**Q 1 . What is the output ?**

**class** A02{

**public** **void** f1(String a) {

System.***out***.println("String method called");

}

}

**class** B01 **extends** A02{

**public** **void** f1(Object a) {

System.***out***.println("Ojbect method called");

}

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

A02 obj=**new** B01();

obj.f1(**null**);

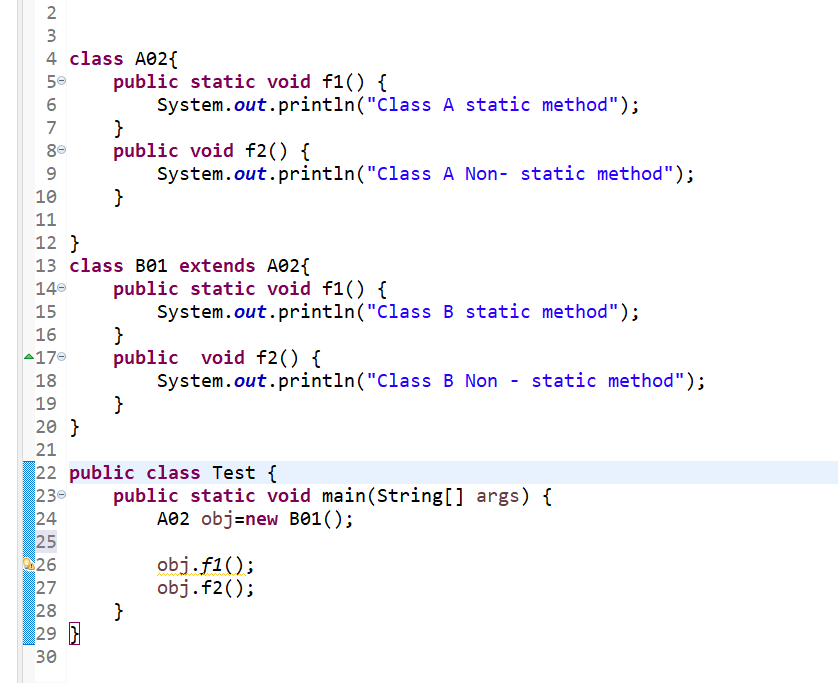
}

}

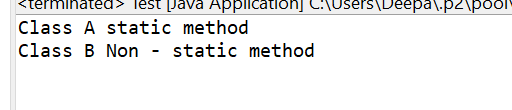
**Ans 1:**

String method called.

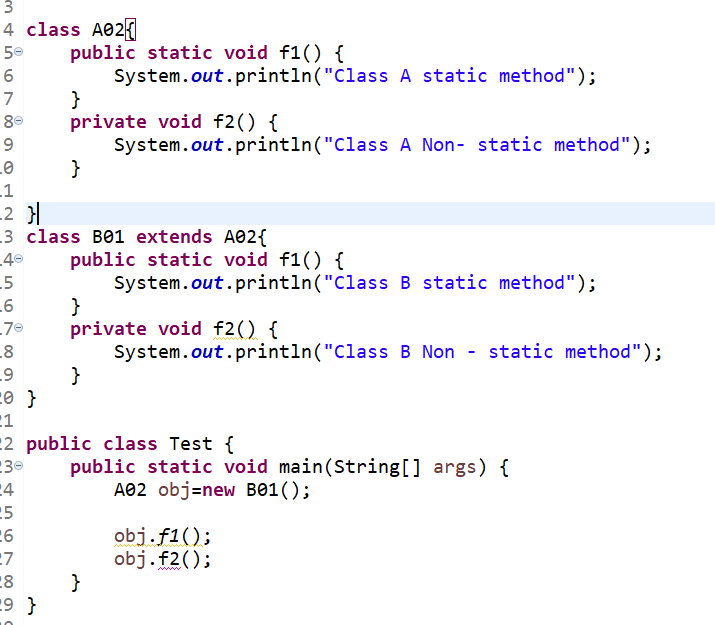
**Q 2. What is output ?**



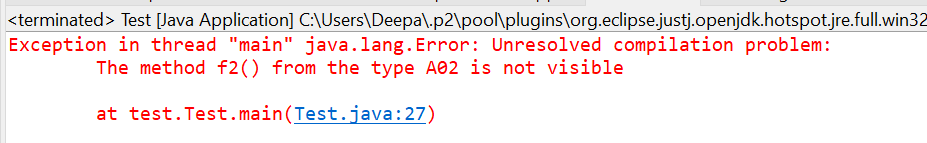
Output 2:



**Q 3 . What is output ? (Private method overriding);**



**Output 3:**

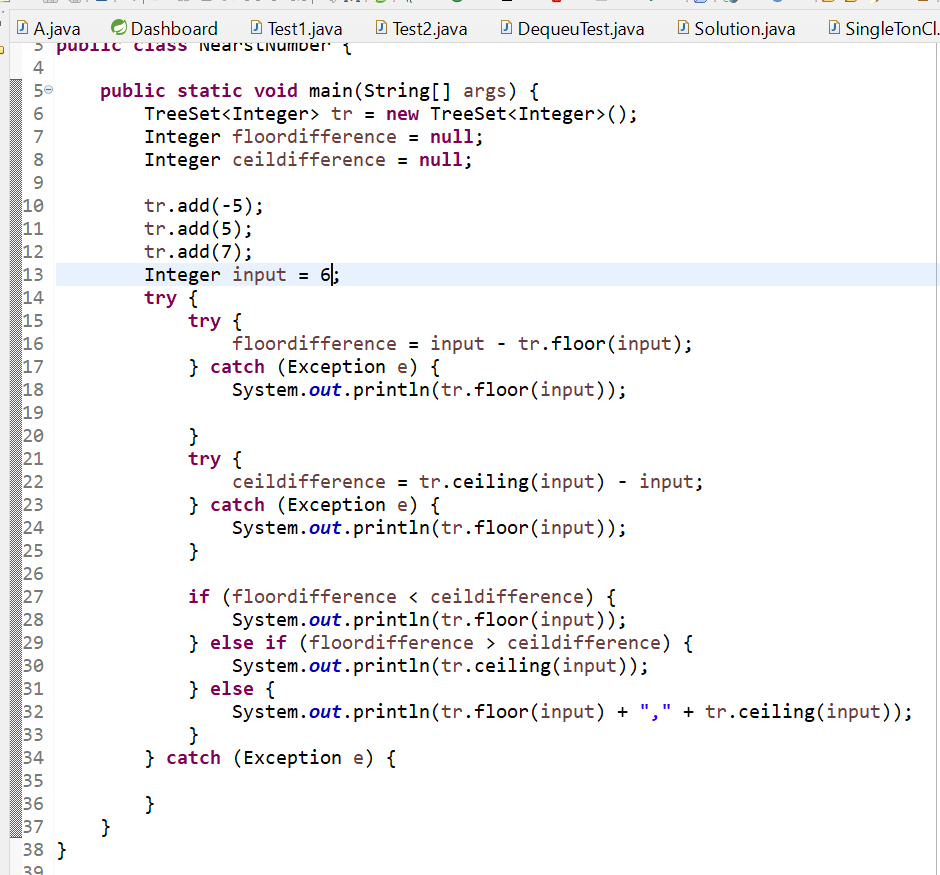


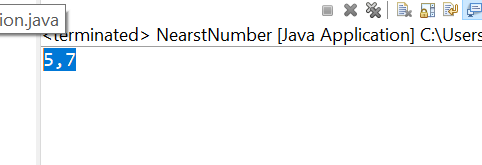
Q. find nearest number ?

Int[] ar=new int[]{-5,5,7}

Input : 6

Output: 5,7





Q .

Problem :

How can you get unique records on the basis of Employee id ?

Data :

Employee [id=1, name=Deepak, salary=1000]

Employee [id=2, name=DJ, salary=2000]

Employee [id=3, name=Anuj, salary=3000]

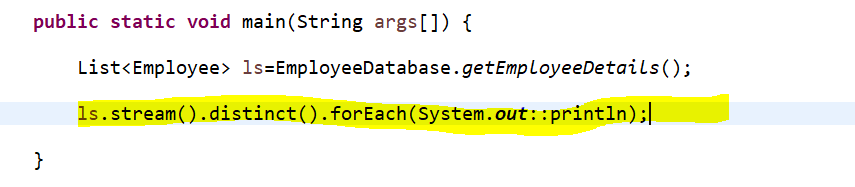
Employee [id=4, name=Kaku, salary=4000]

Employee [id=1, name=Deepak, salary=1000]

Employee [id=5, name=gyan, salary=1000]

Solution :

1. Override hashcode and equals method of employee class .



Output:

Employee [id=1, name=Deepak, salary=1000]

Employee [id=2, name=DJ, salary=2000]

Employee [id=3, name=Anuj, salary=3000]

Employee [id=4, name=Kaku, salary=4000]

Employee [id=5, name=gyan, salary=1000]

Q .

