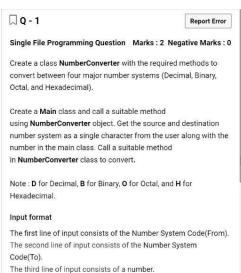
Pause Test

73:57

Submit Test

Test will be auto submitted in 150 minutes from the test start time







JAVA COD Classes and Methods Lab Exercise

The output prints the result after conversion.

Output format

Output format

The output prints the result after conversion.

Test will be auto submitted in 150 minutes from the test start time

CODING 5

```
Q 0 - 1
                                                    Report Error
Single File Programming Question Marks: 2 Negative Marks: 0
Create a class NumberConverter with the required methods to
convert between four major number systems (Decimal, Binary,
Octal, and Hexadecimal).
Create a Main class and call a suitable method
using NumberConverter object. Get the source and destination
number system as a single character from the user along with the
number in the main class. Call a suitable method
in NumberConverter class to convert.
Note: D for Decimal, B for Binary, O for Octal, and H for
Hexadecimal.
Input format
The first line of input consists of the Number System Code(From).
The second line of input consists of the Number System
Code(To).
The third line of input consists of a number.
```

```
Java (11) 🗸 🚳 📠
Fill your code here
                     Scanner sc=new Scanner(System.in);
| char s=sc.nextLine().charAt(0);
| char d=sc.nextLine().charAt(0);
| String input=sc.nextLine();
| ff((s='B')||(s=='b'))
                        =2;
f((d=='B')||(d=='b'))
                     d
                        =2;
f((s=='D')||(s=='D'))
                     s=10;
if((d=='D')||(d=='D'))
                     s=8;
if((d=='0')||(d=='o'))
                     d=8;
if((s=='H')||(s=='h'))
                        =16;
F((d=='H')||(d=='h'))
                      System.out.print(num.converter(input,s,d));
Provide Custom Input
```

Test will be auto submitted in 150 minutes from the test start time





BO Classes

We can use a BO class for computational purposes.

The Stall owners wanted to calculate the total cost of a particular item type for the given timeline. So add a feature in the application to calculate the total cost for the given timeline.

Create a class ItemType with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default, and parameterized constructor.

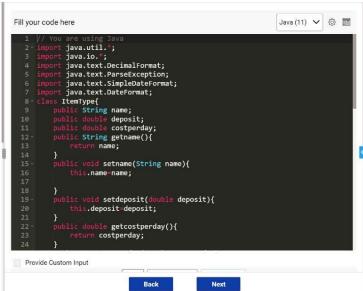
public ItemType(String name, Double deposit, Double costPerDay).

Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType. Write a method calculateCost in ItemTypeBO class.

Method	Method Description
public Double calculateCost(Date	returns a Double which
start Date end.ltemType typelns)	corresponds to the total cost.

Create a driver class Main to test the above classes.

Note: Strictly adhere to the Object-Oriented Specifications given



JAVA_COD_Classes and Methods_Lab Exercise

Submit Test 73:44 Pause Test

Test will be auto submitted in 150 minutes from the test start time



Q-2

Report Error

Single File Programming Question Marks: 2 Negative Marks: 0

BO Classes

We can use a BO class for computational purposes.

The Stall owners wanted to calculate the total cost of a particular item type for the given timeline. So add a feature in the application to calculate the total cost for the given timeline

Create a class ItemType with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default, and parameterized constructor.

public ItemType(String name, Double deposit, Double costPerDay).

Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType. Write a method calculateCost in ItemTypeBO class.

Method	Method Description	
public Double calculateCost(Date start Date end.ltemType typelns)	returns a Double which corresponds to the total cost.	
start Date end item type type ins)	corresponds to the total cos	

Create a driver class Main to test the above classes.

Note: Strictly adhere to the Object-Oriented Specifications given

```
Java (11) 🗸 💮 🛅
Fill your code here
                          voia setaeposit(ao
s.deposit=deposit;
                               le getcostperday(){
  costperday;
                            oid setperday(double costperday)
                      this.costperday=costperday;
                        ItemType(){
                         s.name=null;
s.deposit=0;
                       is.costperday=0;
                        ItemType(String name,double deposit,double costperday){
                          .name=name;
.deposit=deposit;
                    this.costperday=costperday;
System.out.println(this.name
System.out.println(this.depo
System.out.println(this.cost
                                                    his.name);
his.deposit);
Provide Custom Input
                                              Back
                                                                       Next
```

Test will be auto submitted in 150 minutes from the test start time





Single File Programming Question Marks: 2 Negative Marks: 0

BO Classes

We can use a BO class for computational purposes.

The Stall owners wanted to calculate the total cost of a particular item type for the given timeline. So add a feature in the application to calculate the total cost for the given timeline.

Create a class ItemType with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default, and parameterized constructor.

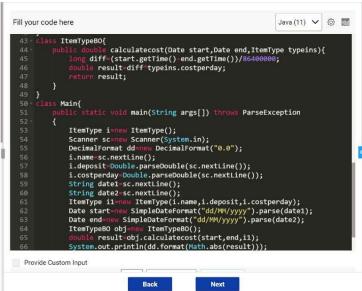
public ItemType(String name, Double deposit, Double costPerDay).

Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType. Write a method calculateCost in ItemTypeBO class.

1	Method	Method Description
	public Double calculateCost(Date start Date end ItemType typeIns)	returns a Double which corresponds to the total cost.

Create a driver class Main to test the above classes.

