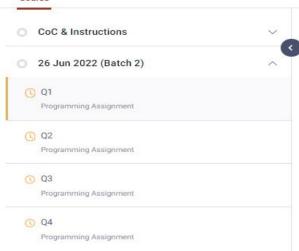
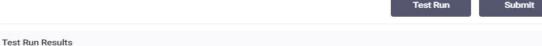


Actual Output



This assignment has public test cases. Please click on "Test Run" button to see the status of public test cases. Assignment will be evaluated only after submitting using "Submit" button below. If you only test run the program, your assignment will not be graded and you will not see your score after the deadline.





Summary: TravellerTest.java:16: error: missing return statement } => 1 error

Public Test: 0/0 Passed

Download All &

Test Case 1

Input	Expected Output		
Rocky Agra Goa	Rocky:[Agra, Goa]		
Shafi Jaipur	Shafi:[Agra, Goa, Jaipur]		

Test Case 2

Rohit Mussoorie

Input **Expected Output Actual Output** Virat Kerala Varanasi Virat:[Kerala, Varanasi]

Rohit: [Kerala, Varanasi, Mussoorie]

```
TravellerTest.java
Java
                                   V
AA + - % 6 0
 1 - import java.util.*;
 2 - class Tourist{
        String tName;
        ArrayList<String> tPlaces;
 4
        public Tourist(String vName, ArrayList<String> vPlaces) {
            this.tName = vName:
 6
            this.tPlaces = vPlaces;
 8
        public Tourist(Tourist t){
 9+
           // copy the values as shown in the test cases to create a new object
10
11
12 -
        public String toString(){
13
          // Override the toString() method
14
15 }
16 - public class TravellerTest{
      public static void main(String[] args){
17 -
        Scanner sc = new Scanner(System.in);
18
        ArrayList<String> vp = new ArrayList<String>();
19
        String n = sc.next();
20
21
        vp.add(sc.next());
        vp.add(sc.next());
22
23
        Tourist t1 = new Tourist(n, vp);
24
        Tourist t2 = new Tourist(t1);
25
        t2.tName = sc.next();
        t2.tPlaces.add(sc.next());
26
27
        System.out.println(t1 + "\n" + t2);
        sc.close();
28
29
```

30

```
TravellerTest.java
Java
AA + - % 0 0
 1 import java.util.*;
 2 * class Tourist{
        String tName;
 4
        ArrayList<String> tPlaces;
 5 -
        public Tourist(String vName, ArrayList<String> vPlaces) {
            this.tName = vName;
 6
            this.tPlaces = vPlaces;
 8
 9 +
        public Tourist(Tourist t){
           // copy the values as shown in the test cases to create a new object
10
11
           this.tName=t.tName;
           ArrayList<String> vp1 = new ArrayList<String>();
12
13 -
           for(String s: t.tPlaces){
14
            vp1.add(s);
15
16
        this.tPlaces=vp1;}
        public String toString(){
17 -
         // Override the toString() method
18
          return(tName + ":" + tPlaces);
19
20
21 }
22 * public class TravellerTest{
      public static void main(String[] args){
23 -
24
        Scanner sc = new Scanner(System.in);
        ArrayList<String> vp = new ArrayList<String>();
25
        String n = sc.next();
26
27
        vp.add(sc.next());
28
        vp.add(sc.next());
        Tourist t1 = new Tourist(n, vp);
29
30
        Tourist t2 = new Tourist(t1);
        t2.tName = sc.next();
31
        t2.tPlaces.add(sc.next());
32
33
        System.out.println(t1 + "\n" + t2);
        sc.close();
34
35
36
```

- (Q1 Programming Assignment
- (Q2 Programming Assignment
- (I) Q3 Programming Assignment
- (Q4 Programming Assignment

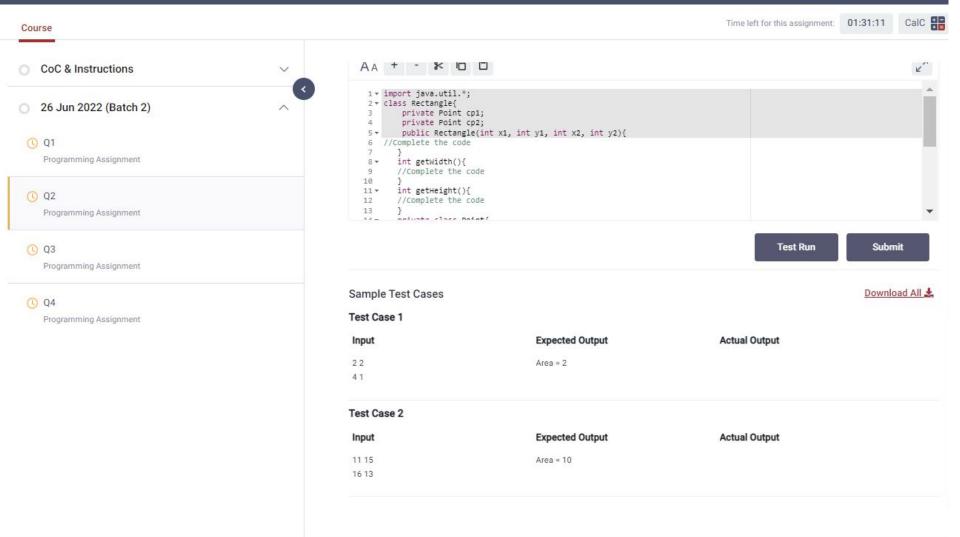
You will be able to resubmit before due date.

