = new WordCounter(s);

        System.out.println("Word count of each unique word is: ");
        System.out.println(wc);

    }
}

```

## Output

```

C:\Users\coolr\.jdk\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1\
Enter a sentence, avoid putting any punctuations.
lets count how many unique words are lets how how words see words here and there and everywhere check again
Word count of each unique word is:
{here=1, again=1, count=1, words=3, lets=2, check=1, many=1, how=3, see=1, are=1, and=2, everywhere=1, unique=1, there=1}

Process finished with exit code 0

```

## 5. Write a custom Comparator to sort a list of Employee objects by their salary and then by name if the salary is the same.

```

import java.util.Comparator;
import java.util.LinkedList;
import java.util.List;
import java.util.Scanner;

class Employee {

```

```

private int sal;
private String name;

Employee(String name, int sal) {
    this.name = name;
    this.sal = sal;
}

public int getSal() {
    return this.sal;
}

public void setSal(int sal) {
    this.sal = sal;
}

public String getName() {
    return this.name;
}

public void setName(String name) {
    this.name = name;
}

@Override
public String toString() {
    return "Employee{" +
        "sal=" + sal +
        ", name='" + name + '\'' +
        '}';
}
}

public class Part_2_Assignment_5 {

    static class EmployeeComparator implements Comparator<Employee> {
        @Override
        public int compare(Employee o1, Employee o2) {
            if (o1.getSal() > o2.getSal()) {
                return 1;
            } else if (o1.getSal() < o2.getSal()) {
                return -1;
            } else {
                return o1.getName().compareTo(o2.getName());
            }
        }
    }

    public static void main(String[] args) {

```

```

Scanner scan = new Scanner(System.in);
System.out.println("Number of Employee ");
int n = scan.nextInt();
System.out.println("Add Employee to the list");

List<Employee> ls = new LinkedList<>();

for (int i = 0; i < n; i++) {
    System.out.println("Add Employee");
    System.out.print("Enter Name : ");
    String name = scan.next();
    System.out.print("Enter Salary :");
    int sal = scan.nextInt();
    ls.add(new Employee(name, sal));
    System.out.println("Employee Added");
    try {
        Thread.sleep(300);
    } catch (InterruptedException e) {
        throw new RuntimeException(e);
    }
}

System.out.println("Before Sorting :");
System.out.println(ls);
ls.sort(new EmployeeComparator());
System.out.println("After Sorting :");
System.out.println(ls);
}
}

```

## Output

```

Number of Employee
5
Add Employee to the list
Add Employee
Enter Name : Sayan
Enter Salary :24000
Employee Added
Add Employee
Enter Name : Ashu
Enter Salary :20000
Employee Added
Add Employee
Enter Name : Rohit
Enter Salary :24000
Employee Added
Add Employee
Enter Name : Sumit
Enter Salary :63200
Employee Added
Add Employee
Enter Name : Sayan
Enter Salary :14000
Employee Added
Before Sorting :
[Employee{sal=24000, name='Sayan'}, Employee{sal=20000, name='Ashu'}, Employee{sal=24000, name='Rohit'}, Employee{sal=63200, name='Sumit'}, Employee{sal=14000, name='Sayan'}]
After Sorting :
[Employee{sal=14000, name='Sayan'}, Employee{sal=20000, name='Ashu'}, Employee{sal=24000, name='Rohit'}, Employee{sal=24000, name='Sayan'}, Employee{sal=63200, name='Sumit'}]

```

---

**Tools Used :**

IntelliJ IDE

java version "1.8.0\_411"

Java(TM) SE Runtime Environment (build 1.8.0\_411-b09)

Java HotSpot(TM) Client VM (build 25.411-b09, mixed mode, sharing)