EXERCISE

1. **Write the following a functional interface and implement it using lambda:**
   a. **First number is greater than second number or not Parameter (int ,int) Return boolean**

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

   c.Concatination of 2 string Parameter (String , String) Return (String)

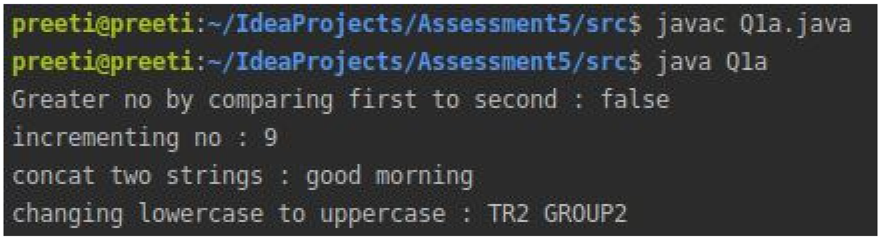
   d. Convert a string to uppercase and return . Parameter (String) Return (String)

CODE:

```
interface Greaterthanint{
	boolean greaterint(int a,int b);
}

interface Incremental1 {
	int increment1(int a);
}

interface Concatenation{
	String concatefun(String a,String b);
}
interface Uppercase{
	String upper(String a);
}
class Q1a{
	public static void main(String[] args) {
	Greaterthanint ob1=(a,b)->{
	return a>b;
	};
	System.out.println("Greater no by comparing first to second : "+ob1.greaterint(3,8));

	Incremental1 ob2=(a)->{
	return a+1;
	};
	System.out.println("incrementing no : "+ob2.increment1(8));

	Concatenation ob3=(a,b)->{
	return a+b;
	};
	System.out.println("concat two strings : "+ob3.concatefun("good ","morning"));
```

```
        Uppercase ob4=(a)->{
        return  a.toUpperCase();
        };
        System.out.println("changing lowercase to uppercase : "+ob4.upper("tr2 group2"));
        }
}
```

OUTPUT:



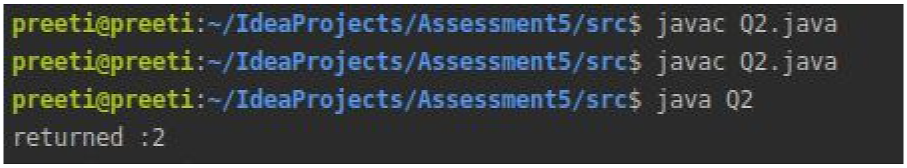
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q1a.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q1a
Greater no by comparing first to second : false
incrementing no : 9
concat two strings : good morning
changing lowercase to uppercase : TR2 GROUP2
```

2. **Create a functional interface whose method takes 2 integers and return one integer.**

CODE:

```
interface Inter
        {
        public int operation(int a,int b);
        }
public class Q2
{
        public static void main(String[] args)
        {
        Inter i=(a,b)-> {
        return b;
        };
        System.out.println("returned :"+i.operation(3,2));
        }
}
```

OUTPUT:



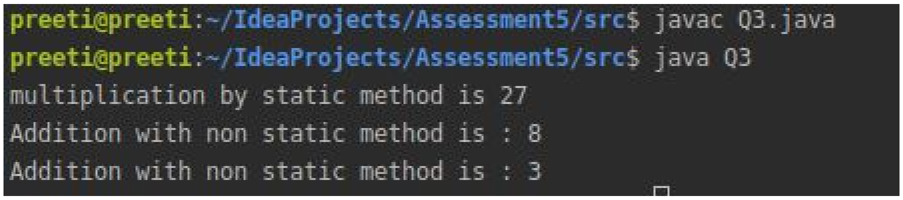
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q2.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q2.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q2
returned :2
```

3. **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.**

CODE:

```
interface Mysum{
        int exercise(int a,int b);
}

class Exercise {
        static int multiply(int a, int b){
        return a*b;}

        int add(int a,int b){
        return a+b;
        }

        int sub(int a,int b){
        return a-b;
        }
}
public class Q3{
        public static void main(String[] args) {
        Mysum obj=Exercise::multiply;
        System.out.println(("multiplication by static method is "+obj.exercise(9,3)));

        obj=new Exercise()::add;
        System.out.println("Addition with non static method is : "+obj.exercise(3,5));

        obj=new Exercise()::sub;
        System.out.println("Addition with non static method is : "+obj.exercise(9,6));
        }
}
```

OUTPUT:



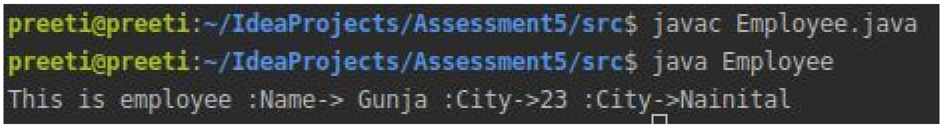
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q3.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q3
multiplication by static method is 27
Addition with non static method is : 8
Addition with non static method is : 3
```

4. **Create an Employee Class with instance variables (String) name, (Integer)age, (String)city and get the instance of the Class using constructor reference**

CODE:

```
public class Employee {
        String name;
        int age;
        String city;
        Employee(String name,int age,String city){
        this.name=name;
        this.age=age;
        this.city=city;
        }

        protected void show(){

        System.out.println("This is employee"+" :Name-> "+name+" :City->"+age+" :City->"+city+" ");
        }

        public static void main(String[] args) {
        Employee emp=new Employee ("Gunja",23,"Nainital");
        emp.show();
        }
}
interface constrefinterface{
        Employee auto (String name,int age,String city);
}
```

OUTPUT:



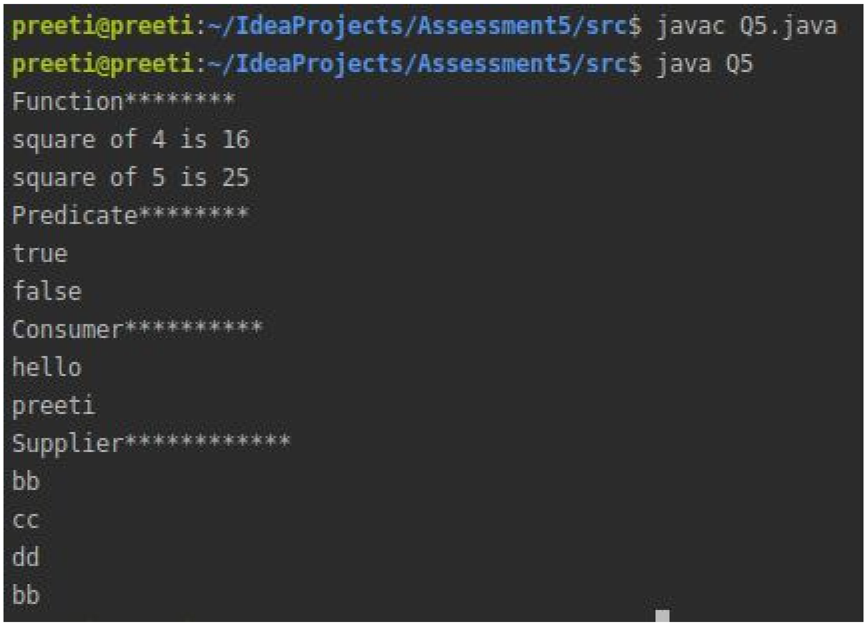
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Employee.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Employee
This is employee :Name-> Gunja :City->23 :City->Nainital
```

5. **Implement following functional interfaces from java.util.function using lambdas:**
    a. **(1) Consumer**
    b. **(2) Supplier**
    c. **(3) Predicate**
    d. **(4) Function**

CODE:

```
import java.util.function.Consumer;
import java.util.function.Function;
import java.util.function.Predicate;
import java.util.function.Supplier;

public class Q5 {
        public static void main(String[] args) {
        System.out.println("Function********");
        Function<Integer,Integer>f=i->i*i;
        System.out.println("square of 4 is "+f.apply(4));
        System.out.println("square of 5 is "+f.apply(5));

        System.out.println("Predicate********");
        Predicate<Integer> p=i->i%2==0;
        System.out.println(p.test(4));
        System.out.println(p.test(5));

        System.out.println("Consumer**********");
        Consumer<String> c=s-> System.out.println(s);
        c.accept("hello");
        c.accept("preeti");

        System.out.println("Supplier************");
        Supplier<String> d=()->{
        String[] d1={"aa","bb","cc","dd"};
        int x=(int)(Math.random()*3+1);
        return d1[x];
        };
        System.out.println(d.get());
        System.out.println(d.get());
        System.out.println(d.get());
        System.out.println(d.get());
        }
}
```

OUTPUT:



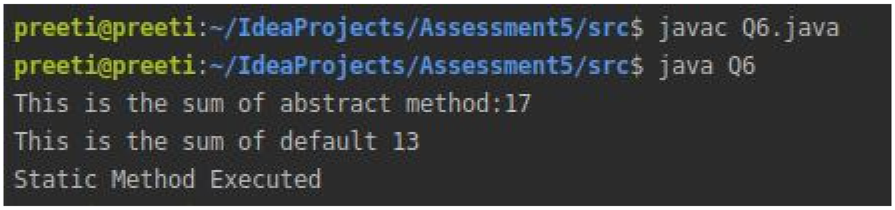
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q5.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q5
Function********
square of 4 is 16
square of 5 is 25
Predicate********
true
false
Consumer**********
hello
preeti
Supplier************
bb
cc
dd
bb
```

6. Create and access default and static method of an interface.

CODE:

```
public class Q6 implements Myinterface3{
        @Override
        public void sum(int a, int b, int c) {
        System.out.println("This is the sum of abstract method:"+(a+b+c));
        }

        @Override
        public void sum1(int a, int b) {
        System.out.println("This is the sum of default "+(a+b));
        }

        public static void main(String[] args) {
        Q6 ques= new Q6();
        ques.sum(4,5,8);
        ques.sum1(4,9);
        Myinterface3.show();
        }
}
interface Myinterface3
{
        public void sum(int a,int b,int c);
        default void sum1(int a,int b){
        System.out.println("The sum : "+(a+b));
        }
        static void show()
        {
        System.out.println("Static Method Executed");
        }
}
```

OUTPUT:



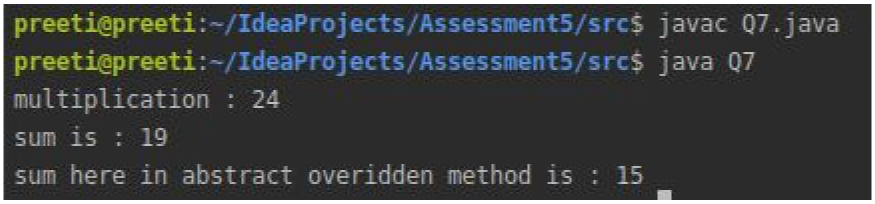
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q6.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q6
This is the sum of abstract method:17
This is the sum of default 13
Static Method Executed
```

7. **Override the default method of the interface.**

CODE:

```
public class Q7 implements Myinterface1 {
    public void sum(int x,int y){
    System.out.println("sum here in abstract overidden method is : "+(x+y));
    }

    @Override
    public void method(int a, int b, int c) {
    int mul=a*b*c;
    System.out.println("multiplication : "+mul);
    Myinterface1.super.method(4,7,8);

    }

    public static void main(String[] args) {
    Q7 ques=new Q7();
    ques.method(2,3,4);
    ques.sum(6,9);

    }

}
interface Myinterface1{
    public void sum(int a, int b);
    default void method(int a,int b,int c){
    System.out.println("sum is : "+(a+b+c));
    }
}
```

OUTPUT:



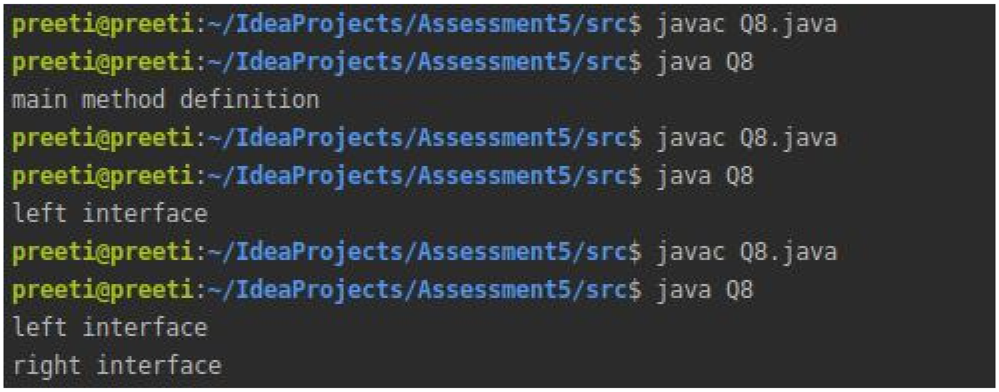
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q7.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q7
multiplication : 24
sum is : 19
sum here in abstract overidden method is : 15
```

8. **Implement multiple inheritance with default method inside interface.**

CODE:

```
interface Left
{
            default void m1()
            {
            System.out.println("left interface");
            }
}
interface Right
{
            default void m1()
            {
            System.out.println("right interface");
            }
}
public class Q8 implements Left,Right
{
            public void m1() {
//          System.out.println("main method definition");

            Left.super.m1();
            Right.super.m1();
            }
            public static void main(String[] args)
            {
            Q8 q=new Q8();
            q.m1();
            }
}
```

OUTPUT:



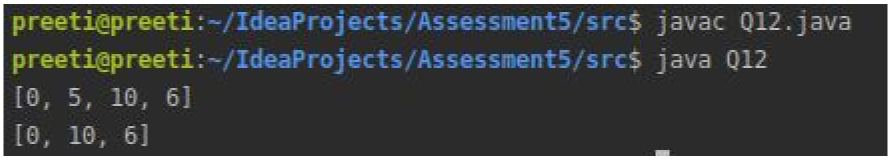
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q8.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q8
main method definition
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q8.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q8
left interface
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q8.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q8
left interface
right interface
```

9. **Collect all the even numbers from an integer list.**

CODE:

```
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public class Q12 {
            public static void main(String[] args) {
//          List<Integer> l= Arrays.asList(1,2,3,4,5,6);

 ArrayList<Integer> l=new ArrayList<Integer>();
 l.add(0);
 l.add(5);
 l.add(10);
 l.add(6);
            System.out.println(l);
            List<Integer> l1=l.stream().filter(l->l%2==0).collect(Collectors.toList());
            System.out.println(l1);

            }
}
```

OUTPUT:



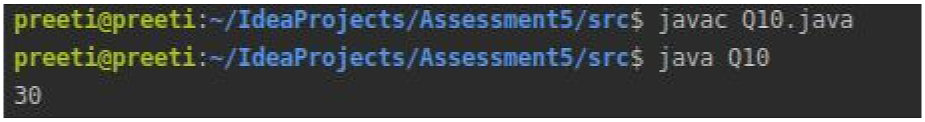
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q12.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q12
[0, 5, 10, 6]
[0, 10, 6]
```

10. Sum all the numbers greater than 5 in the integer list.

CODE:

```
import java.util.Arrays;
import java.util.List;

public class Q10 {
        public static void main(String[] args) {
        List<Integer> list= Arrays.asList(1,2,3,4,5,6,7,8,9);
        System.out.println(
        list
                .stream()
                .filter(e->e>5)
                .mapToInt(e->e)
                .sum()
        );
        }
}
```

OUTPUT:



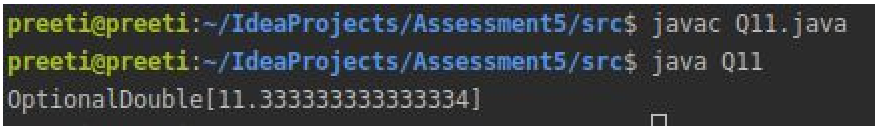
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q10.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q10
30
```

11. Find average of the number inside integer list after doubling it.

CODE:

```
import java.util.Arrays;
import java.util.List;

public class Q11{
        public static void main(String[] args) {
        List<Integer>list=Arrays.asList(1,2,3,4,5,7,9,6);
        System.out.println(
        list
                .stream()
                .filter(e->e>2)
                .map(e->e*2)
                .mapToDouble(e->e)
                .average()
        );

        }
}
```

OUTPUT:



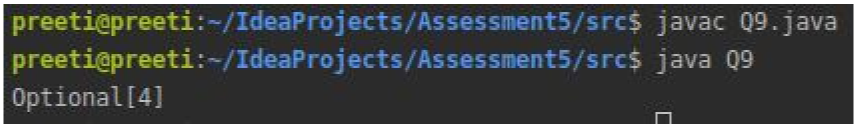
```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q11.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q11
OptionalDouble[11.333333333333334]
```

12. Find the first even number in the integer list which is greater than 3.

CODE:

```
import java.util.Arrays;
import java.util.List;

public class Q9 {
        public static void main(String[] args) {
        List<Integer> list= Arrays.asList(1,2,3,4,5,6,7,8);
        System.out.println(list
                .stream()
                .filter(e-> e>3 && e%2==0)
                .findFirst()
        );
        }
}
```

OUTPUT:



```
preeti@preeti:~/IdeaProjects/Assessment5/src$ javac Q9.java
preeti@preeti:~/IdeaProjects/Assessment5/src$ java Q9
Optional[4]
```