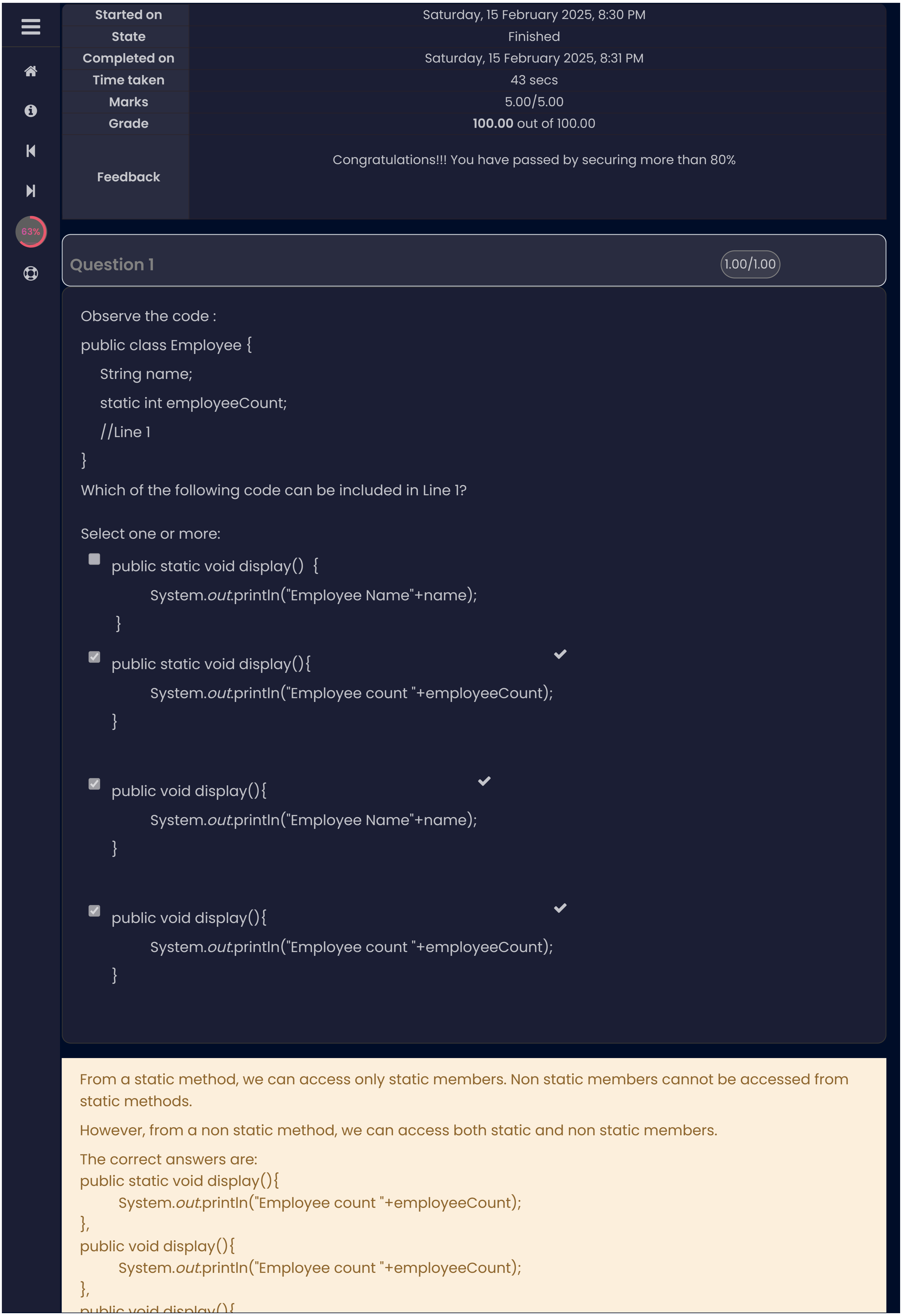


Started on	Saturday, 15 February 2025, 8:30 PM
State	Finished
Completed on	Saturday, 15 February 2025, 8:31 PM
Time taken	43 secs
Marks	5.00/5.00
Grade	**100.00** out of 100.00
Feedback	Congratulations!!! You have passed by securing more than 80%

Question 1

1.00/1.00

Observe the code :

```
public class Employee {
    String name;
    static int employeeCount;
    //Line 1
}
```

Which of the following code can be included in Line 1?

