

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-01-Java Architecture, Language Basics](#) / [Lab-01-Logic Building](#)

Status Finished

Started Saturday, 5 October 2024, 7:12 PM

Completed Saturday, 5 October 2024, 8:27 PM

Duration 1 hour 15 mins

Question 1

Incorrect

Marked out of 5.00

Write a program to find whether the given input number is Odd.

If the given number is odd, the program should return 2 else It should return 1.

Note: The number passed to the program can either be negative. positive or zero. Zero should NOT be treated as Odd.

For example:

Input	Result
123	2
456	1

Answer: (penalty regime: 0 %)

```

1 import java.io.*;
2 import java.util.*;
3 calss u
4 {
5     public int isOdd(int input1)
6     {
7         if(input1%2!=0)
8             return 2;
9         else
10            return 1;
11    }
12 }
```

Syntax Error(s)

```

prog.java:3: error: class, interface, enum, or record expected
calss u
^
prog.java:5: error: class, interface, enum, or record expected
public int isOdd(int input1)
^
prog.java:9: error: class, interface, enum, or record expected
    else
    ^
prog.java:11: error: class, interface, enum, or record expected
}
^
4 errors
```

Question 2

Not answered

Marked out of 5.00

Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number.

The last digit should be returned as a positive number.

For example,

if the given number is 197, the last digit is 7

if the given number is -197, the last digit is 7

For example:

Input	Result
197	7
-197	7

Answer: (penalty regime: 0 %)

1 ||

Question 3

Not answered

Marked out of 5.00

Rohit wants to add the last digits of two given numbers.

For example,

If the given numbers are 267 and 154, the output should be 11.

Below is the explanation:

Last digit of the 267 is 7

Last digit of the 154 is 4

Sum of 7 and 4 = 11

Write a program to help Rohit achieve this for any given two numbers.

Note: Tile sign of the input numbers should be ignored.

i.e.

if the input numbers are 267 and 154, the sum of last two digits should be 11

if the input numbers are 267 and -154, the sum of last two digits should be 11

if the input numbers are -267 and 154, the sum of last two digits should be 11

if the input numbers are -267 and -154, the sum of last two digits should be 11

For example:

Input	Result
267 154	11
267 -154	11
-267 154	11
-267 -154	11

Answer: (penalty regime: 0 %)

1 ||

[◀ Lab-01-MCQ](#)

[Is Even? ►](#)

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-02-Flow Control Statements](#) / [Lab-02-Logic Building](#)

Status Finished

Started Sunday, 6 October 2024, 4:36 PM

Completed Sunday, 6 October 2024, 5:25 PM

Duration 49 mins 7 secs

Question 1

Correct

Marked out of 5.00

Consider the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

4th term: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

And so on. Write a program that takes as parameter an integer n and prints the nth terms of this sequence.

Example Input:

1

Output:

1

Example Input:

4

Output:

1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

For example:

Input	Result
1	1
2	1 2 1
3	1 2 1 3 1 2 1
4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class Sample
3 {
4     public static String print(int n)
5     {
6         if(n==1)
7             return "1";
8         else
9         {
10            String rec = print(n-1);
11            return rec+ " "+n+" "+rec;
12        }
13    }
14    public static void main(String args[])
15    {
16        Scanner s = new Scanner(System.in);
17        int n=s.nextInt();
18        System.out.println(print(n));
19    }
20}
21
22

```

	Input	Expected	Got	
✓	1	1	1	✓
✓	2	1 2 1	1 2 1	✓

	Input	Expected	Got	
✓	3	1 2 1 3 1 2 1	1 2 1 3 1 2 1	✓
✓	4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

Consider a sequence of the form 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149...

Write a method program which takes as parameter an integer n and prints the nth term of the above sequence. The nth term will fit in an integer value.

Example Input:

5

Output:

4

Example Input:

8

Output:

24

Example Input:

11

Output:

149

For example:

Input	Result
5	4
8	24
11	149

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class sample
3 {
4     public static int print(int n)
5     {
6         int a=0,b=1,c=1,d=2;
7         if(n==1)
8         {
9             return 0;
10        }
11        else if(n==2)
12        {
13            return 1;
14        }
15        else if(n==3)
16        {
17            return 1;
18        }
19        else if(n==4)
20        {
21            return 2;
22        }
23        else
24        {
25            for(int i=5;i<=n;i++)
26            {
27                a=b;
28                b=c;
29                c=d;
30                d=a+b+c;
31            }
32            return d;
33        }
34    }

```

```
35
36 public static void main(String args[])
37 {
38     Scanner s = new Scanner(System.in);
39     int n=s.nextInt();
40     System.out.println(print(n));
41 }
42 }
```

	Input	Expected	Got	
✓	5	4	4	✓
✓	8	24	24	✓
✓	11	149	149	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

Write a program that takes as parameter an integer n.

You have to print the number of zeros at the end of the factorial of n.

For example, $3! = 6$. The number of zeros are 0. $5! = 120$. The number of zeros at the end are 1.

Note: $n! < 10^5$

Example Input:

3

Output:

0

Example Input:

60

Output:

14

Example Input:

100

Output:

24

Example Input:

1024

Output:

253

For example:

Input	Result
3	0
60	14
100	24
1024	253

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 // Java program to count trailing 0s in n!
2 import java.util.Scanner;
3 public class Fact
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9         System.out.println(ctz(n));
10        sc.close();
11    }
12
13    public static int ctz(int n)
14    {
15        int c=0;
16        for(int i=5; n/i >=1; i*=5)
17        {
18            c+=n/i;
19        }
20        return c;
21    }
22
23 }
```

	Input	Expected	Got	
✓	3	0	0	✓
✓	60	14	14	✓
✓	100	24	24	✓
✓	1024	253	253	✓

Passed all tests! ✓

◀ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-03-Arrays](#) / [Lab-03-Logic Building](#)

Status Finished

Started Sunday, 6 October 2024, 5:26 PM

Completed Sunday, 6 October 2024, 6:12 PM

Duration 45 mins 17 secs

Question 1

Correct

Marked out of 5.00

Given an integer array as input, perform the following operations on the array, in the below specified sequence.

1. Find the maximum number in the array.
2. Subtract the maximum number from each element of the array.
3. Multiply the maximum number (found in step 1) to each element of the resultant array.

After the operations are done, return the resultant array.

Example 1:

`input1 = 4` (represents the number of elements in the `input1` array)

`input2 = {1, 5, 6, 9}`

Expected Output = `{-72, -36, 27, 0}`

Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

$$\{(1 - 9), (5 - 9), (6 - 9), (9 - 9)\} = \{-8, -4, -3, 0\}$$

Step 3: Multiplying the maximum number 9 to each of the resultant array:

$$\{(-8 \times 9), (-4 \times 9), (3 \times 9), (0 \times 9)\} = \{-72, -36, -27, 0\}$$

So, the expected output is the resultant array `{-72, -36, -27, 0}`.

Example 2:

`input1 = 5` (represents the number of elements in the `input1` array)

`input2 = {10, 87, 63, 42, 2}`

Expected Output = `{-6699, 0, -2088, -3915, -7395}`

Explanation:

Step 1: The maximum number in the given array is 87.

Step 2: Subtracting the maximum number 87 from each element of the array:

$$\{(10 - 87), (87 - 87), (63 - 87), (42 - 87), (2 - 87)\} = \{-77, 0, -24, -45, -85\}$$

Step 3: Multiplying the maximum number 87 to each of the resultant array:

$$\{(-77 \times 87), (0 \times 87), (-24 \times 87), (-45 \times 87), (-85 \times 87)\} = \{-6699, 0, -2088, -3915, -7395\}$$

So, the expected output is the resultant array `{-6699, 0, -2088, -3915, -7395}`.

Example 3:

`input1 = 2` (represents the number of elements in the `input1` array)

`input2 = {-9, 9}`

Expected Output = `{-162, 0}`

Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

$$\{(-9 - 9), (9 - 9)\} = \{-18, 0\}$$

Step 3: Multiplying the maximum number 9 to each of the resultant array:

$$\{(-18 \times 9), (0 \times 9)\} = \{-162, 0\}$$

So, the expected output is the resultant array `{-162, 0}`.

Note: The input array will contain not more than 100 elements

For example:

Input	Result
4 1 5 6 9	-72 -36 -27 0

Input	Result
5 10 87 63 42 2	-6699 0 -2088 -3915 -7395
2 -9 9	-162 0

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class A
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc = new Scanner(System.in);
7         int n=sc.nextInt();
8         int[] a=new int[n];
9         for (int i=0;i<n;i++)
10        {
11            a[i]=sc.nextInt();
12        }
13        int m=a[0];
14        for (int i=1;i<n;i++)
15        {
16            if(a[i]>m)
17            {
18                m=a[i];
19            }
20        }
21        for (int i=0;i<n;i++)
22        {
23            a[i]=(a[i]-m)*m;
24        }
25        for(int num : a)
26        {
27            System.out.print(num + " ");
28        }
29    }
30 }
31 }
```

	Input	Expected	Got	
✓	4 1 5 6 9	-72 -36 -27 0	-72 -36 -27 0	✓
✓	5 10 87 63 42 2	-6699 0 -2088 -3915 -7395	-6699 0 -2088 -3915 -7395	✓
✓	2 -9 9	-162 0	-162 0	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

Given an array of numbers, you are expected to return the sum of the longest sequence of POSITIVE numbers in the array.

If there are NO positive numbers in the array, you are expected to return -1.

In this question's scope, the number 0 should be considered as positive.

Note: If there are more than one group of elements in the array having the longest sequence of POSITIVE numbers, you are expected to return the total sum of all those POSITIVE numbers (see example 3 below).

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

input2 = {-12, -16, 12, 18, 18, 14, -4, -12, -13, 32, 34, -5, 66, 78, 78, -79}

Expected output = 62

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "12, 18, 18, 14", "12", "32, 34", and "66, 78, 78". The first sequence "12, 18, 18, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = $12 + 18 + 18 + 14 = 63$.

Example 2:

input1 = 11

input2 = {-22, -24, 16, -1, -17, -19, -37, -25, -19, -93, -61}

Expected output = -1

Explanation:

There are NO positive numbers in the input array. Therefore, the expected output for such cases = -1.

Example 3:

input1 = 16

input2 = {-58, 32, 26, 92, -10, -4, 12, 0, 12, -2, 4, 32, -9, -7, 78, -79}

Expected output = 174

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "32, 26, 92", "12, 0, 12", "4, 32", and "78". The first and second sequences "32, 26, 92" and "12, 0, 12" are the longest of the four as they contain 4 elements each. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = $(32 + 26 + 92) + (12 + 0 + 12) = 174$.

For example:

Input	Result
16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62
11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1
16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class lps
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8         int[] a = new int[n];
9         for (int i=0;i<n;i++)

```

```

10   {
11     a[i]=sc.nextInt();
12   }
13   int ml=0,cl=0,ms=0,cs=0;
14   boolean has = false;
15   for(int i=0;i<n;i++)
16   {
17     if(a[i]>=0)
18     {
19       has=true;
20       cl++;
21       cs+=a[i];
22     }
23     else
24     {
25       if(cl>ml)
26       {
27         ml=cl;
28         ms=cs;
29       }
30       else if(cl==ml)
31       {
32         ms+=cs;
33       }
34       cl=0;
35       cs=0;
36     }
37   }
38   if(cl>ml)
39   {
40     ms=cs;
41   }
42   else if(cl==ml)
43   {
44     ms+=cs;
45   }
46   System.out.println(has ? ms : -1);
47 }
48 }
```

	Input	Expected	Got	
✓	16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62	62	✓
✓	11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1	-1	✓
✓	16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174	174	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

You are provided with a set of numbers (array of numbers).

You have to generate the sum of specific numbers based on its position in the array set provided to you.

This is explained below:

Example 1:

Let us assume the encoded set of numbers given to you is:

input1:5 and input2: {1, 51, 436, 7860, 41236}

Step 1:

Starting from the 0th index of the array pick up digits as per below:

0th index – pick up the units value of the number (in this case is 1).

1st index - pick up the tens value of the number (in this case it is 5).

2nd index - pick up the hundreds value of the number (in this case it is 4).

3rd index - pick up the thousands value of the number (in this case it is 7).

4th index - pick up the ten thousands value of the number (in this case it is 4).

(Continue this for all the elements of the input array).

The array generated from Step 1 will then be – {1, 5, 4, 7, 4}.

Step 2:

Square each number present in the array generated in Step 1.

{1, 25, 16, 49, 16}

Step 3:

Calculate the sum of all elements of the array generated in Step 2 to get the final result. The result will be = 107.

Note:

- 1) While picking up a number in Step1, if you observe that the number is smaller than the required position then use 0.
- 2) In the given function, input1[] is the array of numbers and input2 represents the number of elements in input1.

Example 2:

input1: 5 and input1: {1, 5, 423, 310, 61540}

Step 1:

Generating the new array based on position, we get the below array:

{1, 0, 4, 0, 6}

In this case, the value in input1 at index 1 and 3 is less than the value required to be picked up based on position, so we use a 0.

Step 2:

{1, 0, 16, 0, 36}

Step 3:

The final result = 53.

For example:

Input	Result
5 1 51 436 7860 41236	107
5 1 5 423 310 61540	53

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class en
3 {
4     public static void main(String[] args)
5     {

```

```

6   Scanner sc = new Scanner(System.in);
7   int n = sc.nextInt();
8   int[] a = new int[n];
9   for(int i=0; i<n;i++)
10  {
11      a[i]=sc.nextInt();
12  }
13  int s=0;
14  for(int i =0; i<n;i++)
15  {
16      int num=a[i];
17      int d=0;
18      for(int j=0;j<=i;j++)
19      {
20          d=num%10;
21          num/=10;
22      }
23      s+=d*d;
24  }
25  System.out.println(s);
26
27
28 }
```

	Input	Expected	Got	
✓	5 1 51 436 7860 41236	107	107	✓
✓	5 1 5 423 310 61540	53	53	✓

Passed all tests! ✓

◀ Lab-03-MCQ

Jump to...

Simple Encoded Array ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-04-Classes and Objects](#) / [Lab-04-Logic Building](#)

Status Finished

Started Monday, 7 October 2024, 9:19 PM

Completed Monday, 7 October 2024, 10:34 PM

Duration 1 hour 14 mins

Question 1

Correct

Marked out of 5.00

Create a class called "Circle" with a radius attribute. You can access and modify this attribute using getter and setter methods. Calculate the area and circumference of the circle.

Area of Circle = πr^2

Circumference = $2\pi r$

Input:

2

Output:

Area = 12.57

Circumference = 12.57

For example:

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 import java.io.*;
2 import java.util.Scanner;
3 class Circle
4 {
5     private double radius;
6     public Circle(double radius){
7         // set the instance variable radius
8         this.radius =radius;
9     }
10    public void setRadius(double radius){
11        // set the radius
12        this.radius=radius;
13    }
14    public double getRadius()    {
15        // return the radius
16        return radius;
17    }
18    }
19    public double calculateArea() { // complete the below statement
20        return Math.PI*radius*radius;
21    }
22    }
23    public double calculateCircumference()    {
24        // complete the statement
25        return 2*Math.PI*radius;
26    }
27 }
28 class prog{
29     public static void main(String[] args) {
30         int r;
31         Scanner sc= new Scanner(System.in);
32         r=sc.nextInt();
33         Circle c= new Circle(r);
34         System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
35         // invoke the calculatecircumference method
36         System.out.println("Circumference = "+String.format("%.2f" , c.calculateCircumference()));
37         sc.close();
38     }
39 }
40 }
41 }
42 }
43 }
44 }
```

	Test	Input	Expected	Got	
✓	1	4	Area = 50.27 Circumference = 25.13	Area = 50.27 Circumference = 25.13	✓
✓	2	6	Area = 113.10 Circumference = 37.70	Area = 113.10 Circumference = 37.70	✓
✓	3	2	Area = 12.57 Circumference = 12.57	Area = 12.57 Circumference = 12.57	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

Create a class Student with two private attributes, name and roll number. Create three objects by invoking different constructors available in the class Student.

Student()

Student(String name)

Student(String name, int rollno)

Input:

No input

Output:**No-arg constructor is invoked****1 arg constructor is invoked****2 arg constructor is invoked****Name =null , Roll no = 0****Name =Rajalakshmi , Roll no = 0****Name =Lakshmi , Roll no = 101****For example:**

Test	Result
1	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101

Answer: (penalty regime: 0 %)

```

1 public class stud{
2     private String name;
3     private int roll;
4     public stud(){
5         System.out.println("No-arg constructor is invoked");
6         name=null;
7         roll=0;
8     }
9     public stud(String name){
10        System.out.println("1 arg constructor is invoked");
11        this.name=name;
12        roll=0;
13    }
14    public stud(String name,int roll){
15        System.out.println("2 arg constructor is invoked");
16        this.name=name;
17        this.roll=roll;
18    }
19    public static void main(String[]args){
20        stud s1=new stud();
21        stud s2=new stud("Rajalakshmi");
22        stud s3=new stud("Lakshmi",101);
23        System.out.println("Name =" +s1.name+ " , Roll no = " +s2.roll);
24        System.out.println("Name =" +s2.name+ " , Roll no = " +s2.roll);
25        System.out.println("Name =" +s3.name+ " , Roll no = " +s3.roll);
26    }
27 }
```

	Test	Expected	Got	
✓	1	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	✓

Passed all tests! ✓

Question 3

Incorrect

Marked out of 5.00

Create a Class Mobile with the attributes listed below,

```
private String manufacturer;
private String operating_system;
public String color;
private int cost;
```

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

```
void setManufacturer(String manufacturer){
    this.manufacturer= manufacturer;
}
```

```
String getManufacturer(){
    return manufacturer;}
```

Display the object details by overriding the `toString()` method.

For example:

Test	Result
1	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000

Answer: (penalty regime: 0 %)

```
1 public class mobile{
2     private String man;
3     private String os;
4     public String clr;
5     private int cost;
6     public mobile(String man,String os,String clr,int cost){
7         this.man=man;
8         this.os=os;
9         this.clr=clr;
10        this.cost=cost;
11    }
12    public String toString(){
13        return"manufacturer="+man+"\n"+ "operating_system =" +os+ "\n" + "color=" +clr+ "\n" + "cost=" +cost;
14    }
15    public static void main(String[]args){
16        mobile mobile=new
17        mobile("Redmi","Andriod","Blue",34000);
18        System.out.println(mobile);
19    }
20}
21}
```

Syntax Error(s)

```
mobile.java:13: error: illegal character: '\'
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: not a statement
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: ';' expected
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: illegal character: '\'
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: not a statement
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: ';' expected
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: illegal character: '\'
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: not a statement
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: ';' expected
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:13: error: unclosed string literal
    return"manufacturer="+man+"\n"+"operating_system ="+os+"\n"+"color="+clr+"\n"+cost="+cost;
               ^
mobile.java:21: error: class, interface, enum, or record expected
}
^
11 errors
```

[◀ Lab-04-MCQ](#)

Jump to...

[Number of Primes in a specified range ►](#)

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-05-Inheritance](#) / [Lab-05-Logic Building](#)

Status Finished

Started Tuesday, 8 October 2024, 9:08 AM

Completed Tuesday, 8 October 2024, 10:17 AM

Duration 1 hour 9 mins

Question 1

Correct

Marked out of 5.00

Create a class Mobile with constructor and a method basicMobile().

Create a subclass CameraMobile which extends Mobile class , with constructor and a method newFeature().

Create a subclass AndroidMobile which extends CameraMobile, with constructor and a method androidMobile().

display the details of the Android Mobile class by creating the instance. .