Problem :

// print employe base on salary higest to lowest.

Data :

Employee [id=1, name=Deepak, salary=1000]

Employee [id=2, name=DJ, salary=2000]

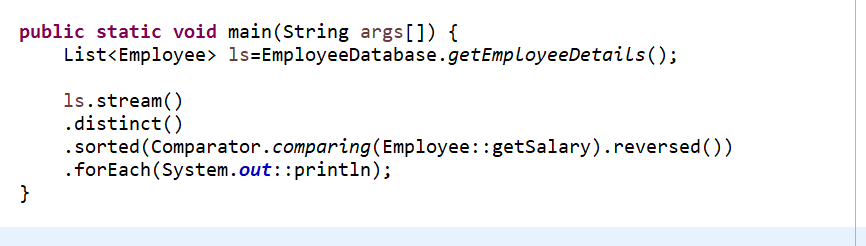
Employee [id=3, name=Anuj, salary=3000]

Employee [id=4, name=Kaku, salary=4000]

Employee [id=1, name=Deepak, salary=1000]

Employee [id=5, name=gyan, salary=1000]

Solution :



Output:

Employee [id=4, name=Kaku, salary=4000]

Employee [id=3, name=Anuj, salary=3000]

Employee [id=2, name=DJ, salary=2000]

Employee [id=1, name=Deepak, salary=1000]

Employee [id=5, name=gyan, salary=1000]

Q .

Problem :

// // calulate total of salaryes

Data :

Employee [id=1, name=Deepak, salary=1000]

Employee [id=2, name=DJ, salary=2000]

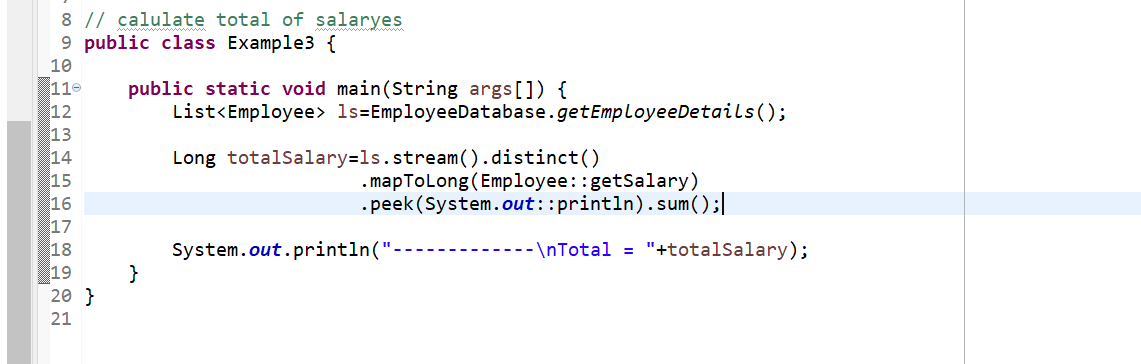
Employee [id=3, name=Anuj, salary=3000]

Employee [id=4, name=Kaku, salary=4000]

Employee [id=1, name=Deepak, salary=1000]

Employee [id=5, name=gyan, salary=1000]

Solution :



Output:

1000

2000

3000

4000

1000

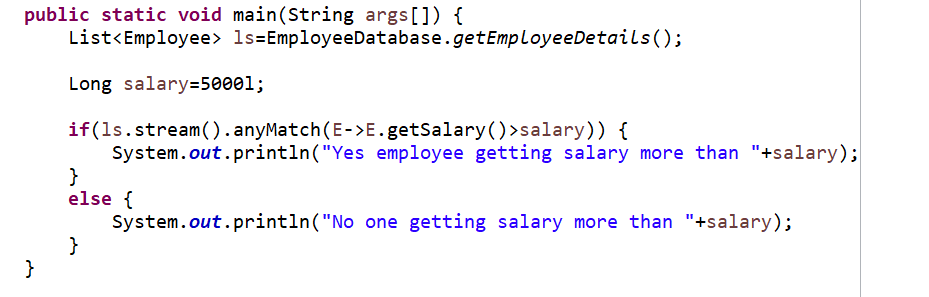
-------------

Total = 11000

Q . problem :