Note: Strictly adhere to the Object-Oriented Specifications given



JAVA COD Classes and Methods Lab Exercise

Submit Test

Pause Test

73:38

Test will be auto submitted in 150 minutes from the test start time



Q-2

Single File Programming Question Marks: 2 Negative Marks: 0

Report Error

BO Classes

We can use a BO class for computational purposes.

The Stall owners wanted to calculate the total cost of a particular item type for the given timeline. So add a feature in the application to calculate the total cost for the given timeline

Create a class ItemType with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default, and parameterized constructor.

public ItemType(String name, Double deposit, Double costPerDay).

Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType. Write a method calculateCost in ItemTypeBO class.

Method	Method Description
public Double calculateCost(Date	returns a Double which
start Date end ItemType typeIns)	corresponds to the total cos

Create a driver class Main to test the above classes.

Note: Strictly adhere to the Object-Oriented Specifications given

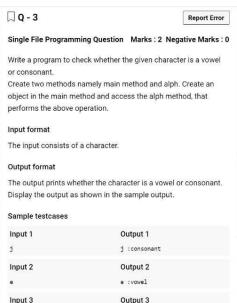
```
Java (11) 🗸 💮 🛅
Fill your code here
                                                                              result:
                                           Main{
                                                                                        void main(String args[]) throws ParseException
                                                 ItemType i=new ItemType();
Scanner sc=new Scanner(System.in);
DecimalFormat dd=new DecimalFormat("0.0");
i.name=sc.nextLine();
i.deposit=Double.parseDouble(sc.nextLine());
i.costperday=Double.parseDouble(sc.nextLine());
String date1-sc.nextLine();
String date2-sc.nextLine();
ItemType i1=new ItemType(i.name,i.deposit,i.costperday);
Date start=new SimpleDateFormat("dd/MM/yyyy").parse(date1);
Date end=new SimpleDateFormat("dd/MM/yyyy").parse(date2);
ItemTypeBO obj=new ItemTypeBO();
double result=obj.calculatecost(start,end,i1);
System.out.println(dd.format(Math.abs(result)));
Provide Custom Input
                                                                                                                     Back
```

Submit Test 72:37

Pause Test

Test will be auto submitted in 150 minutes from the test start time







JAVA_COD_Classes and Methods_Lab Exercise

Test will be auto submitted in 150 minutes from the test start time



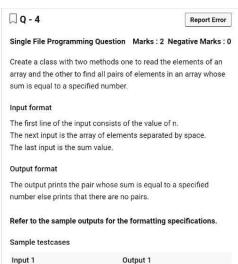
```
Q-4
                                                    Report Error
Single File Programming Question Marks: 2 Negative Marks: 0
Create a class with two methods one to read the elements of an
array and the other to find all pairs of elements in an array whose
sum is equal to a specified number.
The first line of the input consists of the value of n.
The next input is the array of elements separated by space.
The last input is the sum value.
Output format
The output prints the pair whose sum is equal to a specified
number else prints that there are no pairs.
Refer to the sample outputs for the formatting specifications.
Sample testcases
Input 1
                                 Output 1
1 2 3 4 5
                                 4 4
```

Submit Test 72:04

Pause Test

Test will be auto submitted in 150 minutes from the test start time







JAVA_COD_Classes and Methods_Lab Exercise

5 3

1 2 3 4 5

Test will be auto submitted in 150 minutes from the test start time



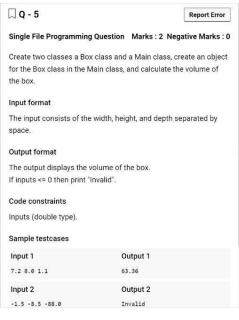
```
Q-5
                                                   Report Error
Single File Programming Question Marks: 2 Negative Marks: 0
Create two classes a Box class and a Main class, create an object
for the Box class in the Main class, and calculate the volume of
the box.
Input format
The input consists of the width, height, and depth separated by
space.
Output format
The output displays the volume of the box.
If inputs <= 0 then print "Invalid".
Code constraints
Inputs (double type).
Sample testcases
Input 1
                                Output 1
7.2 8.0 1.1
                                63.36
                                 Output 2
-1.5 -8.5 -88.0
```

```
Java (11) 🗸 💮 🛅
Fill your code here
                    java.util.;
Box{
                            width;
height;
                             height
depth;
             ass Main{
                                                d main(String args[]){
Box();
                       Box myobj
                                  vol;
                       Goulde Vol;
Scanner scenew Scanner(System.in);
myobj.width=sc.nextDouble();
myobj.height:sc.nextDouble();
myobj.depth=sc.nextDouble();
if(myobj.width>0 88 myobj.height>0 88 myobj.depth>0)
                              vol=myobj.width myobj.height myobj.depth;
System.out.printf("%.2f",vol);
                              System.out.println("Invalid");
Provide Custom Input
```



Test will be auto submitted in 150 minutes from the test start time





```
Fill your code here
                                                                                                                          Java (11) 🗸 🔞 🔳
                  double height;
double depth;
         }
class Main{
                                                    d main(String args[]){
Box();
                         lic static void main(String args[]){
Box myobj=new Box();
double vol;
Scanner sc=new Scanner(System.in);
myobj.width=sc.nextDouble();
myobj.height=sc.nextDouble();
myobj.depth=sc.nextDouble();
if(myobj.width>0 88 myobj.height=0 88 myobj.depth=0)
                                 vol-myobj.width myobj.height myobj.depth;
System.out.printf("%.2f",vol);
                                 System.out.println("Invalid");
Provide Custom Input
                                                  Clear Compile & Run Submit Code
                                                                         Back
```