```
class Mobile{
```

```
}
```

```
class CameraMobile extends Mobile {
```

```
}
```

```
class AndroidMobile extends CameraMobile {
```

```
}
```

expected output:

Basic Mobile is Manufactured

Camera Mobile is Manufactured

Android Mobile is Manufactured

Camera Mobile with 5MG px

Touch Screen Mobile is Manufactured

For example:

Result
Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured

Answer: (penalty regime: 0 %)

```
1 v class mob{
2 v     mob(){
3 v         System.out.println("Basic Mobile is Manufactured");
4 v     }
5 v     Void basmob(){
6 v         System.out.println("Basic Mobile is Manufactured");
7 v         return null;
8 v     }
9 }
10 v class cam extends mob{
11 v     cam(){
12 v         super();
13 v         System.out.println("Camera Mobile is Manufactured");
14 v     }
15 v     void newm(){
16 v         System.out.println("Camera Mobile with 5MG px");
17 v     }
18 }
19 v class and extends cam{
20 v     and(){
21 v         super();
22 v         System.out.println("Android Mobile is Manufactured");
23 v     }
24 v     Void andmob(){
25 v         System.out.println("Touch Screen Mobile is Manufactured");
26 v         return null;
27 v     }
28 }
29 v public class Main{
30 v     public static void main(String[]args){
31 v         and andmob=new and();
32 v         andmob.newm();
33 v         andmob.andmob();
34 v     }
35 }
```

	Expected	Got	
✓	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

create a class called College with attribute String name, constructor to initialize the name attribute , a method called Admitted(). Create a subclass called CSE that extends Student class, with department attribute , Course() method to sub class. Print the details of the Student.

College:

```
String collegeName;
public College() {}
public admitted() {}
```

Student:

```
String studentName;
String department;
public Student(String collegeName, String studentName, String depart) {}
public toString()
```

Expected Output:

A student admitted in REC

CollegeName : REC

StudentName : Venkatesh

Department : CSE

For example:

Result
A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 class College
2 {
3     public String collegeName;
4
5     public College(String collegeName) {
6         // initialize the instance variables
7         this.collegeName=collegeName;
8     }
9
10    public void admitted() {
11        System.out.println("A student admitted in "+collegeName);
12    }
13 }
14 class Student extends College{
15
16     String studentName;
17     String department;
18
19     public Student(String collegeName, String studentName, String department) {
20         // initialize the instance variables
21         super(collegeName);
22         this.studentName=studentName;
23         this.department=department;
24     }
25 }
26
27    public String toString(){
28        // return the details of the student
29        return "CollegeName : "+collegeName+"\n"+ "StudentName : "+studentName+"\n"+ "Department : "+department
30    }
31 }
32 class Main {
33     public static void main (String[] args) {
34         Student s1 = new Student("REC","Venkatesh","CSE");
```

```
35     s1.admitted();  
36         // invoke the admitted() method  
37     System.out.println(s1.toString());  
38 }  
39 }
```

	Expected	Got	
✓	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

Create a class known as "BankAccount" with methods called deposit() and withdraw().

Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

For example:

Result

```
Create a Bank Account object (A/c No. BA1234) with initial balance of $500:  
Deposit $1000 into account BA1234:  
New balance after depositing $1000: $1500.0  
Withdraw $600 from account BA1234:  
New balance after withdrawing $600: $900.0  
Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:  
Try to withdraw $250 from SA1000!  
Minimum balance of $100 required!  
Balance after trying to withdraw $250: $300.0
```

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 class BankAccount {  
2     // Private field to store the account number  
3     private String accountNumber;  
4  
5     // Private field to store the balance  
6     private double balance;  
7  
8     // Constructor to initialize account number and balance  
9     public BankAccount(String accountNumber, double balance)  
10    {  
11        this.accountNumber=accountNumber;  
12        this.balance=balance;  
13    }  
14  
15  
16  
17  
18     // Method to deposit an amount into the account  
19     public void deposit(double amount) {  
20         // Increase the balance by the deposit amount  
21         balance+=amount;  
22     }  
23  
24     // Method to withdraw an amount from the account  
25     public void withdraw(double amount) {  
26         // Check if the balance is sufficient for the withdrawal  
27         if (balance >= amount) {  
28             // Decrease the balance by the withdrawal amount  
29             balance -= amount;  
30         } else {  
31             // Print a message if the balance is insufficient  
32             System.out.println("Insufficient balance");  
33         }  
34     }  
35  
36     // Method to get the current balance  
37     public double getBalance() {  
38         // Return the current balance  
39         return balance;  
40     }  
41     public String getAccountNumber(){  
42         return accountNumber;  
43     }  
44 }  
45  
46 class SavingsAccount extends BankAccount {  
47     // Constructor to initialize account number and balance  
48     public SavingsAccount(String accountNumber, double balance) {  
49         // Call the parent class constructor
```

```

50     // call the parent class constructor
51     super(accountNumber,balance);
52 }
```

	Expected	Got	
✓	Create a Bank Account object (A/c No. BA1234) with initial balance of \$500: Deposit \$1000 into account BA1234: New balance after depositing \$1000: \$1500.0 Withdraw \$600 from account BA1234: New balance after withdrawing \$600: \$900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of \$300: Try to withdraw \$250 from SA1000! Minimum balance of \$100 required! Balance after trying to withdraw \$250: \$300.0	Create a Bank Account object (A/c No. BA1234) with initial balance of \$500: Deposit \$1000 into account BA1234: New balance after depositing \$1000: \$1500.0 Withdraw \$600 from account BA1234: New balance after withdrawing \$600: \$900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of \$300: Try to withdraw \$250 from SA1000! Minimum balance of \$100 required! Balance after trying to withdraw \$250: \$300.0	✓

Passed all tests! ✓

◀ Lab-05-MCQ

Jump to...

Is Palindrome Number? ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-06-String, StringBuffer](#) / [Lab-06-Logic Building](#)

Status Finished

Started Tuesday, 8 October 2024, 5:02 PM

Completed Tuesday, 8 October 2024, 6:17 PM

Duration 1 hour 15 mins

Question 1

Correct

Marked out of 5.00

Given a String input1, which contains many number of words separated by : and each word contains exactly two lower case alphabets, generate an output based upon the below 2 cases.

Note:

1. All the characters in input 1 are lowercase alphabets.
2. input 1 will always contain more than one word separated by :
3. Output should be returned in uppercase.

Case 1:

Check whether the two alphabets are same.

If yes, then take one alphabet from it and add it to the output.

Example 1:

input1 = ww:ii:pp:rr:oo

output = WIPRO

Explanation:

word1 is ww, both are same hence take w

word2 is ii, both are same hence take i

word3 is pp, both are same hence take p

word4 is rr, both are same hence take r

word5 is oo, both are same hence take o

Hence the output is WIPRO

Case 2:

If the two alphabets are not same, then find the position value of them and find maximum value – minimum value.

Take the alphabet which comes at this (maximum value - minimum value) position in the alphabet series.

Example 2"

input1 = zx:za:ee

output = BYE

Explanation

word1 is zx, both are not same alphabets

position value of z is 26

position value of x is 24

max – min will be $26 - 24 = 2$

Alphabet which comes in 2nd position is b

Word2 is za, both are not same alphabets

position value of z is 26

position value of a is 1

max – min will be $26 - 1 = 25$

Alphabet which comes in 25th position is y

word3 is ee, both are same hence take e

Hence the output is BYE

For example:

Input	Result
ww:ii:pp:rr:oo	WIPRO
zx:za:ee	BYE

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 class diff
3 {
4     char different(char a, char b)
5     {
6         if((int)a!=(int)b)
7             return (char)((int)'a'+((int)a-(int)b)-1);
8         return a;
9     }
10 }
11
12 public class Main
13 {
14     public static void main(String[] args)
15     {
16         Scanner sc = new Scanner(System.in);
17         diff z = new diff();
18         String q = sc.nextLine();
19         StringBuffer ans = new StringBuffer();
20         StringBuffer temp = new StringBuffer();
21         for(int i=0;i<q.length();i++)
22         {
23             if(q.charAt(i)==':')
24             {
25                 temp.append(" ");
26             }
27             else
28             {
29                 temp.append(Character.toString(q.charAt(i)));
30             }
31         }
32         String h = temp.toString();
33         for (int i=0;i<temp.length();i++)
34         {
35             if(i%3==0)
36             {
37                 ans.append(Character.toString(z.different(h.charAt(i),h.charAt(i+1))));
38             }
39         }
40         System.out.println(ans.toString().toUpperCase());
41     }
42 }
```

	Input	Expected	Got	
✓	ww:ii:pp:rr:oo	WIPRO	WIPRO	✓
✓	zx:za:ee	BYE	BYE	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

You are provided a string of words and a 2-digit number. The two digits of the number represent the two words that are to be processed.

For example:

If the string is "Today is a Nice Day" and the 2-digit number is 41, then you are expected to process the 4th word ("Nice") and the 1st word ("Today").

The processing of each word is to be done as follows:

Extract the Middle-to-Begin part: Starting from the middle of the word, extract the characters till the beginning of the word.

Extract the Middle-to-End part: Starting from the middle of the word, extract the characters till the end of the word.

If the word to be processed is "Nice":

Its Middle-to-Begin part will be "iN".

Its Middle-to-End part will be "ce".

So, merged together these two parts would form "iNce".

Similarly, if the word to be processed is "Today":

Its Middle-to-Begin part will be "doT".

Its Middle-to-End part will be "day".

So, merged together these two parts would form "doTday".

Note: Note that the middle letter 'd' is part of both the extracted parts. So, for words whose length is odd, the middle letter should be included in both the extracted parts.

Expected output:

The expected output is a string containing both the processed words separated by a space "iNce doTday"

Example 1:

input1 = "Today is a Nice Day"

input2 = 41

output = "iNce doTday"

Example 2:

input1 = "Fruits like Mango and Apple are common but Grapes are rare"

input2 = 39

output = "naMngo arGpes"

Note: The input string input1 will contain only alphabets and a single space character separating each word in the string.

Note: The input string input1 will NOT contain any other special characters.

Note: The input number input2 will always be a 2-digit number ($>=11$ and $<=99$). One of its digits will never be 0. Both the digits of the number will always point to a valid word in the input1 string.

For example:

Input	Result
Today is a Nice Day 41	iNce doTday
Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 public class mix
3 {
4     public static void main(String[] args)
5     {
6         Scanner scan = new Scanner(System.in);
7         String g = scan.nextLine();
8         int n = scan.nextInt(), ones, flag=0;
9         StringBuffer temp = new StringBuffer();
10        StringBuffer temp1 = new StringBuffer();

```

```

11 int space=0;
12 while(n>0)
13 {
14     ones=(n%10)-1;
15     for(int i=0;i<g.length();i++)
16     {
17         if(g.charAt(i)==' ')
18         {
19             space=space+1;
20         }
21         else if(space==ones && flag==0)
22         {
23             temp.append(Character.toString(g.charAt(i)));
24         }
25         else if(space==ones && flag==1)
26         {
27             temp1.append(Character.toString(g.charAt(i)));
28         }
29     }
30     space=0;
31     flag=1;
32     n=n/10;
33 }
34 rew m=new rew();
35 System.out.println(m.r(temp1.toString()) + " " + m.r(temp.toString()));
36 }
37 }
38
39 class rew
40 {
41     String r(String a)
42     {
43         int le = a.length(),n,q;
44         StringBuffer temp3 = new StringBuffer();
45         if(le%2==1)
46         {
47             n=((int)(le/2));
48             q=((int)(le/2));
49         }
50         else
51         {
52             n=((int)(le/2))-1;

```

	Input	Expected	Got	
✓	Today is a Nice Day 41	iNce doTday	iNce doTday	✓
✓	Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes	naMngo arGpes	✓

Passed all tests! ✓

Question 3

Incorrect

Marked out of 5.00

Given 2 strings input1 & input2.

- Concatenate both the strings.
- Remove duplicate alphabets & white spaces.
- Arrange the alphabets in descending order.

Assumption 1:

There will either be alphabets, white spaces or null in both the inputs.

Assumption 2:

Both inputs will be in lower case.

Example 1:

Input 1: apple

Input 2: orange

Output: rponlgea

Example 2:

Input 1: fruits

Input 2: are good

Output: utsroigfeda

Example 3:

Input 1: ""

Input 2: ""

Output: null

For example:

Test	Input	Result
1	apple orange	rponlgea
2	fruits are good	utsroigfeda

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 public class HelloWorld
3 {
4     Scanner scan=new Scanner(System.in);
5     String a = scan.nextLine();
6     String b = scan.nextLine();
7     StringBuffer ab = new StringBuffer();
8     for(int i=0;i<a.length();i++)
9     {
10         if(a.isEmpty())
11         {
12             System.out.print("null");
13         }
14     }
15     for(int i=0;i<b.length();i++)
16     {
17         if(b.isEmpty())
18         {
19             System.out.println("null")
20         }
21     }
22     for(int i=0;i<a.length();i++)
23     {
24         if(a.charAt(i)!=" ")
25         {
26             ab.append(Character.toString(a.charAt(i)));
27         }
28     }
29     System.out.println(ab);
30 }
```

```
28     }
29     for(int i=0;i<b.length();i++)
30     {
31         if(b.charAt(i)!=' ')
32         {
33             ab.append(Character.toString(b.charAt(i)));
34         }
35     }
36     char[] d = ab.toString().toCharArray();
37     Arrays.sort(d);
38     for(int i=d.length-1;i>=1;i--)
39     {
40         if(d[i]!=d[i-1])
41             System.out.print(d[i]);
42     }
43     System.out.print(d[0]);
44 }
45 }
46 }
47 }
```

Syntax Error(s)

```
prog.java:8: error: illegal start of type
    for(int i=0;i<a.length();i++)
        ^
prog.java:8: error: > expected
    for(int i=0;i<a.length();i++)
        ^
prog.java:8: error: <identifier> expected
    for(int i=0;i<a.length();i++)
        ^
prog.java:15: error: illegal start of type
    for(int i=0;i<b.length();i++)
        ^
prog.java:15: error: > expected
    for(int i=0;i<b.length();i++)
        ^
prog.java:15: error: <identifier> expected
    for(int i=0;i<b.length();i++)
        ^
prog.java:19: error: ';' expected
    System.out.println("null")
        ^
prog.java:22: error: illegal start of type
    for(int i=0;i<a.length();i++)
        ^
prog.java:22: error: > expected
    for(int i=0;i<a.length();i++)
        ^
prog.java:22: error: <identifier> expected
    for(int i=0;i<a.length();i++)
        ^
prog.java:24: error: empty character literal
    if(a.charAt(i)!='')
        ^
prog.java:29: error: illegal start of type
    for(int i=0;i<b.length();i++)
        ^
prog.java:29: error: > expected
    for(int i=0;i<b.length();i++)
        ^
prog.java:29: error: <identifier> expected
    for(int i=0;i<b.length();i++)
        ^
prog.java:31: error: empty character literal
    if(b.charAt(i)!='')
        ^
prog.java:37: error: <identifier> expected
    Arrays.sort(d);
        ^
prog.java:37: error: <identifier> expected
    Arrays.sort(d);
        ^
prog.java:38: error: illegal start of type
    for(int i=d.length-1;i>=1;i--)
        ^
prog.java:38: error: <identifier> expected
    for(int i=d.length-1;i>=1;i--)
        ^
prog.java:38: error: <identifier> expected
    for(int i=d.length-1;i>=1;i--)
        ^
prog.java:43: error: <identifier> expected
    System.out.print(d[0]);
        ^
prog.java:43: error: ']' expected
    System.out.print(d[0]);
        ^
prog.java:45: error: class, interface, enum, or record expected
    }
    ^
23 errors
```

[◀ Lab-06-MCQ](#)

Jump to...