Due: 26 Jun 2022 15:30 IST Time Left: 01:31:18

Complete the Java program to compute the area of a rectangle. The inputs for the program are the x and y coordinates of the points denoting the top-left corner and that of the bottom-right corner of the rectangle.

- · Class Rectangle has/should have the following members:
- Two instance variables cp1 and cp2 of type Point, where Point is a private class inside Class Rectangle.
- Constructor Rectangle (int, int, int, int), where the arguments are in the order: x-coordinate and ycoordinate of one corner, x-coordinate and y-coordinate of opposite corner of a rectangle.
- -- Methods int getWidth() and int getHeight() compute the width and the height of the rectangle respectively. These should compute the absolute values (if needed, use method from Math class).
- Class Point is an inner class with two instance variables x and y, and a constructor to initialize them.
- · Class PytClass has the main method that reads in all the four coordinates in the following order: x-coordinate and ycoordinate of one corner, x-coordinate and y- coordinate of opposite corner of a rectangle. It then creates a new Rectangle object using these coordinates, and finds its area.

lava documentation: https://doce.orgolo.com/on/java/javaco/11/doce/api/indox.html



```
PvtClass.java
Java
AA + - % 🗇 🗅
 1 - import java.util.*;
 2 - class Rectangle{
        private Point cp1;
        private Point cp2;
        public Rectangle(int x1, int y1, int x2, int y2){
   //Complete the code
       int getWidth(){
 8+
       //Complete the code
10
11 -
       int getHeight(){
       //Complete the code
12
13
14 -
       private class Point{
       //Complete the code
15
16
17 }
18 - public class PvtClass{
19 -
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
20
            int x1 = sc.nextInt();
21
            int y1 = sc.nextInt();
22
            int x2 = sc.nextInt();
23
            int y2 = sc.nextInt();
24
            Rectangle r = new Rectangle(x1, y1, x2, y2);
25
            System.out.println("Area = " + r.getHeight() * r.getWidth());
26
            sc.close();
27
28
```

29 }

```
Java
                                       PvtClass.java
AA + - % 6 6
 1 import java.util.*;
 2 class Rectangle{
        private Point cp1;
        private Point cp2;
        public Rectangle(int x1, int y1, int x2, int y2){
 6 //Complete the code
        cp1=new Point(x1,y1);
        cp2=new Point(x2,y2);
      int getWidth(){
10-
      //Complete the code
11
       return Math.abs(cp2.x-cp1.x);
12
13
14-
      int getHeight(){
15
      //Complete the code
       return Math.abs(cp2.y-cp1.y);
16
17
18-
       private class Point{
19
          int x;
          int y;
20
          public Point(int x1,int y1){
21-
22
              x=x1;
23
              y=y1;
24
25
       //Complete the code
26
27 }
28 public class PvtClass{
29 -
        public static void main(String[] args) {
30
            Scanner sc = new Scanner(System.in);
31
            int x1 = sc.nextInt();
32
            int y1 = sc.nextInt();
33
            int x2 = sc.nextInt();
34
            int y2 = sc.nextInt();
35
            Rectangle r = new Rectangle(x1, y1, x2, y2);
36
            System.out.println("Area = " + r.getHeight() * r.getWidth());
37
            sc.close();
38
```

39 }



(Q1 Programming Assignment

(Q2 Programming Assignment

(Q3 Programming Assignment

(Q4 Programming Assignment You will be able to resubmit before due date.

