ASSIGNMENT ON LAMBDA EXPRESSION

1.

ANS :-

**package** Expression;

@FunctionalInterface

**interface** Addition {

**public** **int** add(**int** a, **int** b);

}

**package** Expression;

@FunctionalInterface

**public** **interface** Division {

**public** **double** divide(**int** a, **double** b);

}

**package** Expression;

@FunctionalInterface

**public** **interface** Multiplication {

**public** **double** multiply();

}

**package** Expression;

@FunctionalInterface

**public** **interface** Substraction {

**public** **int** substract(**int** a, **int** b);

}

**package** Expression;

**public** **class** Calculate {

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

Addition c = (**int** a, **int** b) -> a + b;

**int** ans = c.add(10, 20);

System.***out***.println("Addition = " + ans);

System.***out***.println("---------------------------------------------------------");

Substraction s = (**int** a, **int** b) -> a - b;

**int** pr = s.substract(10, 20);

System.***out***.println("Subbstraction = " + pr);

System.***out***.println("---------------------------------------------------------");

Division div = (**int** a, **double** b) -> a / b;

**double** d = div.divide(50, 3);

System.***out***.println("Divide = " + d);

System.***out***.println("---------------------------------------------------------");

**int** p = 20;

**double** q = 30.235;

Multiplication mul = () -> p \* q;

**double** mu = mul.multiply();

System.***out***.println("Multiplication = " + mu);

}

}

**OUTPUT:**

Addition = 30

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

Subbstraction = -10

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

Divide = 16.666666666666668

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

Multiplication = 604.7

2.

**package** secondque;

**import** java.util.ArrayList;

**import** java.util.List;

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

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

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

**private** **float** productPrice;

**private** String status;

**public** **float** getProductPrice() {

**return** productPrice;

}

**public** **void** setProductPrice(**float** productPrice) {

**this**.productPrice = productPrice;

}

**public** String getStatus() {

**return** status;

}

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

**this**.status = status;

}

**public** Test(**float** productPrice ,String status) {

// **TODO** Auto-generated method stub

**this**.productPrice= productPrice;

**this**.status=status;

}

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

// **TODO** Auto-generated method stub

Test test1 = **new** Test(1000000, "ACTIVATED");

Test test2 = **new** Test(2000000, "COMPLETED");

Test test3 = **new** Test(1000, "ACTIVATED");

Test test4= **new** Test(2400000, "COMPLETED");

Test test5 = **new** Test(1000000, "ACTIVATED");

List<Test> sEmp = **new** ArrayList<>();

sEmp.add(test1);

sEmp.add(test2);

sEmp.add(test3);

sEmp.add(test4);

sEmp.add(test5);

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

*Orderlist*(sEmp, o->(o.getProductPrice()> 1000000 && (o.getStatus()== "ACCEPTED"|| o.getStatus() == "COMPLETED")),

o->System.***out***.println(o.productPrice) );

}

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

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

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

consumer.accept(o);

}

}

}

OUTPUT:

Printing the orders which price is greater 10000 and status is ACCEPTED/COMPLETED

2000000.0

2400000.0