[Return second word in Uppercase ►](#)

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-07-Interfaces](#) / [Lab-07-Logic Building](#)

Status Finished

Started Tuesday, 8 October 2024, 6:18 PM

Completed Tuesday, 8 October 2024, 7:00 PM

Duration 41 mins 56 secs

Question 1

Correct

Marked out of 5.00

RBI issues all national banks to collect interest on all customer loans.

Create an RBI interface with a variable String parentBank="RBI" and abstract method rateOfInterest().

RBI interface has two more methods default and static method.

```
default void policyNote() {
    System.out.println("RBI has a new Policy issued in 2023.");
}

static void regulations() {
    System.out.println("RBI has updated new regulations on 2024.");
}
```

Create two subclasses SBI and Karur which implements the RBI interface.

Provide the necessary code for the abstract method in two sub-classes.

Sample Input/Output:

RBI has a new Policy issued in 2023
RBI has updated new regulations in 2024.
SBI rate of interest: 7.6 per annum.
Karur rate of interest: 7.4 per annum.

For example:

Test	Result
1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.

Answer: (penalty regime: 0 %)

```
1 interface RBI
2 {
3     String parentBank = "RBI";
4     double rateOfInterest();
5
6     default void policyNote()
7     {
8         System.out.println("RBI has a new Policy issued in 2023");
9     }
10    static void regulations()
11    {
12        System.out.println("RBI has updated new regulations in 2024.");
13    }
14}
15
16 class SBI implements RBI
17 {
18     public double rateOfInterest()
19     {
20         return 7.6;
21     }
22 }
23 class Karur implements RBI
24 {
25     public double rateOfInterest()
26     {
27         return 7.4;
28     }
29 }
30
31 public class Main
32 {
33     public static void main(String[] args)
34     {
35         RBI rbi = new SBI();
36         rbi.policyNote();
```

```
37     . . . . .  
38     RBI.regulations();  
39  
39     SBI sbi = new SBI();  
40     System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");  
41  
42     Karur karur = new Karur();  
43     System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");  
44  
45 }  
46 }
```

	Test	Expected	Got	
✓	1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	✓

Passed all tests! ✓

//

Question 2

Correct

Marked out of 5.00

Create interfaces shown below.

```
interface Sports {
    public void setHomeTeam(String name);
    public void setVisitingTeam(String name);
}
interface Football extends Sports {
    public void homeTeamScored(int points);
    public void visitingTeamScored(int points);}
```

create a class College that implements the Football interface and provides the necessary functionality to the abstract methods.

sample Input:

Rajalakshmi
Saveetha
22
21

Output:

Rajalakshmi 22 scored
Saveetha 21 scored
Rajalakshmi is the Winner!

For example:

Test	Input	Result
1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!

Answer: (penalty regime: 0 %)

Reset answer

```
1 import java.util.Scanner;
2 interface Sports
3 {
4     void setHomeTeam(String name);
5     void setVisitingTeam(String name);
6 }
7 interface Football extends Sports
8 {
9     void homeTeamScored(int points);
10    void visitingTeamScored(int points);
11 }
12
13 class College implements Football
14 {
15     private String homeTeam;
16     private String visitingTeam;
17     private int homeTeamPoints=0;
18     private int visitingTeamPoints=0;
19
20     public void setHomeTeam(String name)
21     {
22         this.homeTeam=name;
23     }
24
25     public void setVisitingTeam(String name)
26     {
27         this.visitingTeam=name;
28     }
29
30     public void homeTeamScored(int points)
31     {
32         homeTeamPoints+=points;
33         System.out.println(homeTeam+" "+points+" scored");
34     }
```

```

35 public void visitingTeamScored(int points)
36 {
37     visitingTeamPoints+=points;
38     System.out.println(visitingTeam+ " "+points+" scored");
39 }
40
41 public void winningTeam()
42 {
43     if(homeTeamPoints>visitingTeamPoints)
44     {
45         System.out.println(homeTeam+ " is the winner!");
46     }
47     else if(homeTeamPoints<visitingTeamPoints)
48     {
49         System.out.println(visitingTeam+ " is the winner!");
50     }
51     else
52     {

```

	Test	Input	Expected	Got	
✓	1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	✓
✓	2	Anna Balaji 21 21	Anna 21 scored Balaji 21 scored It's a tie match.	Anna 21 scored Balaji 21 scored It's a tie match.	✓
✓	3	SRM VIT 20 21	SRM 20 scored VIT 21 scored VIT is the winner!	SRM 20 scored VIT 21 scored VIT is the winner!	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.

```
interface Playable {
    void play();
}

class Football implements Playable {
    String name;
    public Football(String name){
        this.name=name;
    }
    public void play() {
        System.out.println(name+" is Playing football");
    }
}
```

Similarly, create Volleyball and Basketball classes.

Sample output:

```
Sadvin is Playing football
Sanjay is Playing volleyball
Sruthi is Playing basketball
```

For example:

Test	Input	Result
1	Sadvin Sanjay Sruthi	Sadvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball
2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2 interface Playable
3 {
4     void play();
5 }
6
7 class Football implements Playable
8 {
9     String name;
10    public Football(String name)
11    {
12        this.name=name;
13    }
14
15    public void play()
16    {
17        System.out.println(name+ " is Playing football");
18    }
19 }
20
21 class Volleyball implements Playable
22 {
23     String name;
24
25    public Volleyball(String name)
26    {
27        this.name=name;
28    }
29
30    public void play()
31    {
32        System.out.println(name+ " is Playing volleyball");
33    }
34 }
```

```

33     }
34 }
35
36 class Basketball implements Playable
37 {
38     String name;
39
40     public Basketball(String name)
41     {
42         this.name=name;
43     }
44
45     public void play()
46     {
47         System.out.println(name+ " is Playing basketball");
48     }
49 }
50
51 public class Main
52 {

```

	Test	Input	Expected	Got	
✓	1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	✓
✓	2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	✓

Passed all tests! ✓

◀ Lab-07-MCQ

Jump to...

Generate series and find Nth element ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-08 - Polymorphism, Abstract Classes, final Keyword](#) / [Lab-08-Logic Building](#)

Status Finished

Started Tuesday, 8 October 2024, 7:01 PM

Completed Tuesday, 8 October 2024, 7:36 PM

Duration 34 mins 2 secs

Question 1

Correct

Marked out of 5.00

As a logic building learner you are given the task to extract the string which has vowel as the first and last characters from the given array of Strings.

Step1: Scan through the array of Strings, extract the Strings with first and last characters as vowels; these strings should be concatenated.

Step2: Convert the concatenated string to lowercase and return it.

If none of the strings in the array has first and last character as vowel, then return no matches found

input1: an integer representing the number of elements in the array.

input2: String array.

Example 1:

input1: 3

input2: {"oreo", "sirish", "apple"}

output: oreoapple

Example 2:

input1: 2

input2: {"Mango", "banana"}

output: no matches found

Explanation:

None of the strings has first and last character as vowel.

Hence the output is no matches found.

Example 3:

input1: 3

input2: {"Ate", "Ace", "Girl"}

output: ateace

For example:

Input	Result
3 oreo sirish apple	oreoapple
2 Mango banana	no matches found
3 Ate Ace Girl	ateace

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class cse
3 {
4     public static String evs(String[] stringArray)
5     {
6         StringBuilder result = new StringBuilder();
7         String vow = "aeiouAEIOU";
8         for (String s : stringArray)
9         {
10             if(s.length() > 0 && vow.indexOf(s.charAt(0)) != -1 && vow.indexOf(s.charAt(s.length() - 1)) != -1)
11             {
12                 result.append(s);
13             }
14         }
15         return result.length() > 0 ? result.toString().toLowerCase() : "no matches found";
16     }
17 }
18

```

```
19 public static void main(String[] args)
20 {
21     Scanner sc = new Scanner(System.in);
22     int n = sc.nextInt();
23     sc.nextLine();
24
25     String input = sc.nextLine();
26     String[] strings = input.split(" ");
27
28     String result = evs(strings);
29     System.out.println(result);
30     sc.close();
31 }
32 }
33 }
```

	Input	Expected	Got	
✓	3 oreo sirish apple	oreoapple	oreoapple	✓
✓	2 Mango banana	no matches found	no matches found	✓
✓	3 Ate Ace Girl	ateace	ateace	✓

Passed all tests! ✓

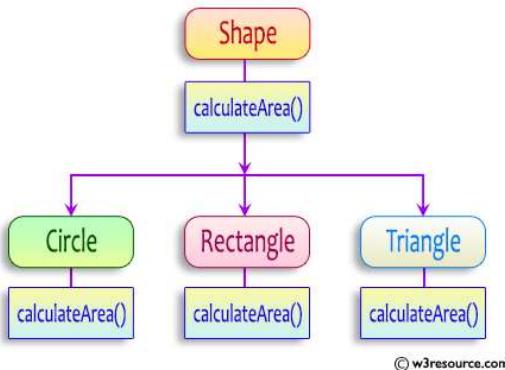
Question 2

Correct

Marked out of 5.00

Create a base class Shape with a method called calculateArea(). Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

In the given exercise, here is a simple diagram illustrating polymorphism implementation:



```

abstract class Shape {
    public abstract double calculateArea();
}
}

```

System.out.printf("Area of a Triangle :%.2f%n",((0.5)*base*height)); // use this statement

sample Input :

```

4 // radius of the circle to calculate area Pi*r*r
5 // length of the rectangle
6 // breadth of the rectangle to calculate the area of a rectangle
4 // base of the triangle
3 // height of the triangle

```

OUTPUT:

Area of a circle :50.27

Area of a Rectangle :30.00

Area of a Triangle :6.00

For example:

Test	Input	Result
1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00
2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 abstract class Shape
4 {
5     public abstract double calculateArea();
6
7 }
8
9 class Circle extends Shape
10 {
11     private double radius;
12 }

```

```

12
13     public Circle (double radius)
14     {
15         this.radius=radius;
16     }
17
18     @Override
19
20     public double calculateArea()
21     {
22         return Math.PI*radius*radius;
23     }
24 }
25
26 class Rectangle extends Shape
27 {
28     private double length;
29     private double breadth;
30
31     public Rectangle (double length, double breadth)
32     {
33         this.length = length;
34         this.breadth = breadth;
35     }
36
37     @Override
38
39     public double calculateArea()
40     {
41         return length*breadth;
42     }
43 }
44
45
46 class Triangle extends Shape
47 {
48     private double base;
49     private double height;
50
51     public Triangle (double base, double height)
52     {

```

	Test	Input	Expected	Got	
✓	1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	✓
✓	2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

1. Final Variable:

- Once a variable is declared `final`, its value cannot be changed after it is initialized.
- It must be initialized when it is declared or in the constructor if it's not initialized at declaration.
- It can be used to define constants

```
final int MAX_SPEED = 120; // Constant value, cannot be changed
```

2. Final Method:

- A method declared `final` cannot be overridden by subclasses.
- It is used to prevent modification of the method's behavior in derived classes.

```
public final void display() {
    System.out.println("This is a final method.");
}
```

3. Final Class:

- A class declared as `final` cannot be subclassed (i.e., no other class can inherit from it).
- It is used to prevent a class from being extended and modified.
- `public final class Vehicle {
 // class code
}`

Given a Java Program that contains the bug in it, your task is to clear the bug to the output.

you should delete any piece of code.

For example:

Test	Result
1	The maximum speed is: 120 km/h This is a subclass of FinalExample.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 final class FinalExample {
2
3     // Final variable
4     final int maxSpeed = 120;
5
6     // Final method
7     public final void display()
8     {
9         System.out.println("The maximum speed is: " + maxSpeed + " km/h");
10    }
11 }
12
13 public class Test
14 {
15     public static void main(String[] args) {
16         FinalExample obj = new FinalExample();
17         obj.display();
18
19         System.out.println("This is a subclass of FinalExample.");
20     }
21 }
```

	Test	Expected	Got	
✓	1	The maximum speed is: 120 km/h This is a subclass of FinalExample.	The maximum speed is: 120 km/h This is a subclass of FinalExample.	✓

Passed all tests! ✓

[◀ Lab-08-MCQ](#)

Jump to...

[FindStringCode ►](#)

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-09-Exception Handling](#) / [Lab-09-Logic Building](#)

Status Finished

Started Wednesday, 16 October 2024, 4:00 PM

Completed Wednesday, 16 October 2024, 4:32 PM

Duration 31 mins 43 secs

Question 1

Correct

Marked out of 5.00

Write a Java program to create a method that takes an integer as a parameter and throws an exception if the number is odd.

Sample input and Output:

```
82 is even.  
Error: 37 is odd.
```

Fill the preloaded answer to get the expected output.

For example:

Result
82 is even. Error: 37 is odd.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 class prog {
2     public static void main(String[] args) {
3         int n = 82;
4         trynumber(n);
5         n = 37;
6         trynumber(n);
7         // call the trynumber(n);
8     }
9
10    public static void trynumber(int n) {
11        try {
12            //call the checkEvenNumber()
13            checkEvenNumber(n);
14            System.out.println(n + " is even.");
15        } catch (Exception e) {
16            System.out.println("Error: " + e.getMessage());
17        }
18    }
19
20    public static void checkEvenNumber(int number) throws Exception {
21        if (number % 2 != 0) {
22            throw new Exception(number + " is odd.");
23        }
24    }
25 }
```

	Expected	Got	
✓	82 is even. Error: 37 is odd.	82 is even. Error: 37 is odd.	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

In the following program, an array of integer data is to be initialized.

During the initialization, if a user enters a value other than an integer, it will throw an InputMismatchException exception.

On the occurrence of such an exception, your program should print "You entered bad data."

If there is no such exception it will print the total sum of the array.

```
/* Define try-catch block to save user input in the array "name"
 If there is an exception then catch the exception otherwise print the total sum of the array. */
```

Sample Input:

```
3
5 2 1
```

Sample Output:

```
8
```

Sample Input:

```
2
1 g
```

Sample Output:

```
You entered bad data.
```

For example:

Input	Result
3	8
5 2 1	
2	You entered bad data.
1 g	

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2 import java.util.InputMismatchException;
3 class prog {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int length = sc.nextInt();
7         // create an array to save user input
8         int[] name = new int[length];
9         int sum=0;//save the total sum of the array.
10
11     /* Define try-catch block to save user input in the array "name"
12     If there is an exception then catch the exception otherwise print
13     the total sum of the array. */
14     try
15     {
16         for(int i=0;i<length;i++)
17         {
18             name[i]=sc.nextInt();
19             sum+=name[i];
20         }
21         System.out.println(sum);
22     }
23     catch (InputMismatchException e)
24     {
25         System.out.println("You entered bad data.");
26     }
27 }
28 }
```

	Input	Expected	Got	
✓	3 5 2 1	8	8	✓
✓	2 1 g	You entered bad data.	You entered bad data.	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

Write a Java program to handle `ArithmetException` and `ArrayIndexOutOfBoundsException`.

Create an array, read the input from the user, and store it in the array.

Divide the 0th index element by the 1st index element and store it.

if the 1st element is zero, it will throw an exception.

if you try to access an element beyond the array limit throws an exception.

Input:

```
5
10 0 20 30 40
```

Output:

java.lang.ArithmetException: / by zero

I am always executed

Input:

```
3
10 20 30
```

Output

`java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3`

I am always executed

For example:

Test	Input	Result
1	6 1 0 4 1 2 8	<code>java.lang.ArithmetException: / by zero</code> I am always executed

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 class prog {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int length = sc.nextInt();
6         int[] arr = new int[length];
7
8         for(int i=0;i<length;i++)
9         {
10             arr[i]=sc.nextInt();
11         }
12         try
13         {
14             int result = arr[0] / arr[1];
15         }
16         catch (ArithmetException e)
17         {
18             System.out.println("java.lang.ArithmetException: / by zero");
19         }
20         try
21         {
22             System.out.println(arr[length]);
23         }
24         catch(ArrayIndexOutOfBoundsException e)
25         {
26             if (length==6)
27             {
28             }
29             else
30             {
31                 System.out.println("java.lang.ArrayIndexOutOfBoundsException: " + e.getMessage());
32             }
33         }
34         finally
35         {
36             System.out.println("I am always executed");
37         }
}

```

```
38 }  
39 }  
40 }
```

	Test	Input	Expected	Got	
✓	1	6 1 0 4 1 2 8	java.lang.ArithmetricException: / by zero I am always executed	java.lang.ArithmetricException: / by zero I am always executed	✓
✓	2	3 10 20 30	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	✓

Passed all tests! ✓

◀ Lab-09-MCQ

Jump to...

The "Nambiar Number" Generator ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-10- Collection- List](#) / [Lab-10-Logic Building](#)

Status Finished

Started Sunday, 3 November 2024, 7:43 AM

Completed Sunday, 3 November 2024, 8:34 AM

Duration 50 mins 24 secs

Question 1

Correct

Marked out of 1.00

Given an ArrayList, the task is to get the first and last element of the ArrayList in Java.

Input: ArrayList = [1, 2, 3, 4]
Output: First = 1, Last = 4

Input: ArrayList = [12, 23, 34, 45, 57, 67, 89]
Output: First = 12, Last = 89

Approach:

1. Get the ArrayList with elements.
2. Get the first element of ArrayList using the get(index) method by passing index = 0.
3. Get the last element of ArrayList using the get(index) method by passing index = size – 1.

Answer: (penalty regime: 0 %)

```

1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 public class FirstAndLastElement
5 {
6     public static void main(String[] args)
7     {
8         Scanner scanner = new Scanner(System.in);
9
10        int n=scanner.nextInt();
11        ArrayList<Integer> list=new ArrayList<>();
12        for(int i=0;i<n;i++)
13        {
14            list.add(scanner.nextInt());
15        }
16
17        printFirstandLast(list);
18    }
19    public static void printFirstandLast(ArrayList<Integer> list)
20    {
21        if(list.isEmpty())
22        {
23            System.out.println("The list is empty.");
24            return;
25        }
26        int first=list.get(0);
27        int last=list.get(list.size()-1);
28        System.out.println("ArrayList: "+list);
29        System.out.println("First : " + first + ", Last : " +last);
30    }
31 }
32

```

	Test	Input	Expected	Got	
✓	1	6 30 20 40 50 10 80	ArrayList: [30, 20, 40, 50, 10, 80] First : 30, Last : 80	ArrayList: [30, 20, 40, 50, 10, 80] First : 30, Last : 80	✓
✓	2	4 5 15 25 35	ArrayList: [5, 15, 25, 35] First : 5, Last : 35	ArrayList: [5, 15, 25, 35] First : 5, Last : 35	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 1.00

The given Java program is based on the ArrayList methods and its usage. The Java program is partially filled. Your task is to fill in the incomplete statements to get the desired output.

