Map -reduce is a data processing method which accepts large volumes of data and reduces it into useful aggregated results.

```
db.test.insertMany([
{
	"_id" : ObjectId("59c2468246f79abd1840c884"),
	"name" : "Arun",
	"roll" : 1,
	"class" : "T.E",
	"branch" : "comp"
},
{
	"_id" : ObjectId("59c2469e46f79abd1840c885"),
	"name" : "Yash",
	"roll" : 2,
	"class" : "T.E",
	"branch" : "IT"
},
{
	"_id" : ObjectId("59c246a946f79abd1840c886"),
	"name" : "Yashashree",
	"roll" : 3,
	"class" : "S.E",
	"branch" : "comp"
},
{
	"_id" : ObjectId("59c246b246f79abd1840c887"),
	"name" : "Rutuja",
	"roll" : 4,
	"class" : "S.E",
	"branch" : "ENTC"
},
{
	"_id" : ObjectId("59c246bd46f79abd1840c888"),
	"name" : "Gaurav",
	"roll" : 5,
	"class" : "B.E",
	"branch" : "comp"
},
{
	"_id" : ObjectId("59c246d646f79abd1840c889"),
	"name" : "Aishwarya",
	"roll" : 6,
	"class" : "B.E",
	"branch" : "comp"
},
{
	"_id" : ObjectId("59c246a946f79abd1840c281"),
	"name" : "Anirudh",
	"roll" : 18,
	"class" : "T.E",
	"branch" : "comp"
}])
```

Syntax for Map-Reduce function in MongoDB:

```
db.collection.mapReduce(Mapfunc, Reducefunc,
{ query:{field:'<value>'}, out:'resultant' } )
```

Mapper function: It accepts data and emits the specified key-value pairs from documents of given collection

Reduce function: It accepts the key-value pairs emitted by the mapper function and reduces it to get a smaller aggregation result. If a key has multiple values, then it accepts an array of those values for that key.

Query: Find year-wise total number of students from computer branch.

```
var Mapfunc= function(){emit(this.class,1)}

var Reducefunc= function(key,values)

{returnArray.sum(values)}


db.test.mapReduce(Mapfunc,Reducefunc,

{out:'resultant',query:{branch:'comp'} } )
```

```
{
    "result" : "resultant",
    "timeMillis" : 460,
    "counts" : {
        "input" : 5,
        "emit" : 5,
        "reduce" : 2,
        "output" : 3
    },
    "ok" : 1
}
```

Collection for Assignment:

```
db.classes.insert({
class : "Philosophy 101",
startDate : new Date(2016, 1, 10),
students : [
{fName : "Dale", lName : "Cooper", age : 42},
{fName : "Lucy", lName : "Moran", age : 35},
{fName : "Tommy", lName : "Hill", age : 44}
],
cost : 1600,
professor : "Paul Slugman",
topics : "Socrates,Plato,Aristotle,Francis Bacon",
book:
{
isbn: "1133612105",
title: "Philosophy : A Text With Readings",
price: 165.42
}
})

db.classes.insert({
class : "College Algebra",
startDate : new Date(2016, 1, 11),
students : [
{fName : "Dale", lName : "Cooper", age : 42},
{fName : "Laura", lName : "Palmer", age : 22},
{fName : "Donna", lName : "Hayward", age : 21},
{fName : "Shelly", lName : "Johnson", age : 24}
],
cost : 1500,
professor : "Rhonda Smith",
topics : "Rational Expressions,Linear Equations,Quadratic Equations",
book:
{
isbn: "0321671791",
title: "College Algebra",
price: 179.40
}
})

db.classes.insert({
class : "Astronomy 101",
startDate : new Date(2016, 1, 11),
students : [
{fName : "Bobby", lName : "Briggs", age : 21},
```

```
{fName : "Laura", lName : "Palmer", age : 22},
{fName : "Audrey", lName : "Horne", age : 20}
],
cost : 1650,
professor : "Paul Slugman",
topics : "Sun,Mercury,Venus,Earth,Moon,Mars",
book:
{
isbn: "0321815351",
title: "Astronomy: Beginning Guide to Univ",
price: 129.45
}
})

db.classes.insert({
class : "Geology 101",
startDate : new Date(2016, 1, 12),
students : [
{fName : "Andy", lName : "Brennan", age : 36},
{fName : "Laura", lName : "Palmer", age : 22},
{fName : "Audrey", lName : "Horne", age : 20}
],
cost : 1450,
professor : "Alice Jones",
topics : "Earth,Moon,Elements,Minerals",
book:
{
isbn: "0321814061",
title: "Earth : An Introduction to Physical Geology",
price: 130.65
}
})

db.classes.insert({
class : "Biology 101",
startDate : new Date(2016, 1, 11),
students : [
{fName : "Andy", lName : "Brennan", age : 36},
{fName : "James", lName : "Hurley", age : 25},
{fName : "Harry", lName : "Truman", age : 41}
],
cost : 1550,
professor : "Alice Jones",
topics : "Earth,Cell,Energy,Genetics,DNA",
book:
{
```

```
isbn: "0547219474",
title: "Holt McDougal Biology",
price: 104.30
}
})

db.classes.insert({
class : "Chemistry 101",
startDate : new Date(2016, 1, 13),
students : [
{fName : "Bobby", lName : "Briggs", age : 21},
{fName : "Donna", lName : "Hayward", age : 21},
{fName : "Audrey", lName : "Horne", age : 20},
{fName : "James", lName : "Hurley", age : 25}
],
cost : 1600,
professor : "Alice Jones",
topics : "Matter,Energy,Atom,Periodic Table",
book:
{
isbn: "0547219474",
title: "Chemistry : Matter and Change",
price: 104.30
}
})
```

Queries:

1. How many classes does “Alice Jones” teach
2. Find the total no. of students enrolled for each class
3. Find the total no. of classes conducted by each professor and also the total cost to attend each of the professor’s classes

Expected Solutions:

1)

```
var mapFunc2 = function(){
emit(this.professor,1);
}

var reduceFunc2 = function(professor, count){
return Array.sum(count);
}

db.classes.mapReduce(
mapFunc2,
reduceFunc2,{
query:{professor: "Alice Jones"},
out: "map_ex_2"
}
)
```

3) Emit a document consisting of both **count as 1 & the cost to attend each professor's classes** to the reduce function:

```
var mapFunc4 = function(){
emit(this.professor, { count: 1, cost: this.cost });
}
```

Reduce down to professor and cost:

```
var reduceFunc4 = function(professor, values){
var value = { count: 0, cost: 0 };

for(i = 0; i < values.length; i++){
value.count += values[i].count;
value.cost += values[i].cost;
}
return value;
}
```