IRC_SKCT_Java2_SB_COD_Classes&Objects

Test Summary

- No. of Sections: 1
- No. of Questions: 5
- Total Duration: 120 min

Section 1 - Coding

Section Summary

- No. of Questions: 5
- Duration: 120 min

Additional Instructions:

None

Q1. Java program to find the count of all digits of a number using class.
In this program, we will read a positive integer number and then calculate the count of all digits using a class.

Input Format

The input consists of a number.

Output Format

The output prints the count of all digits in the number.

Sample Input

```
12345
```

Sample Output

```
Count of all digits: 5
```

Sample Input

```
22
```

Sample Output

```
Count of all digits: 2
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. Write a program to find the number of occurrences of a character in a string. Create a constructor with two parameters, pass the value from the main method to the constructor, and perform the mentioned task in the constructor and display it.

Input Format

Input to get a string in the first line and a character in the second line.

Output Format

Output the number of occurrences of a character in a string.

Sample Input

```
utter
t
```

Sample Output

```
2
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. Create class money with two attributes:
int rupee
int paisa
Include getters, setters, and constructors.
Create the main class and initialize the values for the data members
Get two amounts and print their sum.

Input Format

The input consists of two amounts.
Rupee and Paisa are separated by a space.

Output Format

The output prints the total sum.
Refer sample input and output for formatting specifications.

Sample Input

```
50 85
42 65
```

Sample Output

```
93.50
```

Sample Input

```
254 45
845 20
```

Sample Output

```
1099.65
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. **A Multiplication Game**

John and Michael play the game of multiplication by multiplying an integer **p** by one of the numbers 2 to 9. John always starts with **p = 1** and multiply it by 1, and passes the result to Michael. Then, Michael multiplies the number by 2 and sends the result to John, then John multiplies by 3 and so on. Before a game starts, they draw an integer **N** and the winner is the one who first reaches ***p ≥ n***.
Create a class that has two functions:
1) A function to perform the multiplication operation
2) The main()

Input Format

The input consists of the value of n.

Output Format

The output prints the n value and who won the game separated by a space.
Refer the sample output for formatting specifications.

Sample Input

```
10
```

Sample Output

```
10 Michael wins
```

Sample Input

```
100
```

Sample Output

```
100 John wins
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. **BO Classes**

We can use a BO class for computational purposes.
The Stall owners wanted to calculate the total cost of a particular ItemType 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,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 in the problem statement.
All class names, attribute names and method names should be the same as specified in the problem statement.
Display only 1 digit after decimal while displaying cost.

Input date format is **dd/MM/yyyy.**

Input Format

First line of the input consists of a string
Second and third line of the input consists of double.
Fourth and fifth line consists of starting date and the ending date.

Output Format

Refer sample output.

Sample Input

```
Morsh
1000.00
50.00
12/10/2018
```

Sample Output

```
Morsh
1000.0
50.0
100.0
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
2147483647
```

Output

```
Count of all digits: 10
```

Weightage - 15

Input

```
222222888
```

Output

```
Count of all digits: 9
```

Weightage - 15

Input

```
77777777
```

Output

```
Count of all digits: 8
```

Weightage - 15

Input

```
1234567
```

Output

```
Count of all digits: 7
```

Weightage - 15

Input

```
654321
```

Output

```
Count of all digits: 6
```

Weightage - 15

Input

```
3214
```

Output

```
Count of all digits: 4
```

Weightage - 15

Input

```
12345
```

Output

```
Count of all digits: 5
```

Weightage - 10

Sample Input

```
12345
```

Sample Output

```
Count of all digits: 5
```

Sample Input

```
22
```

Sample Output

```
Count of all digits: 2
```

Solution

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        int count = 0, num;
        Scanner s = new Scanner(System.in);
        num = s.nextInt();
        while(num != 0)
        {
            num /= 10;
            ++count;
        }
        System.out.println("Count of all digits: " + count);
    }
}
```

Q2

Test Case

Input

```
nuts
s
```

Output

```
1
```

Weightage - 20

Input

```
range
u
```

Output

```
0
```

Weightage - 30

Input

```
kettle
e
```

Output

```
2
```

Weightage - 20

Input

```
runner
n
```

Output

```
2
```

Weightage - 20

Input

```
meets
e
```

Output

```
2
```

Weightage - 10

Sample Input

```
utter
t
```

Sample Output

```
2
```

Solution

```
 import java.util.Scanner;
class Main{
Main(String s,char c){
int res = 0;
for (int i=0; i<s.length(); i++) {
if (s.charAt(i) == c)
res++;
}
 System.out.print(res);

}
public static void main(String args[])
{

String str;

Scanner in=new Scanner(System.in);

str=in.nextLine();

char c;

c=in.next().charAt(0);