Select one or more:

□
```
public static void display()  {
        System.out.println("Employee Name"+name);
 }
```

☑ ✓
```
public static void display(){
        System.out.println("Employee count "+employeeCount);
}
```

☑ ✓
```
public void display(){
        System.out.println("Employee Name"+name);
}
```

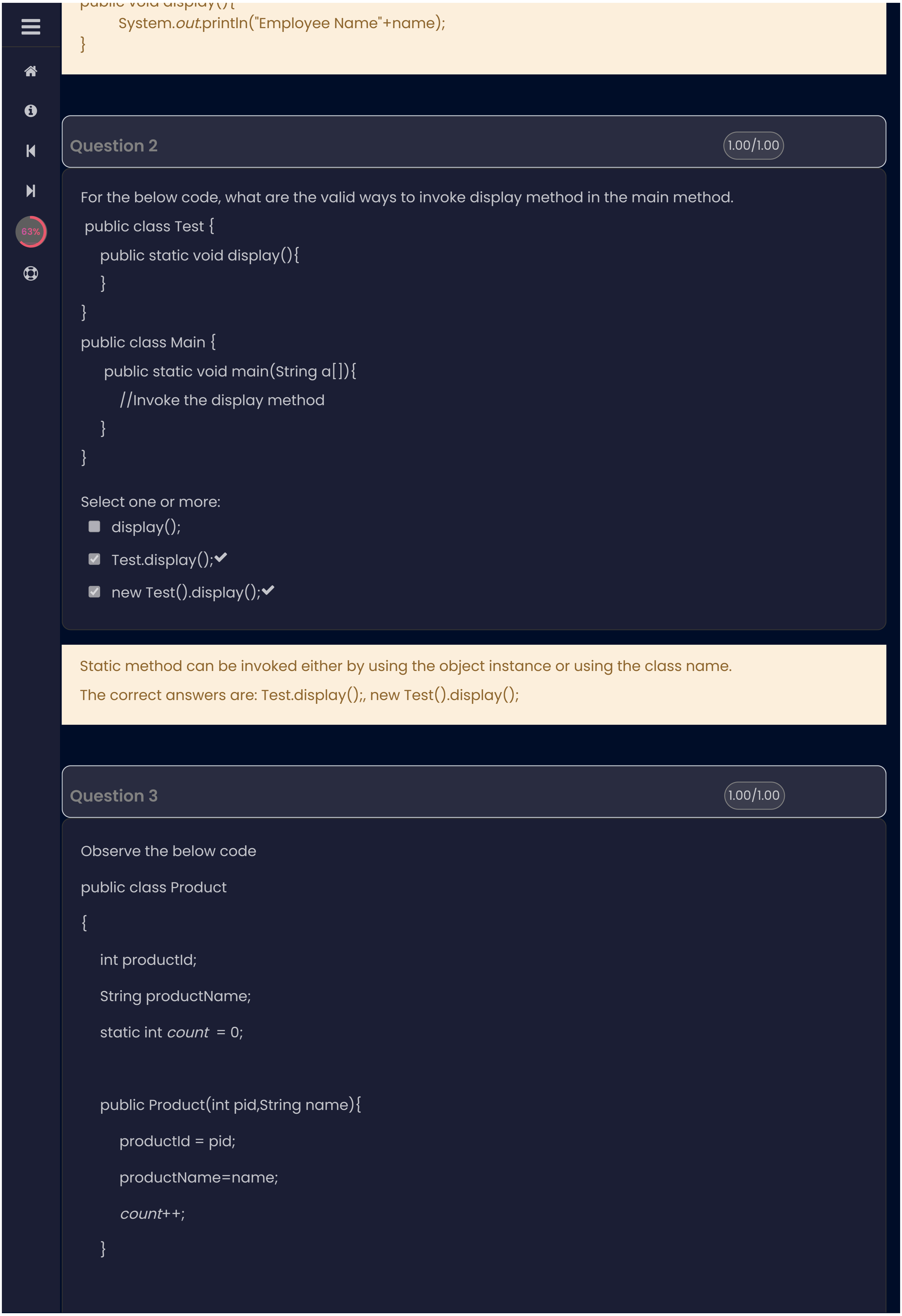
☑ ✓
```
public void display(){
        System.out.println("Employee count "+employeeCount);
}
```