```
list.set();
list.indexOf());
list.lastIndexOf())
list.contains()
list.size());
list.add();
list.remove();
```

The above methods are used for the below Java program.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 class prog {
5
6 public static void main(String[] args)
7 {
8 Scanner sc= new Scanner(System.in);
9 int n = sc.nextInt();
10
11 ArrayList<Integer> list = new ArrayList<Integer>();
12
13 for(int i = 0; i<n;i++)
14 {
15 list.add(sc.nextInt());
16 }
17 // printing initial value ArrayList
18 System.out.println("ArrayList: " +list);
19
20 list.set(1,100);
21 //Replacing the element at index 1 with 100
22
23
24 //Getting the index of first occurrence of 100
25 System.out.println("Index of 100 = "+list.indexOf(100));
26
27 //Getting the index of last occurrence of 100
28 System.out.println("LastIndex of 100 = "+list.lastIndexOf(100));
29 // Check whether 200 is in the list or not
30 System.out.println(list.contains(200)); //Output : false
31 // Print ArrayList size
32 System.out.println("Size Of ArrayList = "+list.size());
33 //Inserting 500 at index 1
34     list.add(1,500); // code here
35 //Removing an element from position 3
36     list.remove(3); // code here
37 System.out.print("ArrayList: " + list);
38 }
39 }
```

	Test	Input	Expected	Got	
✓	1	5 1 2 3 100 5	ArrayList: [1, 2, 3, 100, 5] Index of 100 = 1 LastIndex of 100 = 3 false Size Of ArrayList = 5 ArrayList: [1, 500, 100, 100, 5]	ArrayList: [1, 2, 3, 100, 5] Index of 100 = 1 LastIndex of 100 = 3 false Size Of ArrayList = 5 ArrayList: [1, 500, 100, 100, 5]	✓

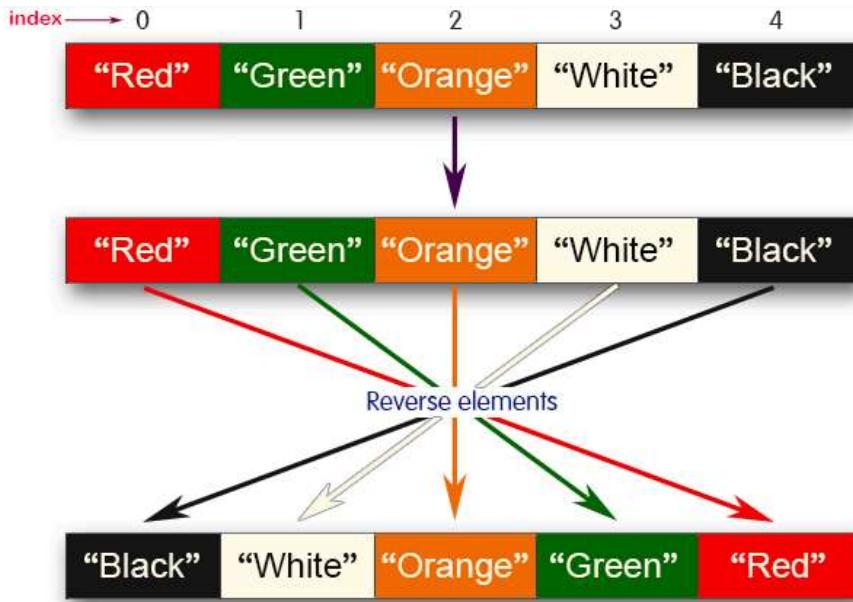
Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

Write a Java program to reverse elements in an array list.



Sample input and Output:

Red
Green
Orange
White
Black

Sample output

List before reversing :
[Red, Green, Orange, White, Black]
List after reversing :
[Black, White, Orange, Green, Red]

Answer: (penalty regime: 0 %)

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.Scanner;
4
5 public class ReverseArrayList
6 {
7     public static void main(String[] args)
8     {
9         Scanner sc = new Scanner(System.in);
10        int n = sc.nextInt();
11        sc.nextLine();
12
13        ArrayList<String>elements = new ArrayList<>();
14
15        for(int i=0;i<n;i++)
16        {
17            String element = sc.nextLine();
18            elements.add(element);
19        }
20
21        System.out.println("List before reversing :\n" + elements);
22        Collections.reverse(elements);
23        System.out.println("List after reversing :\n" + elements);
24
25        sc.close();
26    }
27 }
```

	Test	Input	Expected	Got	
✓	1	5 Red Green Orange White Black	List before reversing : [Red, Green, Orange, White, Black] List after reversing : [Black, White, Orange, Green, Red]	List before reversing : [Red, Green, Orange, White, Black] List after reversing : [Black, White, Orange, Green, Red]	✓
✓	2	4 CSE AIML AIDS CYBER	List before reversing : [CSE, AIML, AIDS, CYBER] List after reversing : [CYBER, AIDS, AIML, CSE]	List before reversing : [CSE, AIML, AIDS, CYBER] List after reversing : [CYBER, AIDS, AIML, CSE]	✓

Passed all tests! ✓

◀ Lab-10-MCQ

Jump to...

Lab-11-MCQ ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-11-Set, Map](#) / [Lab-11-Logic Building](#)

Status Finished

Started Sunday, 17 November 2024, 4:27 PM

Completed Sunday, 17 November 2024, 4:46 PM

Duration 18 mins 53 secs

Question 1

Correct

Marked out of 1.00

Java HashSet class implements the Set interface, backed by a hash table which is actually a [HashMap](#) instance.

No guarantee is made as to the iteration order of the hash sets which means that the class does not guarantee the constant order of elements over time.

This class permits the null element.

The class also offers constant time performance for the basic operations like add, remove, contains, and size assuming the hash function disperses the elements properly among the buckets.

Java HashSet Features

A few important features of HashSet are mentioned below:

- Implements [Set Interface](#).
- The underlying data structure for HashSet is [Hashtable](#).
- As it implements the Set Interface, duplicate values are not allowed.
- Objects that you insert in HashSet are not guaranteed to be inserted in the same order. Objects are inserted based on their hash code.
- NULL elements are allowed in HashSet.
- HashSet also implements **Serializable** and **Cloneable** interfaces.

public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable
Sample Input and Output:

```
5
90
56
45
78
25
78
```

Sample Output:

```
78 was found in the set.
Sample Input and output:
```

```
3
2
7
9
5
```

Sample Input and output:

```
5 was not found in the set.
```

Answer: (penalty regime: 0 %)

Reset answer

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3 class prog {
4     public static void main(String[] args) {
5         Scanner sc= new Scanner(System.in);
6         int n = sc.nextInt();
7         HashSet<Integer>numbers=new HashSet<>();
8         // Create a HashSet object called numbers
9
10
11        // Add values to the set
12        for(int i=0;i<n;i++)
13        {
14            numbers.add(sc.nextInt());
15        }
16        int skey=sc.nextInt();
17
18        // Show which numbers between 1 and 10 are in the set
19        if (numbers.contains(skey))
20        {
21            System.out.println(skey + " was found in the set.");
22        } else {
23            System.out.println(skey + " was not found in the set.");
24        }
25    }
26}
```

	Test	Input	Expected	Got	
✓	1	5 90 56 45 78 25 78	78 was found in the set.	78 was found in the set.	✓
✓	2	3 -1 2 4 5	5 was not found in the set.	5 was not found in the set.	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 1.00

Write a Java program to compare two sets and retain elements that are the same.

Sample Input and Output:

```
5
Football
Hockey
Cricket
Volleyball
Basketball
7 // HashSet 2:
```

```
Golf
Cricket
Badminton
Football
Hockey
Volleyball
Handball
```

SAMPLE OUTPUT:

```
Football
Hockey
Cricket
Volleyball
Basketball
```

Answer: (penalty regime: 0 %)

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 public class CompareSets
5 {
6     public static void main(String[] args)
7     {
8         Scanner sc = new Scanner(System.in);
9         int n1=sc.nextInt();
10        sc.nextLine();
11        HashSet<String>set1 = new HashSet<>();
12        for(int i=0;i<n1;i++)
13        {
14            set1.add(sc.nextLine());
15        }
16        int n2=sc.nextInt();
17        sc.nextLine();
18        HashSet<String>set2=new HashSet<>();
19        for(int i=0;i<n2;i++)
20        {
21            set2.add(sc.nextLine());
22        }
23        set1.retainAll(set2);
24        for(String element:set1)
25        {
26            System.out.println(element);
27        }
28    }
29 }
```

	Test	Input	Expected	Got	
✓	1	5 Football Hockey Cricket Volleyball Basketball 7 Golf Cricket Badminton Football Hockey Volleyball Throwball	Cricket Hockey Volleyball Football	Cricket Hockey Volleyball Football	✓
✓	2	4 Toy Bus Car Auto 3 Car Bus Lorry	Bus Car	Bus Car	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

Java HashMap Methods

[containsKey\(\)](#) Indicate if an entry with the specified key exists in the map[containsValue\(\)](#) Indicate if an entry with the specified value exists in the map[putIfAbsent\(\)](#) Write an entry into the map but only if an entry with the same key does not already exist[remove\(\)](#) Remove an entry from the map[replace\(\)](#) Write to an entry in the map only if it exists[size\(\)](#) Return the number of entries in the map

Your task is to fill the incomplete code to get desired output

Answer: (penalty regime: 0 %)[Reset answer](#)

```
1 import java.util.HashMap;
2 import java.util.Map.Entry;
3 import java.util.Set;
4 import java.util.Scanner;
5 class prog
6 {
7     public static void main(String[] args)
8     {
9         //Creating HashMap with default initial capacity and load factor
10        HashMap<String, Integer> map = new HashMap<String, Integer>();
11
12        String name;
13        int num;
14        Scanner sc= new Scanner(System.in);
15        int n=sc.nextInt();
16        for(int i =0;i<n;i++)
17        {
18            name=sc.next();
19            num= sc.nextInt();
20            map.put(name,num);
21        }
22
23        //Printing key-value pairs
24
25        Set<Entry<String, Integer>> entrySet = map.entrySet();
26
27        for (Entry<String, Integer> entry : entrySet)
28        {
29            System.out.println(entry.getKey()+" : "+entry.getValue());
30        }
31        System.out.println("-----");
32        //Creating another HashMap
33
34        HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
35
36        //Inserting key-value pairs to anotherMap using put() method
37
38        anotherMap.put("SIX", 6);
39
40        anotherMap.put("SEVEN", 7);
41
42        //Inserting key-value pairs of map to anotherMap using putAll() method
43
44        anotherMap. putAll(map); // code here
45
46        //Printing key-value pairs of anotherMap
47
48        entrySet = anotherMap.entrySet();
49
50        for (Entry<String, Integer> entry : entrySet)
51        {
52            System.out.println(entry.getKey()+" : "+entry.getValue());
```

	Test	Input	Expected	Got	
✓	1	3 ONE 1 TWO ----- 2 THREE 3	ONE : 1 TWO : 2 THREE : 3 SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	ONE : 1 TWO : 2 THREE : 3 SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	✓

Passed all tests! ✓

[◀ Lab-11-MCQ](#)

Jump to...

[TreeSet example ►](#)

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-12-Introduction to I/O, I/O Operations, Object Serialization](#) / [Lab-12-Logic Building](#)

Status Finished

Started Sunday, 17 November 2024, 4:47 PM

Completed Sunday, 17 November 2024, 5:26 PM

Duration 38 mins 34 secs

Question 1

Correct

Marked out of 5.00

Given two char arrays input1[] and input2[] containing only lower case alphabets, extracts the alphabets which are present in both arrays (common alphabets).

Get the ASCII values of all the extracted alphabets.

Calculate sum of those ASCII values. Lets call it sum1 and calculate single digit sum of sum1, i.e., keep adding the digits of sum1 until you arrive at a single digit.

Return that single digit as output.

Note:

1. Array size ranges from 1 to 10.
2. All the array elements are lower case alphabets.
3. Atleast one common alphabet will be found in the arrays.

Example 1:

input1: {'a', 'b', 'c'}

input2: {'b', 'c'}

output: 8

Explanation:

'b' and 'c' are present in both the arrays.

ASCII value of 'b' is 98 and 'c' is 99.

$$98 + 99 = 197$$

$$1 + 9 + 7 = 17$$

$$1 + 7 = 8$$

For example:

Input	Result
a b c	8
b c	

Answer: (penalty regime: 0 %)

```

1 import java.util.HashSet;
2 public class CommonAlphabetsASCII
3 {
4     public static void main(String[] args)
5     {
6         String input1="abc";
7         String input2="bc";
8
9         HashSet<Character>set1 = new HashSet<>();
10        for(char ch : input1.toCharArray())
11        {
12            set1.add(ch);
13        }
14        HashSet<Character>set2 = new HashSet<>();
15        for(char ch : input2.toCharArray())
16        {
17            set2.add(ch);
18        }
19        int sum1=0;
20        for(char ch : set1)
21        {
22            if(set2.contains(ch))
23            {
24                sum1+=(int)ch;
25            }
26        }
27        int singleDigitSum = getSingleDigitSum(sum1);
28        System.out.println(singleDigitSum);
29    }
}

```

```
-- 30     private static int getSingleDigitSum(int num)
31  {
32      while(num>9)
33      {
34          int sum=0;
35          while(num>0)
36          {
37              sum+=num%10;
38              num/=10;
39          }
40          num=sum;
41      }
42      return num;
43  }
44 }
```

	Input	Expected	Got	
✓	a b c b c	8	8	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

Write a function that takes an input String (sentence) and generates a new String (modified sentence) by reversing the words in the original String, maintaining the words position.

In addition, the function should be able to control the reversing of the case (upper or lowercase) based on a `case_option` parameter, as follows:

If `case_option` = 0, normal reversal of words i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "orpiW seigoloNhceT eroLagnaB".

If `case_option` = 1, reversal of words with retaining position's case i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "Orpiw Seigolonhcet Erolagnab".

Note that positions 1, 7, 11, 20 and 25 in the original string are uppercase W, T, N, B and L.

Similarly, positions 1, 7, 11, 20 and 25 in the new string are uppercase O, S, O, E and G.

NOTE:

1. Only space character should be treated as the word separator i.e., "Hello World" should be treated as two separate words, "Hello" and "World". However, "Hello,World", "Hello;World", "Hello-World" or "Hello/World" should be considered as a single word.

2. Non-alphabetic characters in the String should not be subjected to case changes. For example, if `case option` = 1 and the original sentence is "Wipro TechNologies, Bangalore" the new reversed sentence should be "Orpiw ,seiGolonhceT Erolagnab". Note that comma has been treated as part of the word "Technologies," and when comma had to take the position of uppercase T it remained as a comma and uppercase T took the position of comma. However, the words "Wipro and Bangalore" have changed to "Orpiw" and "Erolagnab".

3. Kindly ensure that no extra (additional) space characters are embedded within the resultant reversed String.

Examples:

S. No.	input1	input2	output
1	Wipro Technologies Bangalore	0	orpiW seigolonhceT erolagnaB
2	Wipro Technologies, Bangalore	0	orpiW ,seigolonhceT erolagnaB
3	Wipro Technologies Bangalore	1	Orpiw Seigolonhcet Erolagnab
4	Wipro Technologies, Bangalore	1	Orpiw ,seigolonhceT Erolagnab

For example:

Input	Result
Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB
Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB
Wipro Technologies Bangalore 1	Orpiw Seigolonhcet Erolagnab
Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class SentenceReverser
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc = new Scanner(System.in);
7         String s = sc.nextLine();
8
9         int c=0;
10        try
11        {
12            c=sc.nextInt();
13        }
14        catch(Exception e)
15        {
16            System.out.println("ERROR: Please enter a valid integer for the mode (1 or 0).");
17        }
18    }
19 }
```

```

17     sc.close();
18     return;
19 }
20
21 String[] w = s.split(" ");
22 StringBuilder res = new StringBuilder();
23
24 for(String wd:w)
25 {
26     StringBuilder rw = new StringBuilder(wd);
27     rw.reverse();
28
29 if(c==1)
30 {
31     StringBuilder fw = new StringBuilder();
32     for(int i=0;i<wd.length();i++)
33 {
34         char r = rw.charAt(i);
35         char o = wd.charAt(i);
36
37         if(Character.isAlphabetic(o))
38 {
39             if(Character.isUpperCase(o))
40             {
41                 fw.append(Character.toUpperCase(r));
42             }
43             else
44             {
45                 fw.append(Character.toLowerCase(r));
46             }
47         }
48         else
49         {
50             fw.append(r);
51         }
52     }
}

```

	Input	Expected	Got	
✓	Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB	orpiW seigolonhceT erolagnaB	✓
✓	Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB	orpiW ,seigolonhceT erolagnaB	✓
✓	Wipro Technologies Bangalore 1	Orpiw Seigolonhcet Erolagnab	Orpiw Seigolonhcet Erolagnab	✓
✓	Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab	Orpiw ,seigolonhceT Erolagnab	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

You are provided with a string which has a sequence of 1's and 0's.