Main obj=new Main(str,c);

}
}
```

Q3

Test Case

Input

```
854  96
2486  96
```

Output

```
3341.92
```

Weightage - 25

Input

```
8642 25
6428 60
```

Output

```
15070.85
```

Weightage - 25

Input

```
753 65
854 80
```

Output

```
1608.45
```

Weightage - 25

Input

```
8564 25
24687 20
```

Output

```
33251.45
```

Weightage - 25

Sample Input

```
50 85
42 65
```

Sample Output

```
93.50
```

Sample Input

```
254 45
845 20
```

Sample Output

```
1099.65
```

Solution

```
import java.io.*;
import java.util.*;
class money {
    int rupee;
    int paisa;
    public void setRupee(int r) {
        this.rupee = r;
    }
    public void setPaisa(int p) {
        this.paisa = p;
    }
    public int getRupee() {
        return rupee;
    }
    public int getPaisa() {
        return paisa;
    }
}
class Main {
```

```
    public static void main(String [] args) {
        Scanner sc = new Scanner(System.in);
        money [] m = new money[2];
        int i;
        for(i=0;i<2;i++) {
            m[i] = new money();
            m[i].setRupee(sc.nextInt());
            m[i].setPaisa(sc.nextInt());
        }
        int r,p;
        r = m[0].getRupee()+m[1].getRupee();
        p = m[0].getPaisa()+m[1].getPaisa();
        if(p>99) {
            r +=1;
            p = p-100;
        }
        System.out.println(r+"."+p);
    }
}
```

Q4

Test Case

Input

```
3000
```

Output

```
3000 John wins
```

Weightage - 20

Input

```
5550
```

Output

```
5550 Michael wins
```

Weightage - 20

Input

```
40500
```

Output

```
40500 John wins
```

Weightage - 20

Input

```
750
```

Output

```
750 John wins
```

Weightage - 20

Input

```
200
```

Output

```
200 Michael wins
```

Weightage - 20

Sample Input

```
10
```

Sample Output

```
10 Michael wins
```

Sample Input

```
100
```

Sample Output

```
100 John wins
```

Solution

```
import java.io.*;
import java.util.*;
class multiplicationGame {
public static void game(int n) {
    int sum=1,i=2,count=1;
    while(sum<n && i<=9) {
            sum *= i;
            if(sum <n) {
            i++;
            count++;
            }
            else {
                break;
            }
}
if(count%2 !=0) {
    System.out.println(n+" Michael wins");
}
else {
    System.out.println(n+" John wins");
}
}
public static void main (String [] args) {
    int n;
    Scanner sc = new Scanner(System.in);
        n= sc.nextInt();
        game(n);
 }
}
```

Q5

Test Case

Input

```
Morsh
1000.00
50.00
12/10/2018
```

Output

```
Morsh
1000.0
50.0
100.0
```

Weightage - 20

Input

Output

Ankit 2000.00 35.00 08/08/2008	Ankit 2000.0 35.0 350.0

Weightage - 20

Input	Output
Sharma 8000.00 60.00 22/11/1997	Sharma 8000.0 60.0 240.0

Weightage - 20

Input	Output
Williams 25000.00 70.00 01/01/2001	Williams 25000.0 70.0 2100.0

Weightage - 20

Input	Output
Lora 50000.00 80.00 14/02/2004	Lora 50000.0 80.0 560.0

Weightage - 20

Sample Input	Sample Output
Morsh 1000.00 50.00 12/10/2018	Morsh 1000.0 50.0 100.0

Solution

```
import java.io.*;
import java.text.DecimalFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;
class ItemType {
    public String name;
    public double deposit;
    public double costPerDay;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getDeposit() {
        return deposit;
    }
    public void setDeposit(double deposit) {
        this.deposit = deposit;
    }
    public double getCostPerDay() {
```

```
        return costPerDay;
    }
    public void setCostPerDay(double costPerDay) {
        this.costPerDay = costPerDay;
    }
    public ItemType()
    {
        this.name=null;
        this.deposit=0;
        this.costPerDay=0;
    }
    public ItemType(String name, Double deposit, Double costPerDay){
        this.name=name;
        this.deposit=deposit;
        this.costPerDay=costPerDay;
        System.out.println(this.name);
        System.out.println(this.deposit);
        System.out.println(this.costPerDay);
    }
}
class ItemTypeBO {
    public Double calculateCost(Date start,Date end,ItemType typeIns){
        long diff = (start.getTime()-end.getTime())/86400000;
        double result = diff*typeIns.costPerDay;
        return result;
    }
}
class Main {
    public static 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 iBO = new ItemTypeBO();
double result=iBO.calculateCost(start, end, i1);
System.out.println(dd.format(Math.abs(result)));
    }
}
```