

## Find MemberShip Category Count

### Find Membership Category Count

**ZEE Shopping mall** wanted to know how many Members are available for each of their Membership category. The member ship category is of three types (Gold, Silver and Platinum).

You being their software consultant have been approached by them to develop an application which can be used for managing their business. You need to implement a java program using thread to find out the count of members in each membership category. Membership details should be obtained from the user in the console.

#### Component Specification: Member (Model Class)

Type(Class)	Attributes	Methods	Responsibilities
<b>Member</b>	String memberId  String memberName  String category	Include getters and setter method for all the attributes.  Include a three argument constructor in the given order – memberId, memberName and category.	Set the values for all the attributes via constructor.

**Note:** The class and methods should be declared as public and all the attributes should be declared as private.

#### Requirement 1: Count the number of members

Count the number of members available in the memberList based on the membership category to be searched and set the value to count attribute.

#### Component Specification: ZEEShop (Thread Class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities
	ZEEShop	String memberCategory  int count  List<Member>memberList	Include getters and setter method for all the attributes.  Include a two argument	Set the values for all the attributes via constructor.

			constructor with arguments – memberCategory and memberList.	
Count the number of members	ZEEShop		void run()	Count the number of members based on the Membership category and set the value to the countattribute.

**Note:** The class and methods should be declared as public and all the attributes should be declared as private.

Create a class called **Main** with the main method and get the inputs like **number of members**, **member details**, **number of times Membership category needs to be searched** and **Membership category to be searched** from the user.

The member details will be in the form of String in the following format **memberId:memberName:category**.

Parse the member details and set the values for all attributes in **Member** class using **constructor**.

Invoke the ZEEShop thread class for each memberCategory and count the number of members in that category and display the count as shown in the sample input and output.

**Assumption:** The **memberCategory** is **case –sensitive** and will be of only three values – **Platinum** or **Gold** or **Silver**.

**Note:**

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the remaining text represents the output.

Ensure to follow object oriented specifications provided in the question description.  
Ensure to provide the names for classes, attributes and methods as specified in the question description.

Adhere to the code template, if provided.

**Sample Input / Output 1:**

Enter the number of Members:

5

Enter the Member Details:

**101:Tom:Gold**

Enter the Member Details:

**102:Sam:Gold**

Enter the Member Details:

**103:John:Silver**

Enter the Member Details:

**104:Rose:Platinum**

Enter the Member Details:

**105:Tint:Silver**

Enter the number of times Membership category needs to be searched:

4

Enter the Category

**Gold**

Enter the Category

**Silver**

Enter the Category

**Platinum**

Enter the Category

**Gold**

Gold:2

Silver:2

Platinum:1

Gold:2

---

# Grade Calculation

## Grade Calculation

Rita is working as a science teacher in an International school. She is the Class Teacher of class V and was busy in calculating the grade for each student in her class, based on his/her total marks obtained in SA1 assessment.

Since she found it very difficult to calculate the grade, she approached you to develop an application which can be used for completing her task faster. You need to implement a java program using thread to calculate the grade for each student. Student details should be obtained from the user in the console.

**Requirement 1:** Calculate the grade for each student.

Calculate the grade based on total marks (sum of all marks) as shown below obtained by each student and set the same in result attribute for respective student.

Total Marks	Grade
400 to 500	A
300 to 399	B
200 to 299	C
Less than 200	E

**Assumption:** Each student will have only five subjects and marks of each subject will be greater than or equal to 0 and lesser than or equal to 100. Hence the maximum Total marks obtained by each student will be 500. And the minimum Total marks obtained by each student will be 0.

### Component Specification: GradeCalculator (Thread Class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities
	GradeCalculator	String studName  char result	Include getters and setter method for all the attributes.	Set the values for all the attributes via constructor.

		int[] marks	Include a two argument constructor in the given order – studName and marks.	
calculate the grade for each student	GradeCalculator		public void run()	Calculate the grade based on total marks and set the same to result attribute.

**Note:** The class and methods should be declared as public and all the attributes should be declared as private.

Create a class called **Main** with the main method and get the inputs like **number of threads** and **Student details** from the user.

The student details will be in the form of String in the following format **studName:mark1:mark2:mark3:mark4:mark5**.

Parse the student details and set the values of studName and marks attributes in **GradeCalculator** thread class using **constructor**.

Invoke the **GradeCalculator** thread class to calculate the grade based on total marks and set the same to result attribute.

Display the Student name and Grade obtained by each student as shown in the sample input and output.

**Note:**

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the remaining text represents the output.

Ensure to follow object oriented specifications provided in the question description. Ensure to provide the names for classes, attributes and methods as specified in the question description.

Adhere to the code template, if provided.

**Sample Input / Output 1:**

Enter the number of Threads:

**4**

Enter the String:

**Jeba:100:80:90:40:55**

Enter the String

**David:10:8:9:40:5**

Enter the String

**Adam:90:80:90:50:75**

Enter the String

**Rohit:99:99:99:99:99**

Jeba:B

David:E

Adam:B

Rohit:A