This sequence is the encoded version of a English word. You are supposed write a program to decode the provided string and find the original word.

Each alphabet is represented by a sequence of 0s.

This is as mentioned below:

z:0

Y: 00

X : 000

W : 0000

V : 00000

U : 000000

T : 0000000

and so on upto A having 26 0's (000000000000000000000000000000).

The sequence of 0's in the encoded form are separated by a single 1 which helps to distinguish between 2 letters.

Example 1:

input1: 010010001

The decoded string (original word) will be: ZYX

Example 2:

input1: 000010000000000000000010000000000010000000001000000000100000000000001

The decoded string (original word) will be: WIPRO

Note: The decoded string must always be in UPPER case

For example:

Input	Result
010010001	ZYX
000010000000000000000000000000001000000000000100000000000010000000000000001	WIPRO

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3 public class BinaryDecoder
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8
9         String encodedString = sc.nextLine();
10        String[] groups = encodedString.split("1");
11
12        StringBuilder decodedWord = new StringBuilder();
13
14        for(String group : groups)
15        {
16            if(!group.isEmpty())
17            {
18                int length=group.length();
19                char decodedChar = (char)('A'+(26-length));
20
21                decodedWord.append(decodedChar);
22            }
23        }
24        System.out.println(decodedWord.toString());
25    }
26}
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000000010000000000100000000010000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

◀ Lab-12-MCQ

Jump to...

Identify possible words ►

SOURCE CODE

3.0 BANKING SYSTEM

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.util.*;
import java.text.*;
import java.io.*;
import java.awt.PrintJob.*;
import javax.swing.plaf.metal.*;

public class BankSystem extends JFrame implements ActionListener, ItemListener {
    private JDesktopPane desktop = new JDesktopPane ();
    private JMenuBar bar;
    private JMenu mnuFile, mnuEdit, mnuView, mnuOpt, mnuWin, mnuHelp;
    private JMenuItem addNew, printRec, end;
    private JMenuItem deposit, withdraw, delRec, search, searchName;
    private JMenuItem oneByOne, allCustomer;
    private JMenuItem change, style, theme;
    private JMenuItem close, closeAll;
    private JMenuItem content, keyHelp, about;
    private JPopupMenu popMenu = new JPopupMenu ();
    private JMenuItem open, report, dep, with, del, find, all;
    private JToolBar toolBar;
    private JButton btnNew, btnDep, btnWith, btnRec, btnDel, btnSrch, btnHelp, btnKey;
    private JPanel statusBar = new JPanel ();
    private JLabel welcome;
    private JLabel author;
    private String strings[] = {"1. Metal", "2. Motif", "3. Windows"};
    private UIManager.LookAndFeelInfo looks[] = UIManager.getInstalledLookAndFeels ();
    private ButtonGroup group = new ButtonGroup ();
    private JRadioButtonMenuItem radio[] = new JRadioButtonMenuItem[strings.length];
    private java.util.Date currDate = new java.util.Date ();
    private SimpleDateFormat sdf = new SimpleDateFormat ("dd MMMM yyyy",
    Locale.getDefault ());
    private String d = sdf.format (currDate);
    private int count = 0;
    private int rows = 0;
    private int total = 0;
    private String records[][] = new String [500][6];
    private FileInputStream fis;
    private DataInputStream dis;
    public BankSystem () {
```

```
super ("BankSystem [Pvt] Limited.");
UIManager.addPropertyChangeListener (new UISwitchListener
(JComponent)getRootPane()));
bar = new JMenuBar ();
setIconImage (getToolkit().getImage ("Images/Bank.gif"));
setSize (700, 550);      setJMenuBar (bar);
addWindowListener (new WindowAdapter () {
    public void windowClosing (WindowEvent we) {
        quitApp ();
    }
});  
setLocation((Toolkit.getDefaultToolkit().getScreenSize().width - getWidth()) / 2,
(Toolkit.getDefaultToolkit().getScreenSize().height - getHeight()) / 2);
mnuFile = new JMenu ("File");
mnuFile.setMnemonic ((int)'F');
mnuEdit = new JMenu ("Edit");
mnuEdit.setMnemonic ((int)'E');
mnuView = new JMenu ("View");
mnuView.setMnemonic ((int)'V');
mnuOpt = new JMenu ("Options");
mnuOpt.setMnemonic ((int)'O');
mnuWin = new JMenu ("Window");
mnuWin.setMnemonic ((int)'W');
mnuHelp = new JMenu ("Help");
mnuHelp.setMnemonic ((int)'H');
addNew = new JMenuItem ("Open New Account", new ImageIcon
("Images/Open.gif"));
addNew.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_N,
Event.CTRL_MASK));
addNew.setMnemonic ((int)'N');
addNew.addActionListener (this);
printRec = new JMenuItem ("Print Customer Balance", new ImageIcon
("Images/New.gif"));
printRec.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_R,
Event.CTRL_MASK));
printRec.setMnemonic ((int)'R');
printRec.addActionListener (this);
end = new JMenuItem ("Quit BankSystem ?", new ImageIcon ("Images/export.gif"));
end.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_Q, Event.CTRL_MASK));
end.setMnemonic ((int)'Q');
end.addActionListener (this);
deposit = new JMenuItem ("Deposit Money");
deposit.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_T,
Event.CTRL_MASK));
deposit.setMnemonic ((int)'T');
deposit.addActionListener (this);
withdraw = new JMenuItem ("Withdraw Money");
```

```

withdraw.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_W,
Event.CTRL_MASK));
withdraw.setMnemonic ((int)'W');
withdraw.addActionListener (this);
delRec = new JMenuItem ("Delete Customer", new ImageIcon ("Images/Delete.gif"));
delRec.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_D,
Event.CTRL_MASK));
delRec.setMnemonic ((int)'D');
delRec.addActionListener (this);
search = new JMenuItem ("Search By No.", new ImageIcon ("Images/find.gif"));
search.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_S,
Event.CTRL_MASK));
search.setMnemonic ((int)'S');
search.addActionListener (this);
searchName = new JMenuItem ("Search By Name");
searchName.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_M,
Event.CTRL_MASK));
searchName.setMnemonic ((int)'M');
searchName.addActionListener (this);
oneByOne = new JMenuItem ("View One By One");
oneByOne.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_O,
Event.CTRL_MASK));
oneByOne.setMnemonic ((int)'O');
oneByOne.addActionListener (this);
allCustomer = new JMenuItem ("View All Customer", new ImageIcon
("Images/refresh.gif"));
allCustomer.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_A,
Event.CTRL_MASK));
allCustomer.setMnemonic ((int)'A');
allCustomer.addActionListener (this);
change = new JMenuItem ("Change Background Color");
change.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_B,
Event.CTRL_MASK));
change.setMnemonic ((int)'B');
change.addActionListener (this);
style = new JMenu ("Change Layout Style");
style.setMnemonic ((int)'L');
for( int i = 0; i < radio.length ; i++ ) { //Creating the subMenus of Style
Menu.
radio[i] = new JRadioButtonMenuItem (strings[i]); //Build an Array of Layouts to
Apply.
radio[i].addItemListener (this); //Setting their Actions.
group.add (radio[i]); //Making them Grouped.
style.add (radio[i]); //Adding to Style
MenuOption.
}

```

```

MetalTheme[] themes = { new DefaultMetalTheme(), new GreenTheme(), new
AquaTheme(),
                     new SandTheme(), new SolidTheme(), new MilkyTheme(), new
GrayTheme() };
theme = new MetalThemeMenu ("Apply Theme", themes); //Putting the
Themes in ThemeMenu.
theme.setMnemonic ((int)'M');
close = new JMenuItem ("Close Active Window");
close.setMnemonic ((int)'C');
close.addActionListener (this);
closeAll = new JMenuItem ("Close All Windows...");
closeAll.setMnemonic ((int)'A');
closeAll.addActionListener (this);
content = new JMenuItem ("Help Contents", new ImageIcon ("Images/paste.gif"));
content.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_H,
Event.CTRL_MASK));
content.setMnemonic ((int)'H');
content.addActionListener (this);
keyHelp = new JMenuItem ("Help on Shortcuts...");
keyHelp.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_K,
Event.CTRL_MASK));
keyHelp.setMnemonic ((int)'K');
keyHelp.addActionListener (this);
about = new JMenuItem ("About BankSystem", new ImageIcon ("Images/Save.gif"));
about.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_C,
Event.CTRL_MASK));
about.setMnemonic ((int)'C'); about.addActionListener (this);
mnuFile.add (addNew);
mnuFile.add (printRec);
mnuFile.add (end);
mnuEdit.add (withdraw);
mnuEdit.add (delRec);
mnuEdit.add (search);
mnuView.add (oneByOne);
mnuView.add (allCustomer);
mnuOpt.addSeparator ();
mnuOpt.add (theme);
mnuWin.add (closeAll);
mnuHelp.addSeparator ();
mnuHelp.addSeparator ();
bar.add (mnuFile);
bar.add (mnuView);
bar.add (mnuWin);
open = new JMenuItem ("Open New Account", new ImageIcon ("Images/Open.gif"));
open.addActionListener (this);
report = new JMenuItem ("Print BankSystem Report", new ImageIcon
("Images/New.gif"));

```

```

report.addActionListener (this);
dep = new JMenuItem ("Deposit Money");
dep.addActionListener (this);
with = new JMenuItem ("Withdraw Money");
with.addActionListener (this);
del = new JMenuItem ("Delete Customer", new ImageIcon ("Images/Delete.gif"));
del.addActionListener (this);
find = new JMenuItem ("Search Customer", new ImageIcon ("Images/find.gif"));
find.addActionListener (this);
all = new JMenuItem ("View All Customer", new ImageIcon ("Images/refresh.gif"));
all.addActionListener (this);
popMenu.add (open);           popMenu.add (report);
popMenu.add (dep);           popMenu.add (with);
popMenu.add (del);           popMenu.add (find);
popMenu.add (all);
addMouseListener (new MouseAdapter () {
    public void mousePressed (MouseEvent me) { checkMouseTrigger (me); }
    public void mouseReleased (MouseEvent me) { checkMouseTrigger (me); }
    private void checkMouseTrigger (MouseEvent me) {
        if (me.isPopupTrigger ())
            popMenu.show (me.getComponent (), me.getX (), me.getY ());
    }
});

```

```

btnNew = new JButton (new ImageIcon ("Images>NotePad.gif"));
btnNew.setToolTipText ("Create New Account");
btnNew.addActionListener (this);
btnDep = new JButton (new ImageIcon ("Images>ImationDisk.gif"));
btnDep.setToolTipText ("Deposit Money");
btnDep.addActionListener (this);
btnWith = new JButton (new ImageIcon ("Images>SuperDisk.gif"));
btnWith.setToolTipText ("Withdraw Money");
btnWith.addActionListener (this);
btnRec = new JButton (new ImageIcon ("Images>Paproll.gif"));
btnRec.setToolTipText ("Print Customer Balance");
btnRec.addActionListener (this);
btnDel = new JButton (new ImageIcon ("Images>Toaster.gif"));
btnDel.setToolTipText ("Delete Customer");
btnDel.addActionListener (this);
btnSrch = new JButton (new ImageIcon ("Images>Search.gif"));
btnSrch.setToolTipText ("Search Customer");
btnSrch.addActionListener (this);
btnHelp = new JButton (new ImageIcon ("Images>Help.gif"));
btnHelp.setToolTipText ("Help on Bank System");
btnHelp.addActionListener (this);
btnKey = new JButton (new ImageIcon ("Images>Keys.gif"));
btnKey.setToolTipText ("Shortcut Keys of BankSystem");
btnKey.addActionListener (this);

```



```

        WithdrawMoney withMon = new WithdrawMoney ();
        desktop.add (withMon);           withMon.show ();
    }
}

else if (obj == delRec || obj == del || obj == btnDel) {

    boolean b = openChildWindow ("Delete Account Holder");
    if (b == false) {
        DeleteCustomer delCus = new DeleteCustomer ();
        desktop.add (delCus);   delCus.show ();
    }
}

else if (obj == search || obj == find || obj == btnSrch) {

    boolean b = openChildWindow ("Search Customer [By No.]");
    if (b == false) {
        FindAccount fndAcc = new FindAccount ();
        desktop.add (fndAcc);   fndAcc.show ();
    }
}

else if (obj == searchName) {

    boolean b = openChildWindow ("Search Customer [By Name]");
    if (b == false) {
        FindName fndNm = new FindName ();
        desktop.add (fndNm);   fndNm.show ();
    }
}

else if (obj == oneByOne) {

    boolean b = openChildWindow ("View Account Holders");
    if (b == false) {
        ViewOne vwOne = new ViewOne ();
        desktop.add (vwOne);   vwOne.show ();
    }
}

else if (obj == allCustomer || obj == all) {

    boolean b = openChildWindow ("View All Account Holders");
    if (b == false) {
        ViewCustomer viewCus = new ViewCustomer ();
        desktop.add (viewCus);   viewCus.show ();
    }
}

else if (obj == change) {

    Color cl = new Color (153, 153, 204);
}

```

```

cl = JColorChooser.showDialog (this, "Choose Background Color", cl);
if (cl == null) { }
else {
    desktop.setBackground (cl);    desktop.repaint ();
}
else if (obj == close) {

try {
    desktop.getSelectedFrame().setClosed(true);
}
catch (Exception CloseExc) { }

}
else if (obj == closeAll) {

JInternalFrame Frames[] = desktop.getAllFrames () //Getting all Open Frames.
for(int getFrameLoop = 0; getFrameLoop < Frames.length; getFrameLoop++) {
    try {
        Frames[getFrameLoop].setClosed (true); //Close the frame.
    }
    catch (Exception CloseExc) { } //if we can't close it then we have a
problem.
}
}
else if (obj == content || obj == btnHelp) {

boolean b = openChildWindow ("BankSystem Help");
if (b == false) {
    BankHelp hlpBank = new BankHelp ("BankSystem Help",
"Help/Bank.htm");
    desktop.add (hlpBank);
    hlpBank.show ();
}
}
else if (obj == keyHelp || obj == btnKey) {

boolean b = openChildWindow ("BankSystem Keys");
if (b == false) {
    BankHelp hlpKey = new BankHelp ("BankSystem Keys",
"Help/Keys.htm");
    desktop.add (hlpKey);
    hlpKey.show ();
}
}
else if (obj == about) {

String msg = "BankSystem [Pvt] Limited.\n\n" + "Created & Designed By:\n" +
"Muhammad Wasif Javed\n\n" + "E-mail me:\n"
}
}

```

```

wasi_javed@hotmail.com";
JOptionPane.showMessageDialog (this, msg, "About BankSystem",
JOptionPane.PLAIN_MESSAGE);

}
}

public void itemStateChanged (ItemEvent e) {

for( int i = 0; i < radio.length; i++ )
    if(radio[i].isSelected()) {
        changeLookAndFeel (i);
    }
}

private void quitApp () {
try {
    //Show a Confirmation Dialog.
    int reply = JOptionPane.showConfirmDialog (this,
        "Are you really want to exit\nFrom BankSystem?",
        "BankSystem - Exit", JOptionPane.YES_NO_OPTION,
JOptionPane.PLAIN_MESSAGE);
    //Check the User Selection.
    if (reply == JOptionPane.YES_OPTION) {
        setVisible (false); //Hide the Frame.
        dispose();           //Free the System Resources.
        System.out.println ("Thanks for Using BankSystem\nAuthor - Muhammad
Wasif Javed");
        System.exit (0);     //Close the Application.
    }
    else if (reply == JOptionPane.NO_OPTION) {
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
    }
}

catch (Exception e) {}

}

public void changeLookAndFeel (int val) {

try {
    UIManager.setLookAndFeel (looks[val].getClassName());
    SwingUtilities.updateComponentTreeUI (this);
}
catch (Exception e) {} }

}

private boolean openChildWindow (String title) {
JInternalFrame[] childs = desktop.getAllFrames ();
for (int i = 0; i < childs.length; i++) {

```

```

        if (childs[i].getTitle().equalsIgnoreCase (title)) {
            childs[i].show ();
            return true;
        }
    }
    return false;
}
void getAccountNo () {
    String printing;
    rows = 0;
    boolean b = populateArray ();
    if (b == false) { }
    else {
        try {
            printing = JOptionPane.showInputDialog (this, "Enter Account No. to
Print Customer Balance.\n" +
                "(Tip: Account No. Contains only Digits)", "BankSystem - PrintRecord",
                JOptionPane.PLAIN_MESSAGE);
            if (printing == null) { }
            if (printing.equals ("")) {
                JOptionPane.showMessageDialog (this, "Provide Account No. to
Print.", "BankSystem - EmptyField",
                JOptionPane.PLAIN_MESSAGE);
                getAccountNo ();
            }
            else {
                findRec (printing);
            }
        }
        catch (Exception e) { }
    }
}
boolean populateArray () {
    boolean b = false;
    try {
        fis = new FileInputStream ("Bank.dat");
        dis = new DataInputStream (fis);
        //Loop to Populate the Array.
        while (true) {
            for (int i = 0; i < 6; i++) {
                records[rows][i] = dis.readUTF ();
            }
            rows++;
        }
    }
    catch (Exception ex) {
        total = rows;
        if (total == 0) {

```

```

        JOptionPane.showMessageDialog (null, "Records File is Empty.\nEnter
Records First to Display.",
                                "BankSystem - EmptyFile", JOptionPane.PLAIN_MESSAGE);
        b = false;
    }
    else {
        b = true;
        try {
            dis.close();
            fis.close();
        }
        catch (Exception exp) { }
    }
}
return b;
}

}

```

//Function use to Find Record by Matching the Contents of Records Array with
InputBox.

