Q1) Write the following a functional interface and implement it using lambda:

* To check whether the first number is greater than second number or not, Parameter (int ,int ) Return type boolean

public interface Fi {

public boolean Greater(int a , int b);

}

public class Add {

public static void main(String[] args) {

Fi aw = (int a, int b) -> {

boolean ans = false;

if (a > b) {

ans = true;

} else {

ans = false;

}

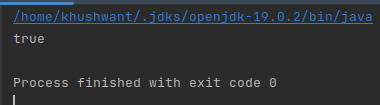
return ans;

};

System.*out*.println(aw.Greater(20,12));

}

}



* Increment the number by 1 and return incremented value Parameter (int) Return int

public interface Fi {

public int Incr(int a );

}

public class Inc {

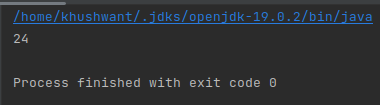
public static void main(String[] args) {

Fi ab = (int a) -> { return ++a;};

System.*out*.println(ab.Incr(23));

}

}



* Concatination of 2 string Parameter (String , String ) Return (String)

public interface Fi {

public String Conc(String a,String b);

}

public class Con {

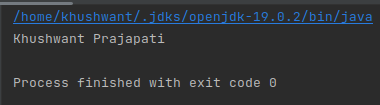
public static void main(String[] args) {

Fi ab = (String a,String b) -> { return a + b; };

System.*out*.println(ab.Conc("Khushwant"," Prajapati"));

}

}



* Convert a string to uppercase and return . Parameter (String) Return (String)

public interface Fi {

public String Conc(String a);

}

public class Con {

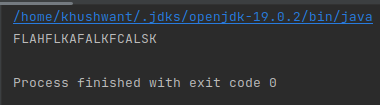
public static void main(String[] args) {

Fi ab = (String a) -> { String b = a.toUpperCase(); return b; };

System.*out*.println(ab.Conc("flahflkafalkfcalsk"));

}

}



Q2) Using (instance) Method reference create and apply add and subtract method and using (Static) Method reference create and apply multiplication method for the functional interface created

interface myInterface4{

int methodRef(int a,int b);

}

public class Ques2 {

static int multi(int a, int b){

return a\*b;

}

int sum(int a, int b){

return a+b;

}

int sub(int a, int b){

return a-b;

}

public static void main(String[]args){

myInterface4 multi = Ques2::*multi*;

System.*out*.println("Multiple = "+multi.methodRef(2,3));

myInterface4 sum = new Ques2()::sum;

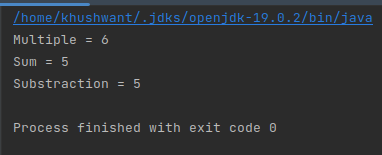
System.*out*.println("Sum = "+sum.methodRef(2,3));

myInterface4 sub = new Ques2()::sub;

System.*out*.println("Substraction = "+sub.methodRef(7,2));

}

}



Q3) Implement multiple inheritance with default method inside interface.

interface Animal {

default void move() {

System.*out*.println("This animal can move");

}

}

interface Mammal {

default void eat() {

System.*out*.println("This mammal can eat");

}

}

class Dog implements Animal, Mammal {

public static void main(String args[]) {

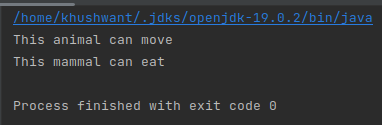
Dog dog = new Dog();

dog.move();

dog.eat();

}

}



Q4) Write a program to implement constructor reference

@FunctionalInterface

interface MyInter {

Message send(String message);

}

class Message {

public Message(String message) {

System.*out*.println(message);

}

}

class Test {

public static void main(String[] args) {

MyInter myInterface = Message::new;

myInterface.send("Hello!");

}

}

