**ACD\_BDD\_Session\_6\_Assignment\_1\_Main**

**Problem Statement**

Find the total number of medalsals that each country won in swimming.

**Solution**

package com.acadgild.olympics;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.TextInputFormat;

import org.apache.hadoop.mapred.TextOutputFormat;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class OlympicsMedals {

public static class Map extends Mapper <LongWritable, Text, Text, IntWritable>

{

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException

{

String[] splits = value.toString().split("\t", -1);

for(int i=0;i<splits.length;i++){

if("".equals(splits[i])) (splits[i]) = "0";

}

String country = splits[2];

int medals = Integer.parseInt(splits[9]);

String sport = splits[5];

if(sport.equalsIgnoreCase("Swimming")){

context.write(new Text(country), new IntWritable(medals));

}

}

}

public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable>

{

public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException

{

int count = 0;

for(IntWritable value : values)

{

count = count + value.get();

}

context.write(key, new IntWritable(count));

}

}

public static void main(String[] args) throws IOException, ClassNotFoundException, InterruptedException {

// TODO Auto-generated method stub

Configuration conf = new Configuration();

Job job = new Job(conf, "Olympics Swimmimg");

job.setJarByClass(OlympicsSwimming.class);

job.setMapperClass(Map.class);

job.setReducerClass(Reduce.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

job.waitForCompletion(true);

}

}

**Output**