From a static method, we can access only static members. Non static members cannot be accessed from static methods.

However, from a non static method, we can access both static and non static members.

The correct answers are:

```
public static void display(){
        System.out.println("Employee count "+employeeCount);
},
public void display(){
        System.out.println("Employee count "+employeeCount);
},
public void display(){
```



```
public void display(){
        System.out.println("Employee Name"+name);
}
```

Question 2 1.00/1.00

For the below code, what are the valid ways to invoke display method in the main method.

```
 public class Test {
    public static void display(){
    }
}
public class Main {
     public static void main(String a[]){
        //Invoke the display method
    }
}
```

Select one or more:

- □ display();
- ☑ Test.display();✓
- ☑ new Test().display();✓

Static method can be invoked either by using the object instance or using the class name.

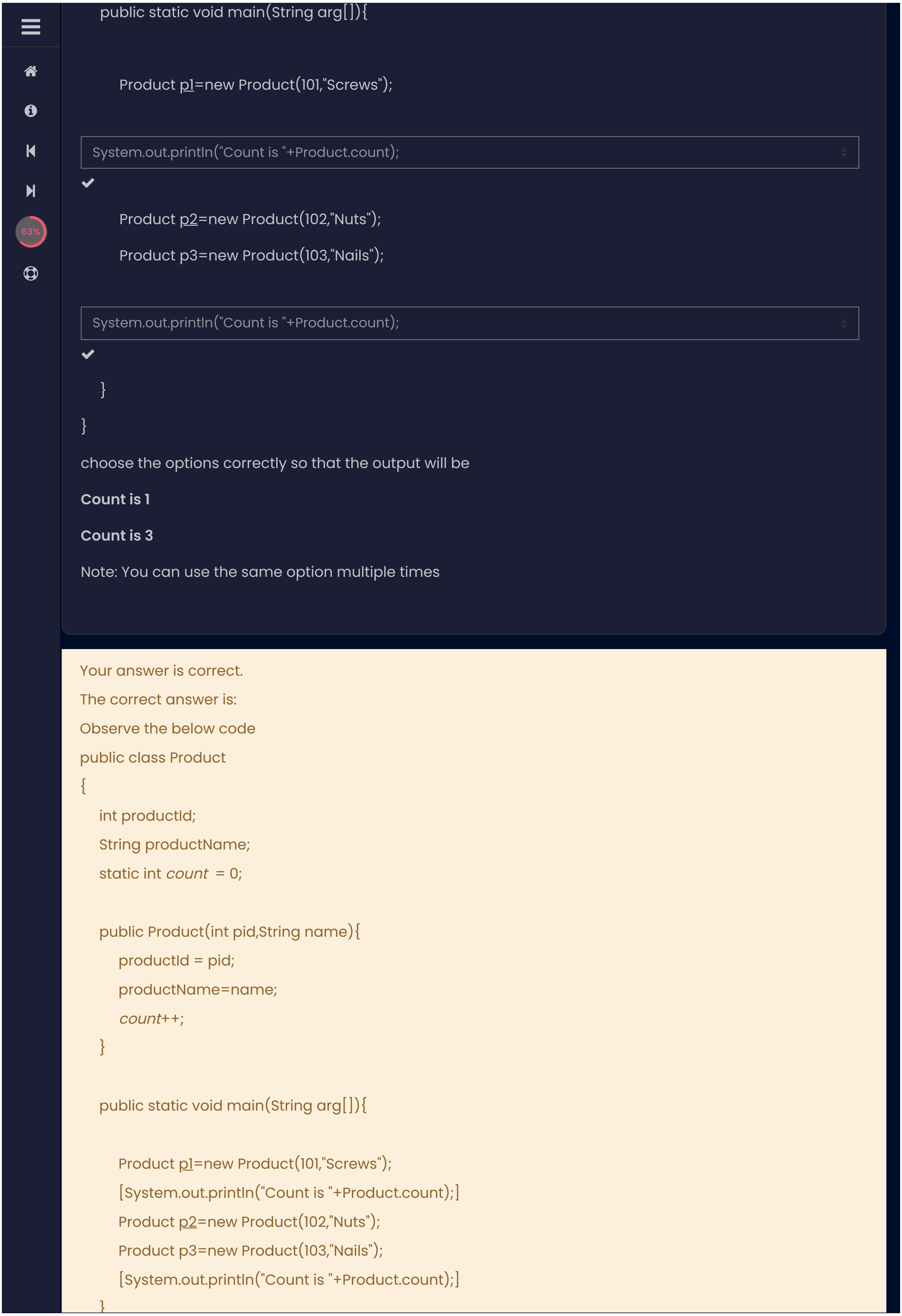
The correct answers are: Test.display();, new Test().display();