```

void findRec (String rec) {

boolean found = false;
for (int x = 0; x < total; x++) {
    if (records[x][0].equals (rec)) {
        found = true;
        printRecord (makeRecordPrint (x));
        break;
    }
}
if (found == false) {
    JOptionPane.showMessageDialog (this, "Account No. " + rec + " doesn't Exist.",
                                "BankSystem - WrongNo", JOptionPane.PLAIN_MESSAGE);
    getAccountNo ();
}
}

}

```

//Function use to make Current Record ready for Print.

```

String makeRecordPrint (int rec) {

String data;
String data0 = "      BankSystem [Pvt] Limited.          \n";      //Page Title.
String data1 = "      Customer Balance Report.      \n\n";   //Page Header.
String data2 = " Account No.:    " + records[rec][0] + "\n";

```

```

String data3 = " Customer Name: " + records[rec][1] + "\n";
String data4 = " Last Transaction: " + records[rec][2] + ", " + records[rec][3] + ", " +
records[rec][4] + "\n";
String data5 = " Current Balance: " + records[rec][5] + "\n\n";
String data6 = " Copyright 2003 Muhammad Javed.\n"; //Page Footer.
String sep0 = " -----\n";
String sep1 = " -----\n";
String sep2 = " -----\n";
String sep3 = " -----\n";
String sep4 = " -----\n\n";

data = data0 + sep0 + data1 + data2 + sep1 + data3 + sep2 + data4 + sep3 + data5 +
sep4 + data6;
return data;

}

```

//Function use to Print the Current Record.

```

void printRecord (String rec) {

StringReader sr = new StringReader (rec);
LineNumberReader lnr = new LineNumberReader (sr);
Font typeface = new Font ("Times New Roman", Font.PLAIN, 12);
Properties p = new Properties ();
PrintJob pJob = getToolkit().getPrintJob (this, "Print Customer Balance Report", p);

if (pJob != null) {
    Graphics gr = pJob.getGraphics ();
    if (gr != null) {
        FontMetrics fm = gr.getFontMetrics (typeface);
        int margin = 20;
        int pageHeight = pJob.getPageDimension().height - margin;
        int fontHeight = fm.getHeight();
            int fontDescent = fm.getDescent();
        int curHeight = margin;
        String nextLine;
        gr.setFont (typeface);

        try {
            do {
                nextLine = lnr.readLine ();
                if (nextLine != null) {
                    if ((curHeight + fontHeight) > pageHeight) {//New
Page.
                        gr.dispose();
                        gr = pJob.getGraphics ();

```

```

                curHeight = margin;
            }
            curHeight += fontHeight;
            if (gr != null) {
                gr.setFont (typeface);
                gr.drawString (nextLine, margin, curHeight -
fontDescent);
            }
        }
    }
    while (nextLine != null);
}
catch (EOFException eof) { }
catch (Throwable t) { }
}
gr.dispose();
}
if (pJob != null)
    pJob.end ();
}
}

```

3.1 CREATE NEW ACCOUNT

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

```

```

public class NewAccount extends JInternalFrame implements ActionListener {
    private JPanel jpInfo = new JPanel();
    private JLabel lbNo, lbName, lbDate, lbDeposit;
    private JTextField txtNo, txtName, txtDeposit;
    private JComboBox cboMonth, cboDay, cboYear;
    private JButton btnSave, btnCancel;
    private int count = 0;    private int rows = 0;
    private int total = 0;
    private String records[][] = new String [500][6];
    private String saves[][] = new String [500][6];
    private FileInputStream fis;
    private DataInputStream dis;
    NewAccount () {
        super ("Create New Account", false, true, false, true);
        setSize (335, 235);
        jpInfo.setBounds (0, 0, 500, 115);
        jpInfo.setLayout (null);
        lbNo = new JLabel ("Account No:");
        lbNo.setForeground (Color.black);
        lbNo.setBounds (15, 20, 80, 25);

```

```

lbName = new JLabel ("Person Name:");
lbName.setForeground (Color.black);
lbName.setBounds (15, 55, 80, 25);
lbDate = new JLabel ("Deposit Date:");
lbDate.setForeground (Color.black);
lbDate.setBounds (15, 90, 80, 25);
lbDeposit = new JLabel ("Dep. Amount:");
lbDeposit.setForeground (Color.black);
lbDeposit.setBounds (15, 125, 80, 25);
txtNo = new JTextField ();
txtNo.setHorizontalAlignment (JTextField.RIGHT);
txtNo.setBounds (105, 20, 205, 25);
txtName = new JTextField ();
txtName.setBounds (105, 55, 205, 25);
txtDeposit = new JTextField ();
txtDeposit.setHorizontalAlignment (JTextField.RIGHT);
txtDeposit.setBounds (105, 125, 205, 25);
txtNo.addKeyListener (new KeyAdapter() {
    public void keyTyped (KeyEvent ke) {
        char c = ke.getKeyChar ();
        if (!((Character.isDigit (c) || (c ==
KeyEvent.VK_BACK_SPACE)))) {
            Toolkit.beep ();
            ke.consume ();
        }
    }
});
txtDeposit.addKeyListener (new KeyAdapter() {
    public void keyTyped (KeyEvent ke) {
        char c = ke.getKeyChar ();
        if (!((Character.isDigit (c) || (c ==
KeyEvent.VK_BACK_SPACE)))) {
            Toolkit.beep ();
            ke.consume ();
        }
    }
});
String Months[] = {"January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December"};
cboMonth = new JComboBox (Months);
cboDay = new JComboBox ();
cboYear = new JComboBox ();
for (int i = 1; i <= 31; i++) {
    String days = "" + i;
    cboDay.addItem (days);
}
for (int i = 2000; i <= 2024; i++) {

```

```

        String years = "" + i;
        cboYear.addItem (years);
    }
cboMonth.setBounds (105, 90, 92, 25);
cboDay.setBounds (202, 90, 43, 25);
cboYear.setBounds (250, 90, 60, 25);
btnSave = new JButton ("Save");
btnSave.setBounds (20, 165, 120, 25);
btnSave.addActionListener (this);
btnCancel = new JButton ("Cancel");
btnCancel.setBounds (185, 165, 120, 25);
btnCancel.addActionListener (this);
jpInfo.add (lbNo);
jpInfo.add (txtNo);
jpInfo.add (lbName);
jpInfo.add (txtName);
jpInfo.add (lbDate);
jpInfo.add (cboMonth);
jpInfo.add (cboDay);
jpInfo.add (cboYear);
jpInfo.add (lbDeposit);
jpInfo.add (txtDeposit);
jpInfo.add (btnSave);
jpInfo.add (btnCancel);
getContentPane().add (jpInfo);
setVisible (true);
}
public void actionPerformed (ActionEvent ae) {
    Object obj = ae.getSource();
    if (obj == btnSave) {
        if (txtNo.getText().equals("")) {
            JOptionPane.showMessageDialog (this, "Please! Provide Id of
Customer.",
                                         "BankSystem - EmptyField",
                                         JOptionPane.PLAIN_MESSAGE);
            txtNo.requestFocus();
        }
        else if (txtName.getText().equals("")) {
            JOptionPane.showMessageDialog (this, "Please! Provide Name of
Customer.",
                                         "BankSystem - EmptyField",
                                         JOptionPane.PLAIN_MESSAGE);
            txtName.requestFocus ();
        }
        else if (txtDeposit.getText().equals("")) {
            JOptionPane.showMessageDialog (this, "Please! Provide Deposit
Amount.",
```

```

        "BankSystem - EmptyField",
JOptionPane.PLAIN_MESSAGE);
            txtDeposit.requestFocus ();
        }
    else {
        populateArray (); //Load All Existing Records in Memory.
        findRec (); //Finding if Account No. Already Exist or Not.
    }
}
if (obj == btnCancel) {
    txtClear ();
    setVisible (false);
    dispose();
}
}

void populateArray () {
try {
    fis = new FileInputStream ("Bank.dat");
    dis = new DataInputStream (fis);
    //Loop to Populate the Array.
    while (true) {
        for (int i = 0; i < 6; i++) {
            records[rows][i] = dis.readUTF ();
        }
        rows++;
    }
}
catch (Exception ex) {
    total = rows;
    if (total == 0) { }
    else {
        try {
            dis.close();
            fis.close();
        }
        catch (Exception exp) { }
    }
}
}

void findRec () {

boolean found = false;
for (int x = 0; x < total; x++) {
    if (records[x][0].equals (txtNo.getText())) {
        found = true;
        JOptionPane.showMessageDialog (this, "Account No. " +
txtNo.getText () + " is Already Exist.");
    }
}
}

```

```

        "BankSystem - WrongNo",
JOptionPane.PLAIN_MESSAGE);
            txtClear ();
            break;
        }
    }
    if (found == false) {
        saveArray ();
    }
}

//Function use to add new Element to Array.
void saveArray () {

    saves[count][0] = txtNo.getText ();
    saves[count][1] = txtName.getText ();
    saves[count][2] = "" + cboMonth.getSelectedItem ();
    saves[count][3] = "" + cboDay.getSelectedItem ();
    saves[count][4] = "" + cboYear.getSelectedItem ();
    saves[count][5] = txtDeposit.getText ();
    saveFile (); //Save This Array to File.
    count++;
}

void saveFile () {
    try {
        FileOutputStream fos = new FileOutputStream ("Bank.dat", true);
        DataOutputStream dos = new DataOutputStream (fos);
        dos.writeUTF (saves[count][0]);
        dos.writeUTF (saves[count][1]);
        dos.writeUTF (saves[count][2]);
        dos.writeUTF (saves[count][3]);
        dos.writeUTF (saves[count][4]);
        dos.writeUTF (saves[count][5]);
        JOptionPane.showMessageDialog (this, "The Record has been Saved
Successfully",
        "BankSystem - Record Saved",
JOptionPane.PLAIN_MESSAGE);
            txtClear ();
            dos.close();
            fos.close();
        }
    catch (IOException ioe) {
        JOptionPane.showMessageDialog (this, "There are Some Problem with
File",
        "BankSystem - Problem",

```

```

JOptionPane.PLAIN_MESSAGE);
}

}

//Function use to Clear all TextFields of Window.
void txtClear () {

    txtNo.setText ("");
    txtName.setText ("");
    txtDeposit.setText ("");
    txtNo.requestFocus ();

}
}

```

3.2 DELETE CUSTOMERS

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class DeleteCustomer extends JInternalFrame implements ActionListener {

    private JPanel jpDel = new JPanel();
    private JLabel lbNo, lbName, lbDate, lbBal;
    private JTextField txtNo, txtName, txtDate, txtBal;
    private JButton btnDel, btnCancel;
    private int recCount = 0;
    private int rows = 0;
    private int total = 0;
    private String records[][] = new String [500][6];
    private FileInputStream fis;
    private DataInputStream dis;
    DeleteCustomer () {
        super ("Delete Account Holder", false, true, false, true);
        setSize (350, 235);
        jpDel.setLayout (null);
        lbNo = new JLabel ("Account No:");
        lbNo.setForeground (Color.black);
        lbNo.setBounds (15, 20, 80, 25);
        lbName = new JLabel ("Person Name:");
        lbName.setForeground (Color.black);
        lbName.setBounds (15, 55, 90, 25);
        lbDate = new JLabel ("Last Transaction:");
        lbDate.setForeground (Color.black);
        lbDate.setBounds (15, 90, 100, 25);
        lbBal = new JLabel ("Balance:");

```

```

lbBal.setForeground (Color.black);
lbBal.setBounds (15, 125, 80, 25);
txtNo = new JTextField ();
txtNo.setHorizontalAlignment (JTextField.RIGHT);
txtNo.setBounds (125, 20, 200, 25);
txtName = new JTextField ();
txtName.setEnabled (false);
txtName.setBounds (125, 55, 200, 25);
txtDate = new JTextField ();
txtDate.setEnabled (false);
txtDate.setBounds (125, 90, 200, 25);
txtBal = new JTextField ();
txtBal.setEnabled (false);
txtBal.setHorizontalAlignment (JTextField.RIGHT);
txtBal.setBounds (125, 125, 200, 25);
btnDel = new JButton ("Delete");
btnDel.setBounds (20, 165, 120, 25);
btnDel.addActionListener (this);
btnCancel = new JButton ("Cancel");
btnCancel.setBounds (200, 165, 120, 25);
btnCancel.addActionListener (this);
jpDel.add (lbNo);      jpDel.add (txtNo);
jpDel.add (lbName);    jpDel.add (txtName);
jpDel.add (lbDate);    jpDel.add (txtDate);
jpDel.add (lbBal);     jpDel.add (txtBal);
jpDel.add (btnDel);    jpDel.add (btnCancel);
txtNo.addKeyListener (new KeyAdapter() {
    public void keyTyped (KeyEvent ke) {
        char c = ke.getKeyChar ();
        if (!((Character.isDigit (c) || (c == KeyEvent.VK_BACK_SPACE)))) {
            Toolkit().beep ();
            ke.consume ();
        }
    }
});
txtNo.addFocusListener (new FocusListener () {
    public void focusGained (FocusEvent e) { }
    public void focusLost (FocusEvent fe) {
        if (txtNo.getText().equals ("")) { }
        else {
            rows = 0;
            populateArray ();  findRec ();
        }
    }
});
getContentPane().add (jpDel);
populateArray (); //Load All Existing Records in Memory.
setVisible (true); }

```

```

public void actionPerformed (ActionEvent ae) {
    Object obj = ae.getSource();
    if (obj == btnDel) {
        if (txtNo.getText().equals("")) {
            JOptionPane.showMessageDialog (this, "Please! Provide Id of Customer.",
                                         "BankSystem - EmptyField",
                                         JOptionPane.PLAIN_MESSAGE);
            txtNo.requestFocus();
        }
        else {
            deletion (); //Confirm Deletion of Current Record.
        }
    }
    if (obj == btnCancel) {
        txtClear ();
        setVisible (false);
        dispose();
    }
}
void populateArray () {
    try {
        fis = new FileInputStream ("Bank.dat");
        dis = new DataInputStream (fis);
        //Loop to Populate the Array.
        while (true) {
            for (int i = 0; i < 6; i++) {
                records[rows][i] = dis.readUTF ();
            }
            rows++;
        }
    }
    catch (Exception ex) {
        total = rows;
        if (total == 0) {
            JOptionPane.showMessageDialog (null, "Records File is
Empty.\nEnter Records First to Display.",
                                         "BankSystem - EmptyFile",
                                         JOptionPane.PLAIN_MESSAGE);
            btnEnable ();
        }
        else {
            try {
                dis.close();
                fis.close();
            }
            catch (Exception exp) { }
        }
    }
}

```

```

void findRec () {
    boolean found = false;
    for (int x = 0; x < total; x++) {
        if (records[x][0].equals (txtNo.getText())) {
            found = true;
            showRec (x);
            break;
        }
    }
    if (found == false) {
        String str = txtNo.getText ();
        txtClear ();
        JOptionPane.showMessageDialog (this, "Account No. " + str + " doesn't
Exist.",

"BankSystem - WrongNo",
JOptionPane.PLAIN_MESSAGE);
    }
}

//Function which display Record from Array onto the Form.
void showRec (int intRec) {

    txtNo.setText (records[intRec][0]);
    txtName.setText (records[intRec][1]);
    txtDate.setText (records[intRec][2] + ", " + records[intRec][3] + ", " +
records[intRec][4]);
    txtBal.setText (records[intRec][5]);
    recCount = intRec;

}

//Confirming the Deletion Decision made By User of Program.
void deletion () {

    try {
        //Show a Confirmation Dialog.
        int reply = JOptionPane.showConfirmDialog (this,
            "Are you Sure you want to Delete\n" + txtName.getText () +
"Record From BankSystem?",

            "Bank System - Delete", JOptionPane.YES_NO_OPTION,
JOptionPane.PLAIN_MESSAGE);
        //Check the User Selection.
        if (reply == JOptionPane.YES_OPTION) {
            delRec (); //Delete the Selected Contents of Array.
        }
        else if (reply == JOptionPane.NO_OPTION) { }
    }
}

```

```

        }

    catch (Exception e) {}

}

//Function use to Delete an Element from the Array.
void delRec () {

    try {
        if (records != null) {
            for(int i = recCount; i < total; i++) {
                for (int r = 0; r < 6; r++) {
                    records[i][r] = records[i+1][r];
                    if (records[i][r] == null) break;
                }
            }
            total = total - 1;
            deleteFile ();
        }
    }
    catch (ArrayIndexOutOfBoundsException ex) { }

}

//Function use to Save Records to File After Deleting the Record of User Choice.
void deleteFile () {

    try {
        FileOutputStream fos = new FileOutputStream ("Bank.dat");
        DataOutputStream dos = new DataOutputStream (fos);
        if (records != null) {
            for (int i = 0; i < total; i++) {
                for (int r = 0; r < 6; r++) {
                    dos.writeUTF (records[i][r]);
                    if (records[i][r] == null) break;
                }
            }
        }
        JOptionPane.showMessageDialog (this, "Record has been Deleted
Successfully.", "BankSystem - Record Deleted",
JOptionPane.PLAIN_MESSAGE);
        txtClear ();
    }
    else { }
    dos.close();
    fos.close();
}

```

```

        }
    catch (IOException ioe) {
        JOptionPane.showMessageDialog (this, "There are Some Problem with
File",
                                         "BankSystem - Problem",
JOptionPane.PLAIN_MESSAGE);
    }

}

//Function use to Clear all TextFields of Window.
void txtClear () {

    txtNo.setText ("");
    txtName.setText ("");
    txtDate.setText ("");
    txtBal.setText ("");
    txtNo.requestFocus ();