// Can you check if any of employee getting salary higer then 5000

Solution:



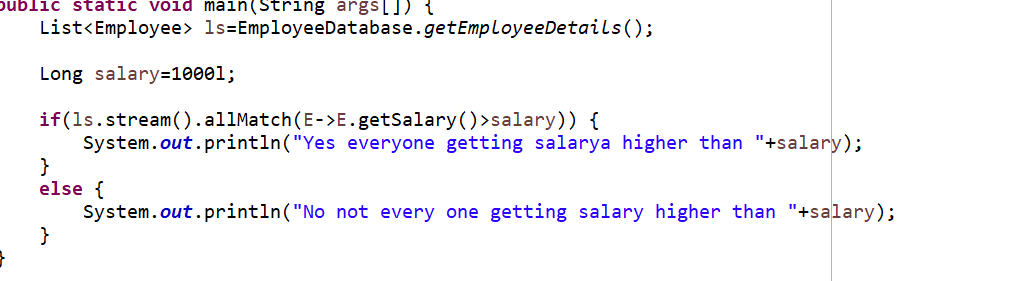
Output:

No one getting salary more than 5000

Q.Problem :

// I hope every one getting salary more than 1000

Solution :

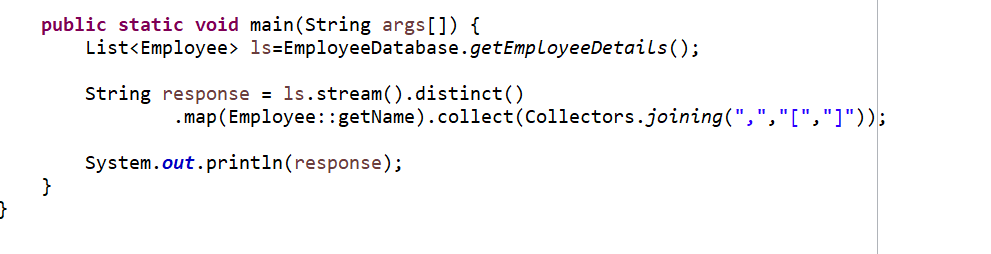


Output :

No not every one getting salary higher than 1000

Q.problem:

// can you print all name comma saperated and close them into []



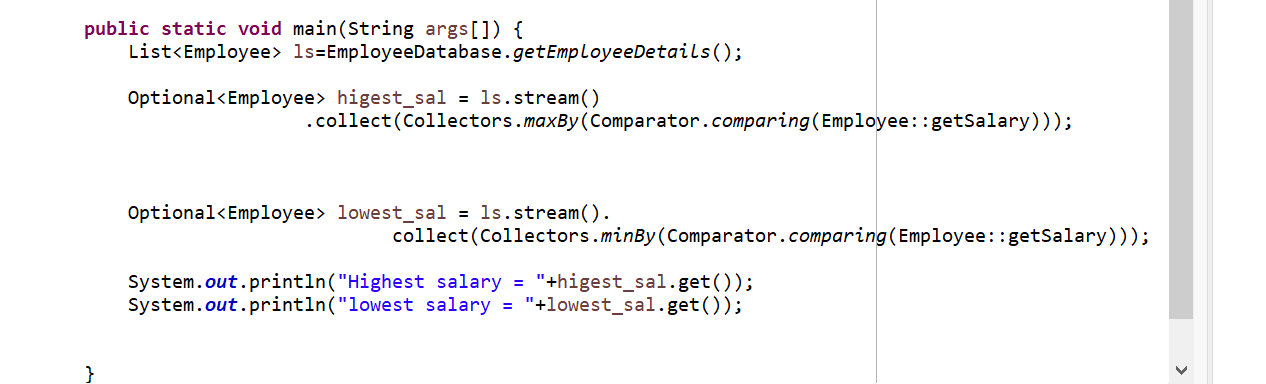
Output :

[Deepak,DJ,Anuj,Kaku,gyan]

Problem :

// print higest salarya employee and lowest salary employee

Solution:



Output:

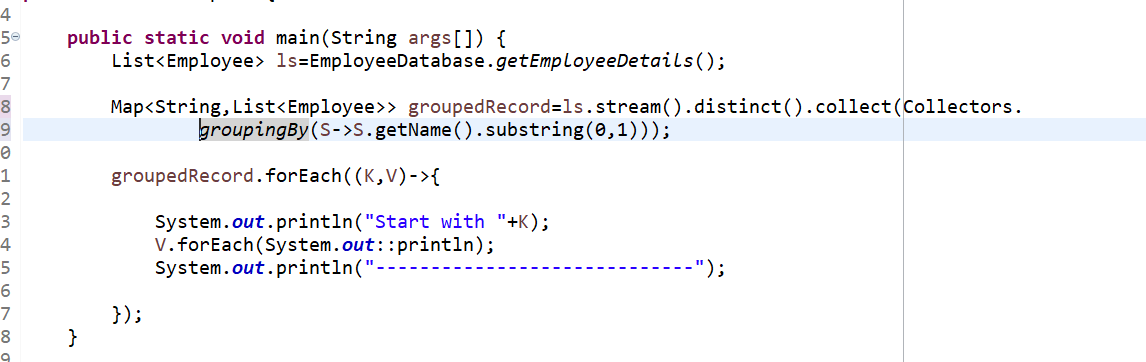
Highest salary = Employee [id=4, name=Kaku, salary=4000]

lowest salary = Employee [id=1, name=Deepak, salary=1000]

Problem :

// Can you group employe on the base of first latter of there name;

Solution :



Output:

Start with A

Employee [id=3, name=Anuj, salary=3000]

-----------------------------

Start with D

Employee [id=1, name=Deepak, salary=1000]

Employee [id=2, name=DJ, salary=2000]

-----------------------------

Start with g

Employee [id=5, name=gyan, salary=1000]

-----------------------------

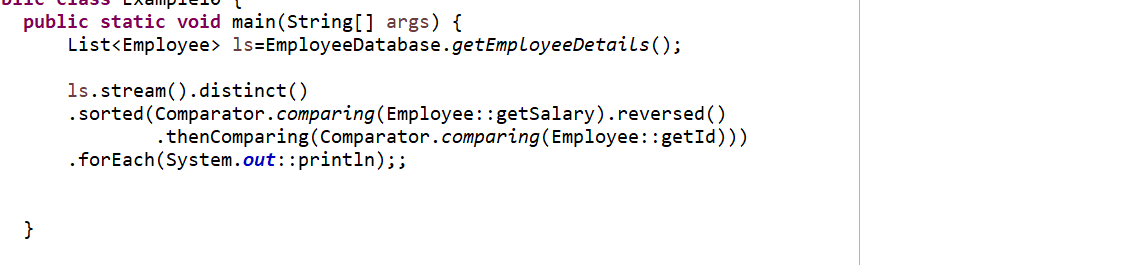
Start with K

Employee [id=4, name=Kaku, salary=4000]

Problem :

// compare salary to hight to low if salary is same the id low to hight

Solution:



Output :

Employee [id=4, name=Kaku, salary=4000]

Employee [id=3, name=Anuj, salary=3000]

Employee [id=2, name=DJ, salary=2000]

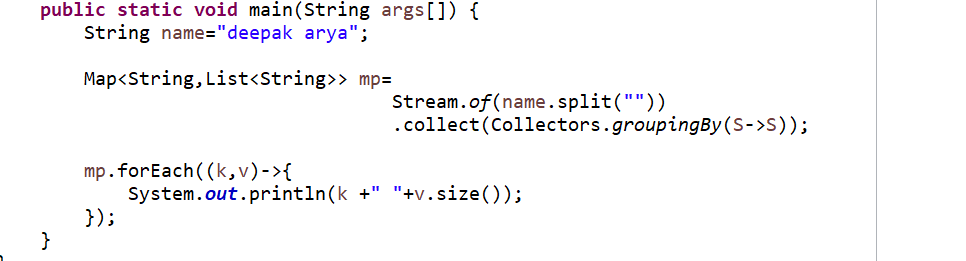
Employee [id=1, name=Deepak, salary=1000]

Employee [id=5, name=gyan, salary=1000]

Problem:

// retive occurence of charater

Solution :



Output :

1

p 1

a 3

r 1

d 1

e 2

y 1

k 1