1. **import** java.util.Scanner;

@FunctionalInterface

**interface** Airthematic {

**int** operations(**int** a, **int** b);

}

**public** **class** Airth\_op {

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

Scanner scn = **new** Scanner(System.***in***);

System.***out***.println("Enter two arguments");

**int** n = scn.nextInt();

**int** m = scn.nextInt();

Airthematic add = (a,b) -> (a+b);

System.***out***.println("Addition: "+add.operations(n,m));

Airthematic sub = (a,b) -> (a-b);

System.***out***.println("Subtraction: "+sub.operations(n,m));

Airthematic mul = (a,b) -> (a\*b);

System.***out***.println("Multiplication: "+mul.operations(n,m));

Airthematic div = (a,b) -> (a/b);

System.***out***.println("Division: "+div.operations(n,m));

}

1. **public** **class** orders {

**private** **int** order\_id;

**private** **int** price;

**private** String status;

**public** orders(**int** order\_id, **int** price, String status) {

**this**.order\_id = order\_id;

**this**.price = price;

**this**.status = status;

}

**public** **int** getPrice() {

**return** price;

}

**public** **void** setPrice(**int** price) {

**this**.price = price;

}

**public** **int** getOrder\_id() {

**return** order\_id;

}

**public** **void** setOrder\_id(**int** order\_id) {

**this**.order\_id = order\_id;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** String toString() {

**return** "Order: " + order\_id + " - price = " + price + " , status = " + status;

}

}

**import** java.util.Arrays;

**import** java.util.List;

**import** java.util.function.Consumer;

**import** java.util.function.Predicate;

**public** **class** Print\_orders {

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

List<orders> order = Arrays.*asList*(

**new** orders(1, 4000, "INPROCESS"),

**new** orders(2, 10000, "ACCEPTED"),

**new** orders(3, 10000, "INPROCESS"),

**new** orders(4, 7000, "COMPLETED"),

**new** orders(5, 10000, "COMPLETED"),

**new** orders(6, 50000, "ACCEPTED"),

**new** orders(7, 60000, "COMPLETED"),

**new** orders(8, 15000, "ACCEPTED"),

**new** orders(9, 10000, "INPROCESS"),

**new** orders(10, 20000, "COMPLETED")

);

System.***out***.println("\n\nPrinting the orders which price is greater 10000 and status is ACCEPTED/COMPLETED ");

*Orderlist*(order, o->(o.getPrice()> 10000 && (o.getStatus()== "ACCEPTED"|| o.getStatus() == "COMPLETED")),o->System.***out***.println(o) );

}

**private** **static** **void** Orderlist(List<orders> order, Predicate<orders> predicate, Consumer<orders> consumer) {

**for**(orders o : order) {

**if**(predicate.test(o))

consumer.accept(o);

}

}

}