}

//Function use to Lock Controls of Window.
void btnEnable () {

    txtNo.setEnabled (false);
    btnDel.setEnabled (false);

}

```

3.3DEPOSIT MONEY

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class DepositMoney extends JInternalFrame implements ActionListener {
    private JPanel jpDep = new JPanel();
    private JLabel lbNo, lbName, lbDate, lbDeposit;
    private JTextField txtNo, txtName, txtDeposit;
    private JComboBox cboMonth, cboDay, cboYear;
    private JButton btnSave, btnCancel;
    private int recCount = 0;
    private int rows = 0;
    private int total = 0;
    private int curr;
    private int deposit;

```

```
private String records[][] = new String [500][6];
private FileInputStream fis;
private DataInputStream dis;
DepositMoney () {
    super ("Deposit Money", false, true, false, true);
    setSize (335, 235);
    jpDep.setLayout (null);
    lbNo = new JLabel ("Account No:");
    lbNo.setForeground (Color.black);
    lbNo.setBounds (15, 20, 80, 25);
    lbName = new JLabel ("Person Name:");
    lbName.setForeground (Color.black);
    lbName.setBounds (15, 55, 80, 25);
    lbDate = new JLabel ("Deposit Date:");
    lbDate.setForeground (Color.black);
    lbDate.setBounds (15, 90, 80, 25);
    lbDeposit = new JLabel ("Dep. Amount:");
    lbDeposit.setForeground (Color.black);
    lbDeposit.setBounds (15, 125, 80, 25);
    txtNo = new JTextField ();
    txtNo.setHorizontalAlignment (JTextField.RIGHT);
    txtNo.setBounds (105, 20, 205, 25);
    txtName = new JTextField ();
    txtName.setEnabled (false);
    txtName.setBounds (105, 55, 205, 25);
    txtDeposit = new JTextField ();
    txtDeposit.setHorizontalAlignment (JTextField.RIGHT);
    txtDeposit.setBounds (105, 125, 205, 25);
    String Months[] = { "January", "February", "March", "April", "May", "June",
        "July", "August", "September", "October", "November", "December" };
    cboMonth = new JComboBox (Months);
    cboDay = new JComboBox ();
    cboYear = new JComboBox ();
    for (int i = 1; i <= 31; i++) {
        String days = "" + i;
        cboDay.addItem (days);
    }
    for (int i = 2000; i <= 2024; i++) {
        String years = "" + i;
        cboYear.addItem (years);
    }
    cboMonth.setBounds (105, 90, 92, 25);
    cboDay.setBounds (202, 90, 43, 25);
    cboYear.setBounds (250, 90, 60, 25);
    btnSave = new JButton ("Save");
    btnSave.setBounds (20, 165, 120, 25);
    btnSave.addActionListener (this);
```

```

btnCancel = new JButton ("Cancel");
btnCancel.setBounds (185, 165, 120, 25);
btnCancel.addActionListener (this);
txtNo.addKeyListener (new KeyAdapter() {
    public void keyTyped (KeyEvent ke) {
        char c = ke.getKeyChar ();
        if (!((Character.isDigit (c) || (c == KeyEvent.VK_BACK_SPACE)))) {
            Toolkit().beep ();
            ke.consume ();
        }
    }
});
txtDeposit.addKeyListener (new KeyAdapter() {
    public void keyTyped (KeyEvent ke) {
        char c = ke.getKeyChar ();
        if (!((Character.isDigit (c) || (c == KeyEvent.VK_BACK_SPACE)))) {
            Toolkit().beep ();
            ke.consume ();
        }
    }
});
//Checking the Account No. Provided By User on Lost Focus of the TextBox.
txtNo.addFocusListener (new FocusListener () {
    public void focusGained (FocusEvent e) { }
    public void focusLost (FocusEvent fe) {
        if (txtNo.getText().equals ("")) { }
        else {
            rows = 0;
            populateArray (); //Load All Existing Records in Memory.
            findRec (); //Finding if Account No. Already Exist or Not.
        }
    }
});
//Adding the All the Controls to Panel.
jpDep.add (lbNo);      jpDep.add (txtNo);
jpDep.add (lbName);    jpDep.add (txtName);
jpDep.add (lbDate);    jpDep.add (cboMonth);
jpDep.add (cboDay);    jpDep.add (cboYear);
jpDep.add (lbDeposit); jpDep.add (txtDeposit);
jpDep.add (btnSave);   jpDep.add (btnCancel);
getContentPane().add (jpDep);
populateArray (); setVisible (true);      }

```

```

public void actionPerformed (ActionEvent ae) {
    Object obj = ae.getSource();
    if (obj == btnSave) {
        if (txtNo.getText().equals("")) {
            JOptionPane.showMessageDialog (this, "Please! Provide Id of Customer.",
                "BankSystem - EmptyField",
                JOptionPane.PLAIN_MESSAGE);
            txtNo.requestFocus();
        }
        else if (txtDeposit.getText().equals("")) {
            JOptionPane.showMessageDialog (this, "Please! Provide Deposit Amount.",
                "BankSystem - EmptyField",
                JOptionPane.PLAIN_MESSAGE);
            txtDeposit.requestFocus ();
        }
        else {
            editRec (); //Update the Contents of Array.
        }
    }
    if (obj == btnCancel) {
        txtClear (); setVisible (false); dispose();
    }
}
void populateArray () {
    try {
        fis = new FileInputStream ("Bank.dat");
        dis = new DataInputStream (fis);
        //Loop to Populate the Array.
        while (true) {
            for (int i = 0; i < 6; i++) {
                records[rows][i] = dis.readUTF ();
            }
            rows++;
        }
    }
    catch (Exception ex) {
        total = rows;
        if (total == 0) {
            JOptionPane.showMessageDialog (null, "Records File is Empty.\nEnter
Records First to Display.",
                "BankSystem - EmptyFile",
                JOptionPane.PLAIN_MESSAGE);
            btnEnable ();
        }
        else {
            try {
                dis.close();

```

```

        fis.close();
    }
    catch (Exception exp) { }
}
}

void findRec () {
boolean found = false;
for (int x = 0; x < total; x++) {
    if (records[x][0].equals (txtNo.getText())) {
        found = true;
        showRec (x);
        break;
    }
}
if (found == false) {
    String str = txtNo.getText ();
    txtClear ();
    JOptionPane.showMessageDialog (this, "Account No. " + str + " doesn't Exist.",
                                  "BankSystem - WrongNo",
                                  JOptionPane.PLAIN_MESSAGE);
}
}

public void showRec (int intRec) {
txtNo.setText (records[intRec][0]);
txtName.setText (records[intRec][1]);
curr = Integer.parseInt (records[intRec][5]);
recCount = intRec;
}

void txtClear () {
txtNo.setText ("");
txtName.setText ("");
txtDeposit.setText ("");
txtNo.requestFocus ();
}

public void editRec () {
deposit = Integer.parseInt (txtDeposit.getText ());
records[recCount][0] = txtNo.getText ();
records[recCount][1] = txtName.getText ();
records[recCount][2] = "" + cboMonth.getSelectedItem ();
records[recCount][3] = "" + cboDay.getSelectedItem ();
records[recCount][4] = "" + cboYear.getSelectedItem ();
records[recCount][5] = "" + (curr + deposit);
editFile (); //Save This Array to File.
}

}

```

```

public void editFile () {
    try {
        FileOutputStream fos = new FileOutputStream ("Bank.dat");
        DataOutputStream dos = new DataOutputStream (fos);
        if (records != null) {
            for (int i = 0; i < total; i++) {
                for (int c = 0; c < 6; c++) {
                    dos.writeUTF (records[i][c]);
                    if (records[i][c] == null) break;
                }
            }
            JOptionPane.showMessageDialog (this, "The File is Updated
Successfully",
                                         "BankSystem - Record Saved",
                                         JOptionPane.PLAIN_MESSAGE);
            txtClear ();
            dos.close();
            fos.close();
        }
    } catch (IOException ioe) {
        JOptionPane.showMessageDialog (this, "There are Some Problem with File",
                                     "BankSystem - Problem", JOptionPane.PLAIN_MESSAGE);
    }
}

void btnEnable () {
    txtNo.setEnabled (false);
    cboMonth.setEnabled (false);
    cboDay.setEnabled (false);
    cboYear.setEnabled (false);
    txtDeposit.setEnabled (false);
    btnSave.setEnabled (false);
}

}}

```

3.4 FINDING AN ACCOUNT

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
public class FindAccount extends JInternalFrame implements ActionListener {
    private JPanel jpFind = new JPanel();
    private JLabel lbNo, lbName, lbDate, lbBal;
    private JTextField txtNo, txtName, txtDate, txtBal;
    private JButton btnFind, btnCancel;
    private int count = 0;
    private int rows = 0;

```

```

private      int total = 0;
private String records[][] = new String [500][6];
private FileInputStream fis;
private DataInputStream dis;
FindAccount () {
    super ("Search Customer [By No.]", false, true, false, true);
    setSize (350, 235);
    jpFind.setLayout (null);
    lbNo = new JLabel ("Account No:");
    lbNo.setForeground (Color.black);
    lbNo.setBounds (15, 20, 80, 25);
    lbName = new JLabel ("Person Name:");
    lbName.setForeground (Color.black);
    lbName.setBounds (15, 55, 80, 25);
    lbDate = new JLabel ("Last Transaction:");
    lbDate.setForeground (Color.black);
    lbDate.setBounds (15, 90, 100, 25);
    lbBal = new JLabel ("Balance:");
    lbBal.setForeground (Color.black);
    lbBal.setBounds (15, 125, 80, 25);
    txtNo = new JTextField ();
    txtNo.setHorizontalAlignment (JTextField.RIGHT);
    txtNo.setBounds (125, 20, 200, 25);
    txtName = new JTextField ();
    txtName.setEnabled (false);
    txtName.setBounds (125, 55, 200, 25);
    txtDate = new JTextField ();
    txtDate.setEnabled (false);
    txtDate.setBounds (125, 90, 200, 25);
    txtBal = new JTextField ();
    txtBal.setHorizontalAlignment (JTextField.RIGHT);
    txtBal.setEnabled (false);
    txtBal.setBounds (125, 125, 200, 25);
    txtNo.addKeyListener (new KeyAdapter() {
        public void keyTyped (KeyEvent ke) {
            char c = ke.getKeyChar ();
            if (!((Character.isDigit (c) || (c == KeyEvent.VK_BACK_SPACE)))) {
                Toolkit().beep ();
                ke.consume ();
            }
        }
    });
    btnFind = new JButton ("Search");
    btnFind.setBounds (20, 165, 120, 25);
    btnFind.addActionListener (this);
    btnCancel = new JButton ("Cancel");

```

```

btnCancel.setBounds (200, 165, 120, 25);
btnCancel.addActionListener (this);
jpFind.add (lbNo);
jpFind.add (txtNo);
jpFind.add (lbName);
jpFind.add (txtName);
jpFind.add (lbDate);
jpFind.add (txtDate);
jpFind.add (lbBal);
jpFind.add (txtBal);
jpFind.add (btnFind);
jpFind.add (btnCancel);
getContentPane().add (jpFind);
populateArray ();
setVisible (true);
}
public void actionPerformed (ActionEvent ae) {
    Object obj = ae.getSource();
    if (obj == btnFind) {
        if (txtNo.getText().equals("")){
            JOptionPane.showMessageDialog (this, "Please! Provide Id of
Customer to Search.",
                "BankSystem - EmptyField", JOptionPane.PLAIN_MESSAGE);
            txtNo.requestFocus();
        }
        else {
            rows = 0;
            populateArray (); //Load All Existing Records in Memory.
            findRec (); //Finding if Account No. Exist or Not.
        }
    }
    if (obj == btnCancel) {
        txtClear ();
        setVisible (false);
        dispose();
    }
}
void populateArray () {
    try {
        fis = new FileInputStream ("Bank.dat");
        dis = new DataInputStream (fis);
        //Loop to Populate the Array.
        while (true) {
            for (int i = 0; i < 6; i++) {
                records[rows][i] = dis.readUTF ();
            }
            rows++;
        }
    }
}

```

```

        }
    catch (Exception ex) {
        total = rows;
        if (total == 0) {
            JOptionPane.showMessageDialog (null, "Records File is
Empty.\nEnter Records First to Display.",
"BankSystem - EmptyFile", JOptionPane.PLAIN_MESSAGE);
            btnEnable ();
        }
    else {
        try {
            dis.close();
            fis.close();
        }
        catch (Exception exp) { }
    }
}

void findRec () {
    boolean found = false;
    for (int x = 0; x < total; x++) {
        if (records[x][0].equals (txtNo.getText())) {
            found = true;
            showRec (x);
            break;
        }
    }
    if (found == false) {
        JOptionPane.showMessageDialog (this, "Account No. " + txtNo.getText ()
+ " doesn't Exist.",
                "BankSystem - WrongNo",
                JOptionPane.PLAIN_MESSAGE);
        txtClear ();
    }
}

//Function which display Record from Array onto the Form.
public void showRec (int intRec) {

    txtNo.setText (records[intRec][0]);
    txtName.setText (records[intRec][1]);
    txtDate.setText (records[intRec][2] + ", " + records[intRec][3] + ", " +
records[intRec][4]);
    txtBal.setText (records[intRec][5]);
}

```

```

}

//Function use to Clear all TextFields of Window.
void txtClear () {

    txtNo.setText ("");
    txtName.setText ("");
    txtDate.setText ("");
    txtBal.setText ("");
    txtNo.requestFocus ();
}
void btnEnable () {
    txtNo.setEnabled (false);
    btnFind.setEnabled (false);
}
}

```

3.5 WITHDRAW MONEY

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class WithdrawMoney extends JInternalFrame implements ActionListener {
    private JPanel jpWith = new JPanel();
    private JLabel lbNo, lbName, lbDate, lbWithdraw;
    private JTextField txtNo, txtName, txtWithdraw;
    private JComboBox cboMonth, cboDay, cboYear;
    private JButton btnSave, btnCancel;
    private int recCount = 0;
    private int rows = 0;
    private int total = 0;
    private int curr;
    private int withdraw;
    private String records[][] = new String [500][6];
    private FileInputStream fis;
    private DataInputStream dis;
    WithdrawMoney () {
        super ("Withdraw Money", false, true, false, true);
        setSize (335, 235);
        jpWith.setLayout (null);
        lbNo = new JLabel ("Account No:");
        lbNo.setForeground (Color.black);
        lbNo.setBounds (15, 20, 80, 25);
        lbName = new JLabel ("Person Name:");
        lbName.setForeground (Color.black);
        lbName.setBounds (15, 55, 80, 25);
        lbDate = new JLabel ("With. Date:");
        lbDate.setForeground (Color.black);
        lbDate.setBounds (15, 90, 80, 25);
        lbWithdraw = new JLabel ("With. Amount:");

```

```

lbWithdraw.setForeground (Color.black);
lbWithdraw.setBounds (15, 125, 80, 25);
txtNo = new JTextField ();
txtNo.setHorizontalAlignment (JTextField.RIGHT);
//Checking the Account No. Provided By User on Lost Focus of the TextBox.
txtNo.addFocusListener (new FocusListener () {
    public void focusGained (FocusEvent e) { }
    public void focusLost (FocusEvent fe) {
        if (txtNo.getText().equals ("")) { }
        else {
            rows = 0;
            populateArray (); //Load All Existing Records in Memory.
            findRec (); //Finding if Account No. Already Exist or

```

Not.

```

        }
    }
});
txtNo.setBounds (105, 20, 205, 25);

txtName = new JTextField ();
txtName.setEnabled (false);
txtName.setBounds (105, 55, 205, 25);
txtWithdraw = new JTextField ();
txtWithdraw.setHorizontalAlignment (JTextField.RIGHT);
txtWithdraw.setBounds (105, 125, 205, 25);
txtNo.addKeyListener (new KeyAdapter() {
    public void keyTyped (KeyEvent ke) {
        char c = ke.getKeyChar ();
        if (!((Character.isDigit (c) || (c == KeyEvent.VK_BACK_SPACE)))) {
            Toolkit().beep ();
            ke.consume ();
        }
    }
});
txtWithdraw.addKeyListener (new KeyAdapter() {
    public void keyTyped (KeyEvent ke) {
        char c = ke.getKeyChar ();
        if (!((Character.isDigit (c) || (c ==
KeyEvent.VK_BACK_SPACE)))) {
            Toolkit().beep ();
            ke.consume ();
        }
    }
});

```

```

String Months[] = {"January", "February", "March", "April", "May", "June",
    "July", "August", "September", "October", "November", "December"};
cboMonth = new JComboBox (Months);
cboDay = new JComboBox ();
cboYear = new JComboBox ();
for (int i = 1; i <= 31; i++) {
    String days = "" + i;
    cboDay.addItem (days);
}
for (int i = 1900; i <= 2024; i++) {
    String years = "" + i;
    cboYear.addItem (years);
}
cboMonth.setBounds (105, 90, 92, 25);
cboDay.setBounds (202, 90, 43, 25);
cboYear.setBounds (250, 90, 60, 25);
btnSave = new JButton ("Save");
btnSave.setBounds (20, 165, 120, 25);
btnSave.addActionListener (this);
btnCancel = new JButton ("Cancel");
btnCancel.setBounds (185, 165, 120, 25);
btnCancel.addActionListener (this);
jpWith.add (lbNo);
jpWith.add (txtNo);
jpWith.add (lbName);
jpWith.add (txtName);
jpWith.add (lbDate);
jpWith.add (cboMonth);
jpWith.add (cboDay);
jpWith.add (cboYear);
jpWith.add (lbWithdraw);
jpWith.add (txtWithdraw);
jpWith.add (btnSave);
jpWith.add (btnCancel);
getContentPane().add (jpWith);
populateArray (); //Load All Existing Records in Memory.
//In the End Showing the New Account Window.
setVisible (true);

}

public void actionPerformed (ActionEvent ae) {
    Object obj = ae.getSource();
    if (obj == btnSave) {
        if (txtNo.getText().equals("")){
            JOptionPane.showMessageDialog (this, "Please! Provide Id of
Customer.");
        }
    }
}