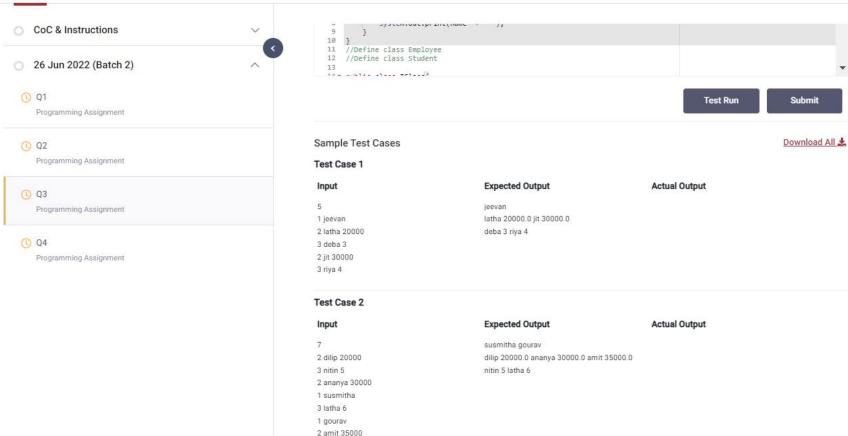
Due: 26 Jun 2022 15:30 IST Time Left: 01:30:54

Complete the Java program that defines three classes Person, Employee and Student. The classes Employee and student inherit from Person. The input to the program are the number of persons, followed by an integer indicating the type of person (1 for Person, 2 for Employee and 3 for Student), name and a number (if type is Employee or Student) that denotes salary for an Employee or semester for a Student. Your program should implement a method printall that is capable of printing a list of objects each of which may be of type Person, Employee Of Student.

- · Class Person has been defined.
- · Define class Employee which has an additional instance variable (as compared to class Person) salary of type double, and overrides the method show of class Person. Write the appropriate constructor for the same.
- · Define class student which has an additional instance variable (as compared to class Person) semester of type int, and overrides the method show of class Person. Write the appropriate constructor for the same.
- · Class TClass has two methods main and printAll. Complete the definition of method printAll that accepts a list of the following types only:
- List<Person>, List<Employee>, and List<Student> and prints the objects using method show.

Java documentation: https://docs.oracle.com/en/java/javase/11/docs/api/jndex.html





44

```
private String name;
       public Person(String n){
           name = n;
       public void show(){
 8
           System.out.print(name + " ");
 9
10 }
11 //Define class Employee
12 //Define class Student
13
14 - public class TClass{
15
       //Define method printAll
       public static void main(String[] args){
16+
           Scanner sc = new Scanner(System.in);
17
           List<Person> lp = new ArrayList<Person>();
18
           List<Employee> le = new ArrayList<Employee>();
19
           List<Student> ls = new ArrayList<Student>();
20
21
           int n = sc.nextInt(); //number of inputs
22
           for(int i = 0; i < n; i++){
23 -
               int t = sc.nextInt(); //type of person
24
               String s1 = sc.next(); //name
25
               if(t == 2){
26 -
                   double s2 = sc.nextDouble(); //salary
27
                   le.add(new Employee(s1, s2));
28
29
30 -
               else if(t == 3){
                   int s3 = sc.nextInt(); //semester
31
                   ls.add(new Student(s1, s3));
32
33
               else {
34 -
                   lp.add(new Person(s1));
35
36
37
           printAll(lp);
38
39
           System.out.println();
           printAll(le);
40
           System.out.println();
41
           printAll(ls);
42
43
```



O 26 Jun 2022 (Batch 2)

(L) Q1 Programming Assignment

(I) Q2 Programming Assignment

Programming Assignment

(I) Q3

(Q4 Programming Assignment You will be able to resubmit before due date.

Due: 26 Jun 2022 15:30 IST

Time Left: 01:30:25

Complete the Java program that takes as input the names of four customers and the number of items bought by each of them, and prints the name of the customer who has bought the maximum number of items.

- · Interface Iterator has the following members:
- Two abstract methods: has next() and get next()
- Two instance variables: String custName and int numItems
- A constructor to initialize the instance variables

· Class Customer has the following members:

- Accessor methods for the instance variables
- Overridden toString method to display the customer name.
- · Class CustomerList has the following members:
- An instance variable: Customer[] cArr
- A constructor to initialize the instance variable
- Method public Iterator getIterator() which returns an inner class object.
- Private inner class CustIterator which implements the Iterator interface and has the following members.
 - * One instance variable: private int index

(L) Q2

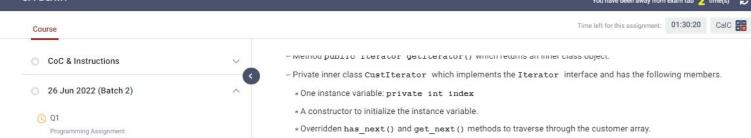
(I) Q3

(I) Q4

Programming Assignment

Programming Assignment

Programming Assignment



- · Class IteratorTest has the following methods.
- The main method takes the name and the number of items bought by each customer, as input.
- It also has method getMaxCustomer (CustomerList cList) to find the customer who has bought the maximum number of items. For ease of implementation, you may assume that there is exactly one customer who has bought the maximum number of items.