Question 3 1.00/1.00

Observe the below code

```
public class Product
{
    int productId;
    String productName;
    static int count  = 0;

    public Product(int pid,String name){
        productId = pid;
        productName=name;
        count++;
    }
```



```
    public static void main(String arg[]){

        Product p1=new Product(101,"Screws");
```

System.out.println("Count is "+Product.count); ✓

```
        Product p2=new Product(102,"Nuts");
        Product p3=new Product(103,"Nails");
```

System.out.println("Count is "+Product.count); ✓

```
    }
}
```

choose the options correctly so that the output will be

Count is 1

Count is 3

Note: You can use the same option multiple times

Your answer is correct.

The correct answer is:

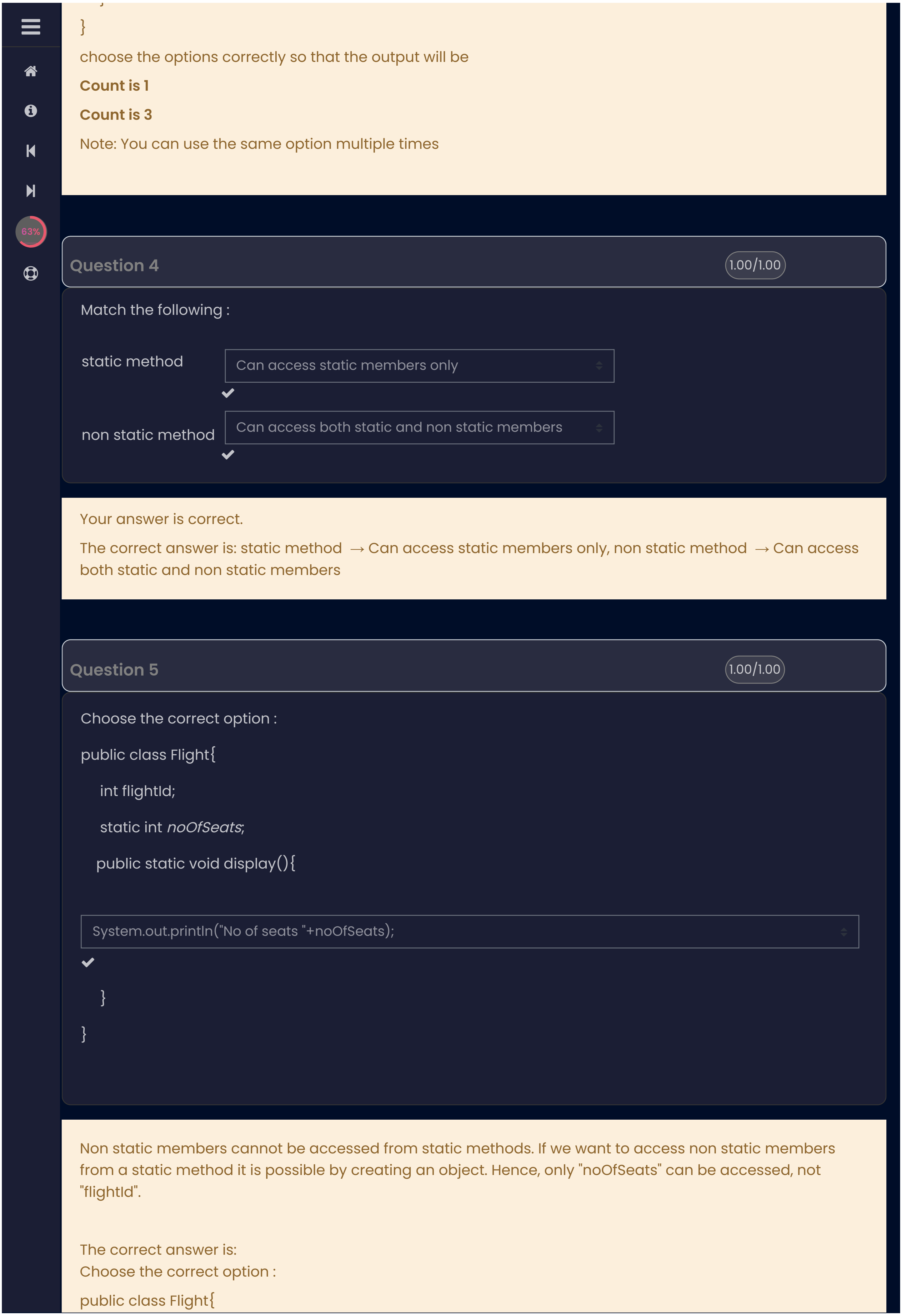
Observe the below code

```
public class Product
{
    int productId;
    String productName;
    static int count  = 0;

    public Product(int pid,String name){
        productId = pid;
        productName=name;
        count++;
    }

    public static void main(String arg[]){

        Product p1=new Product(101,"Screws");
        [System.out.println("Count is "+Product.count);]
        Product p2=new Product(102,"Nuts");
        Product p3=new Product(103,"Nails");
        [System.out.println("Count is "+Product.count);]
    }
```



}

choose the options correctly so that the output will be

Count is 1

Count is 3

Note: You can use the same option multiple times

Question 4

1.00/1.00

Match the following :

static method — Can access static members only ✓

non static method — Can access both static and non static members ✓

Your answer is correct.

The correct answer is: static method → Can access static members only, non static method → Can access both static and non static members

Question 5

1.00/1.00

Choose the correct option :

```
public class Flight{
    int flightId;
    static int noOfSeats;
   public static void display(){

System.out.println("No of seats "+noOfSeats);
✓
    }
}
```

Non static members cannot be accessed from static methods. If we want to access non static members from a static method it is possible by creating an object. Hence, only "noOfSeats" can be accessed, not "flightId".

The correct answer is:
Choose the correct option :
public class Flight{

```
    int flightId;
    static int noOfSeats;
   public static void display(){
        [System.out.println("No of seats "+noOfSeats);]
    }
}
```