Java documentation:https://docs.oracle.com/en/java/javase/11/docs/api/index.html

This assignment has public test cases. Please click on "Test Run" button to see the status of public test cases. Assignment will be evaluated only after submitting using "Submit" button below. If you only test run the program, your assignment will not be graded and you will not see your score after the deadline. Choose Language

Download All &



CoC & Instructions

26 Jun 2022 (Batch 2)

(L) Q1 Programming Assignment

(L) Q2 Programming Assignment

(I) Q3 Programming Assignment

(I) Q4 Programming Assignment



Sample Test Cases

Test Case 1

^

Input **Expected Output Actual Output** Priya 12 Reshma

Reshma 13 Vignesh 4

Vidya 6

Test Case 2

Sukesh 1151 Mukesh 1152

Input **Expected Output Actual Output** Deepa 1000 Mukesh Prakash 1150

```
Java
                                                            IteratorTest.java
AA - * 6 0
 1 - import java.util.*;
 2 - interface Iterator{
        public boolean has_next();
        public Object get_next();
 6 - class Customer{
        private String custName;
        private int numItems:
        public Customer(String custName, int num) {
            this.custName = custName;
            this.numItems = num;
        public String getName() {
            return custName;
        public int getNumItems() {
            return numItems;
        public String toString() {
            return custName;
22 }
23 - class CustomerList {
        private Customer[] cArr;
        public CustomerList(Customer[] ca) {
            cArr = ca;
        public Iterator getIterator() {
            return new CustIterator();
        private class CustIterator implements Iterator(
            private int index;
            public CustIterator() {
                index = -1;
            public boolean has_next() {
                if(index < cArr.length - 1)
                   return true;
                return false:
            public Object get_next() {
                index++:
                return cArr[index];
47 - public class IteratorTest {
        public static Customer getMaxCustomer(CustomerList cList) {
         //Complete the definition of this method
        public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            Customer[] cA = new Customer[4];
             for(int i=0;i<4;i++) {
                cA[i] = new Customer(sc.next(), sc.nextInt());
            CustomerList custList = new CustomerList(cA);
            System.out.println(getMaxCustomer(custList));
            sc.close();
```

10

11

12

13 + 14

15 16 -

17

18 19 -

20

21

24

25 *

26

27 28 -

29

30

31 -32

33.

34

35

38 39

40

41 -42

48 +

49 50

51 52 +

53

54 55 *

56

57 58

59 60

61 62 63

36 * 37

```
Java
                                                                          IteratorTest.iava
AA + - % 6 6
  1 - import java.util.*;
  2 - interface Iterator{
          public boolean has next();
          public Object get_next();
  6 - class Customer{
          private String custName;
          private int numItems;
          public Customer(String custName, int num) {
               this.custName = custName;
 18
 11
               this.numItems = num;
          public String getName() {
   return custName;
 13 -
 14
 16 *
          public int getNumItems() {
 17
               return numItems;
 18
19 -
          public String toString() {
 20
21
22
               return custName:
 23 * class CustomerList {
 24 -
          private Customer[] cArr;
          public CustomerList(Customer[] ca) {
 26
27
               cArr = ca;
          public Iterator getIterator() {
   return new CustIterator();
 28 -
 29
 38
 31 -
          private class CustIterator implements Iterator{
 32
33 *
34
               private int index;
               public CustIterator() {
                   index = -1;
              public boolean has_next() {
   if(index < cArr.length - 1)</pre>
                      return true;
                   return false;
 40
               public Object get_next() {
 41 -
 42
43
                   index++:
                   return cArr[index];
 44
 45
46
 47 + public class IteratorTest {
          public static Customer getMaxCustomer(CustomerList cList) {
          //Complete the definition of this method
 51
          Iterator it=cList.getIterator();
 52
53
          int val=0;
          Customer cu=null;
 54
55 •
          Customer cu1=nu11;
          while(it.has_next()){
 56
57 •
                cu1=(Customer)it.get_next();
               if(cu1.getNumItems()>val){
   val=cu1.getNumItems();
 58
 59
                    cu=cu1;
 60
 61
               // System.out.println(cu.getName());
 62
 63
           return cu;
 64
 65 +
          public static void main(String[] args) {
 66
67
68 =
               Scanner sc=new Scanner(System.in);
               Customer[] cA = new Customer[4];
               for(int i=0;i<4;i++) {
                   cA[i] = new Customer(sc.next(), sc.nextInt());
 69
78
71
72
73
74
75
              CustomerList custList = new CustomerList(cA);
System.out.println(getMaxCustomer(custList));
               sc.close();
```