```

```

        "BankSystem - EmptyField",
JOptionPane.PLAIN_MESSAGE);
            txtNo.requestFocus();
        }
    else if (txtWithdraw.getText().equals("")) {
        JOptionPane.showMessageDialog (this, "Please! Provide Withdraw
Amount.",
                "BankSystem - EmptyField",
JOptionPane.PLAIN_MESSAGE);
            txtWithdraw.requestFocus ();
        }
    else {
        withdraw = Integer.parseInt (txtWithdraw.getText ());
        if (curr == 0) {
            JOptionPane.showMessageDialog (this, txtName.getText () +
" doesn't have any Amount in Balance.",
                "BankSystem - EmptyAccount",
JOptionPane.PLAIN_MESSAGE);
            txtClear ();
        }
        else if (withdraw > curr) {
            JOptionPane.showMessageDialog (this, "Withdraw Amount
can't greater than Actual Balance.",
                "BankSystem - Large Amount",
JOptionPane.PLAIN_MESSAGE);
            txtWithdraw.setText ("");
            txtWithdraw.requestFocus ();
        }
        else {
            editRec (); //Update the Contents of Array.
        }
    }
if (obj == btnCancel) {
    txtClear ();
    setVisible (false);
    dispose();
}
}

void populateArray () {
try {
    fis = new FileInputStream ("Bank.dat");
    dis = new DataInputStream (fis);
    //Loop to Populate the Array.
    while (true) {
        for (int i = 0; i < 6; i++) {
            records[rows][i] = dis.readUTF ();
        }
    }
}
}

```

```

        }
        rows++;
    }
}
catch (Exception ex) {
    total = rows;
    if (total == 0) {
        JOptionPane.showMessageDialog (null, "Records File is
Empty.\nEnter Records First to Display.",
                                "BankSystem - EmptyFile",
JOptionPane.PLAIN_MESSAGE);
        btnEnable ();
    }
    else {
        try {
            dis.close();
            fis.close();
        }
        catch (Exception exp) { }
    }
}
void findRec () {

    boolean found = false;
    for (int x = 0; x < total; x++) {
        if (records[x][0].equals (txtNo.getText())) {
            found = true;
            showRec (x);
            break;
        }
    }
    if (found == false) {
        String str = txtNo.getText ();
        txtClear ();
        JOptionPane.showMessageDialog (this, "Account No. " + str + " doesn't
Exist.",
                                "BankSystem - WrongNo", JOptionPane.PLAIN_MESSAGE);
    }
}
//Function which display Record from Array onto the Form.
public void showRec (int intRec) {

    txtNo.setText (records[intRec][0]);
    txtName.setText (records[intRec][1]);
    curr = Integer.parseInt (records[intRec][5]);
}

```

```

recCount = intRec;

}

void txtClear () {
txtNo.setText ("");
txtName.setText ("");
txtWithdraw.setText ("");
txtNo.requestFocus ();

}

public void editRec () {
records[recCount][0] = txtNo.getText ();
records[recCount][1] = txtName.getText ();
records[recCount][2] = "" + cboMonth.getSelectedItem ();
records[recCount][3] = "" + cboDay.getSelectedItem ();
records[recCount][4] = "" + cboYear.getSelectedItem ();
records[recCount][5] = "" + (curr - withdraw);
editFile (); //Save This Array to File.

}

public void editFile () {
try {
    FileOutputStream fos = new FileOutputStream ("Bank.dat");
    DataOutputStream dos = new DataOutputStream (fos);
    if (records != null) {
        for (int i = 0; i < total; i++) {
            for (int c = 0; c < 6; c++) {
                dos.writeUTF (records[i][c]);
                if (records[i][c] == null) break;
            }
        }
    }
    JOptionPane.showMessageDialog (this, "The File is Updated
Successfully",
                                "BankSystem - Record Saved",
                                JOptionPane.PLAIN_MESSAGE);
    txtClear ();
    dos.close(); fos.close();
}
}

catch (IOException ioe) {
    JOptionPane.showMessageDialog (this, "There are Some Problem with
File",
                                "BankSystem - Problem", JOptionPane.PLAIN_MESSAGE);
}
}

```

```

        void btnEnable () {
            txtNo.setEnabled (false);
            cboMonth.setEnabled (false);
            cboDay.setEnabled (false);
            cboYear.setEnabled (false);
            txtWithdraw.setEnabled (false);
            btnSave.setEnabled (false);

        }
    }
}

```

3.6 VIEW ALL CUSTOMERS

```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.table.DefaultTableModel;
public class ViewCustomer extends JInternalFrame {
    private JPanel jpShow = new JPanel ();
    private DefaultTableModel dtmCustomer;
    private JTable tbCustomer;
    private JScrollPane jspTable;
    private int row = 0;
    private int total = 0;
    private String rowData[][];
    private FileInputStream fis;
    private DataInputStream dis;
    ViewCustomer () {
        super ("View All Account Holders", false, true, false, true);
        setSize (475, 280);
        jpShow.setLayout (null);
        populateArray ();
        tbCustomer = makeTable ();
        jspTable = new JScrollPane (tbCustomer);
        jspTable.setBounds (20, 20, 425, 200);
        jpShow.add (jspTable);
        getContentPane().add (jpShow);
        setVisible (true);
    }
    void populateArray () {
        String rows[][] = new String [500][6];
        try {
            fis = new FileInputStream ("Bank.dat");
            dis = new DataInputStream (fis);
            while (true) {

```

```

        for (int i = 0; i < 6; i++) {
            rows[row][i] = dis.readUTF ();
        }
        row++;
    }
}
catch (Exception ex) {
    total = row;
    rowData = new String [total][4];
    if (total == 0) {
        JOptionPane.showMessageDialog (null, "Records File is
Empty.\nEnter Records to Display.", "BankSystem - EmptyFile", JOptionPane.PLAIN_MESSAGE);
    }
    else {
        for (int i = 0; i < total; i++) {
            rowData[i][0] = rows[i][0];
            rowData[i][1] = rows[i][1];
            rowData[i][2] = rows[i][2] + ", " + rows[i][3] + ", " +
rows[i][4];
            rowData[i][3] = rows[i][5];
        }
        try {
            dis.close();
            fis.close();
        }
        catch (Exception exp) { }
    }
}
}

//Function to Create the Table and Add Data to Show.
private JTable makeTable () {

    //String Type Array use to Give Table Column Names.
    String cols[] = {"Account No.", "Customer Name", "Opening Date", "Bank
Balance"};
    dtmCustomer = new DefaultTableModel (rowData, cols);
    tbCustomer = new JTable (dtmCustomer) {
        public boolean isCellEditable (int iRow, int iCol) {
            return false; //Disable All Columns of Table.
        }
    };
    (tbCustomer.getColumnModel().getColumn(0)).setPreferredWidth (180);
    (tbCustomer.getColumnModel().getColumn(1)).setPreferredWidth (275);
}

```

```

(tbCustomer.getColumnModel().getColumn(2)).setPreferredWidth (275);
(tbCustomer.getColumnModel().getColumn(3)).setPreferredWidth (200);
tbCustomer.setRowHeight (20);
tbCustomer.setSelectionMode (ListSelectionModel.SINGLE_SELECTION);
return tbCustomer;

}
}

```

3.7 VIEW PARTICULAR CUSTOMER

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class ViewOne extends JInternalFrame implements ActionListener {
    private JPanel jpRec = new JPanel();
    private JLabel lbNo, lbName, lbDate, lbBal;
    private JTextField txtNo, txtName, txtDate, txtBal, txtRec;
    private JButton btnFirst, btnBack, btnNext, btnLast;

    private int recCount = 0;
    private int rows = 0;      private int total = 0;
    private String records[][] = new String [500][6];
    private FileInputStream fis;  private DataInputStream dis;
    ViewOne () {
        super ("View Account Holders", false, true, false, true);
        setSize (350, 235);
        jpRec.setLayout (null);
        lbNo = new JLabel ("Account No:");
        lbNo.setForeground (Color.black);
        lbNo.setBounds (15, 20, 80, 25);
        lbName = new JLabel ("Person Name:");
        lbName.setForeground (Color.black);
        lbName.setBounds (15, 55, 80, 25);
        lbDate = new JLabel ("Last Transaction:");
        lbDate.setForeground (Color.black);
        lbDate.setBounds (15, 90, 100, 25);
        lbBal = new JLabel ("Balance:");
        lbBal.setForeground (Color.black);
        lbBal.setBounds (15, 125, 80, 25);
        txtNo = new JTextField ();
        txtNo.setHorizontalAlignment (JTextField.RIGHT);
        txtNo.setEnabled (false);
        txtNo.setBounds (125, 20, 200, 25);
        txtName = new JTextField ();
        txtName.setEnabled (false);

```

```

txtName.setBounds (125, 55, 200, 25);
txtDate = new JTextField ();
txtDate.setEnabled (false);
txtDate.setBounds (125, 90, 200, 25);
txtBal = new JTextField ();
txtBal.setHorizontalAlignment (JTextField.RIGHT);
txtBal.setEnabled (false);
txtBal.setBounds (125, 125, 200, 25);
btnFirst = new JButton ("<<");
btnFirst.setBounds (15, 165, 50, 25);
btnFirst.addActionListener (this);
btnBack = new JButton ("<");
btnBack.setBounds (65, 165, 50, 25);
btnBack.addActionListener (this);
btnNext = new JButton (">");
btnNext.setBounds (225, 165, 50, 25);
btnNext.addActionListener (this);
btnLast = new JButton (">>");
btnLast.setBounds (275, 165, 50, 25);
btnLast.addActionListener (this);
txtRec = new JTextField ();
txtRec.setEnabled (false);
txtRec.setHorizontalAlignment (JTextField.CENTER);
txtRec.setBounds (115, 165, 109, 25);
jpRec.add (lbNo); jpRec.add (txtNo); jpRec.add (lbName);
jpRec.add (txtName); jpRec.add (lbDate); jpRec.add (txtDate);
jpRec.add (lbBal);      jpRec.add (txtBal);
jpRec.add (btnFirst);   jpRec.add (btnBack);
jpRec.add (btnNext);    jpRec.add (btnLast);
jpRec.add (txtRec);
getContentPane().add (jpRec);
populateArray ();         showRec (0);
setVisible (true);
}
public void actionPerformed (ActionEvent ae) {
    Object obj = ae.getSource();
    if (obj == btnFirst) {
        recCount = 0;
        showRec (recCount);
    }
    else if (obj == btnBack) {
        recCount = recCount - 1;
        if (recCount < 0) {
            recCount = 0;
            showRec (recCount);
            JOptionPane.showMessageDialog (this, "You are on First Record.",
                                         "BankSystem - 1st Record",
                                         
```

```

JOptionPane.PLAIN_MESSAGE);
        }
    else {
        showRec (recCount);
    }
}
else if (obj == btnNext) {
    recCount = recCount + 1;
    if (recCount == total) {
        recCount = total - 1;
        showRec (recCount);
        JOptionPane.showMessageDialog (this, "You are on Last Record.",
                                     "BankSystem - End of Records",
                                     JOptionPane.PLAIN_MESSAGE);
    }
    else {
        showRec (recCount);
    }
}
else if (obj == btnLast) {
    recCount = total - 1;
    showRec (recCount);
}
}

void populateArray () {

try {
    fis = new FileInputStream ("Bank.dat");
    dis = new DataInputStream (fis);
    //Loop to Populate the Array.
    while (true) {
        for (int i = 0; i < 6; i++) {
            records[rows][i] = dis.readUTF ();
        }
        rows++;
    }
}
catch (Exception ex) {
    total = rows;
    if (total == 0) {
        JOptionPane.showMessageDialog (null, "Records File is
Empty.\nEnter Records First to Display.",
                                     "BankSystem - EmptyFile",
                                     JOptionPane.PLAIN_MESSAGE);
        btnEnable ();
    }
    else {
}
}
}

```

```

        try {
            dis.close();
            fis.close();
        }
        catch (Exception exp) { }
    }

}

//Function which display Record from Array onto the Form.
public void showRec (int intRec) {

    txtNo.setText (records[intRec][0]);
    txtName.setText (records[intRec][1]);
    txtDate.setText (records[intRec][2] + ", " + records[intRec][3] + ", " +
records[intRec][4]);
    txtBal.setText (records[intRec][5]);
    if (total == 0) {
        txtRec.setText (intRec + "/" + total); //Showing Record No. and Total
Records.
        txtDate.setText ("");
    }
    else {
        txtRec.setText ((intRec + 1) + "/" + total); //Showing Record No. and Total
Records.
    }
}

//Function use to Lock all Buttons of Window.
void btnEnable () {

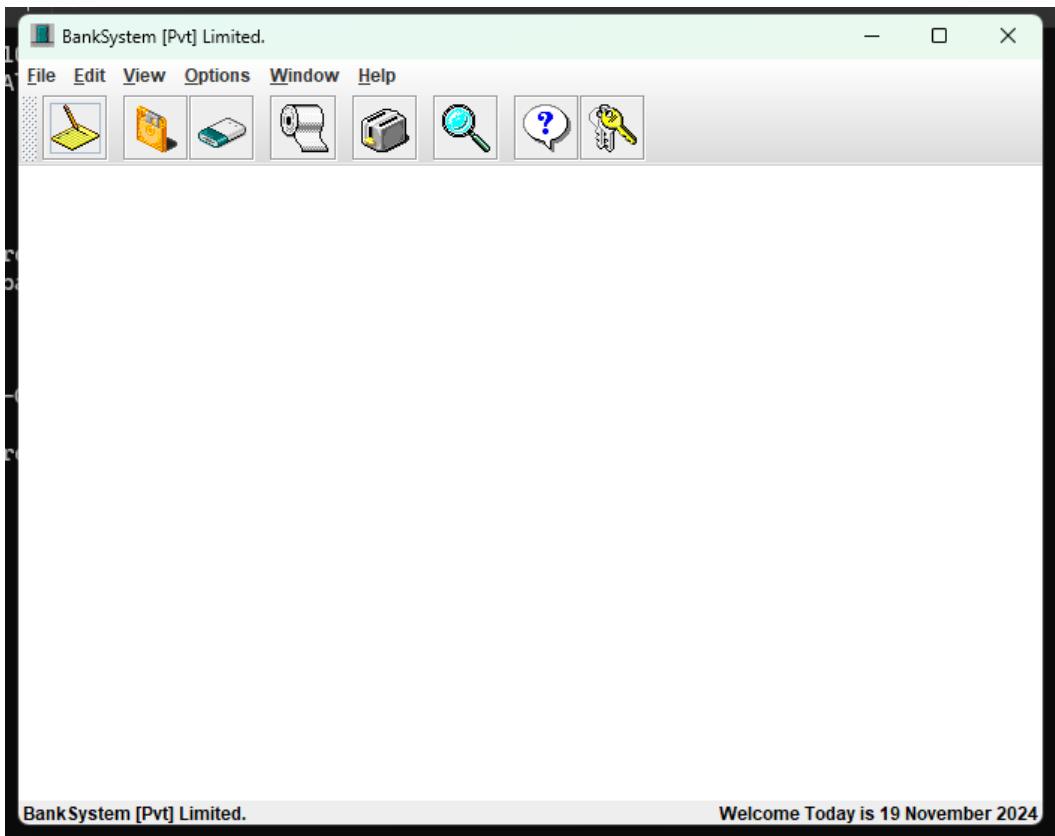
    btnFirst.setEnabled (false);
    btnBack.setEnabled (false);
    btnNext.setEnabled (false);
    btnLast.setEnabled (false);

}

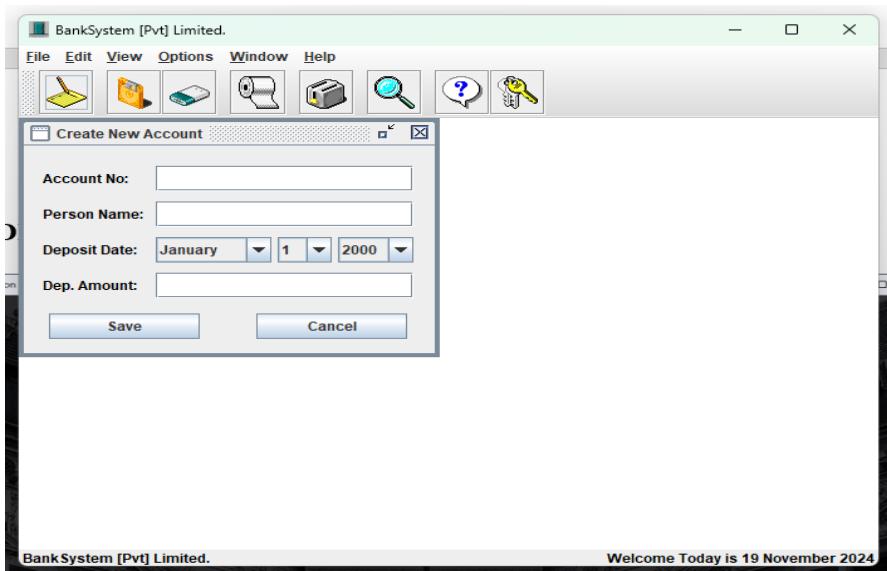
```

SNAPSHOTS

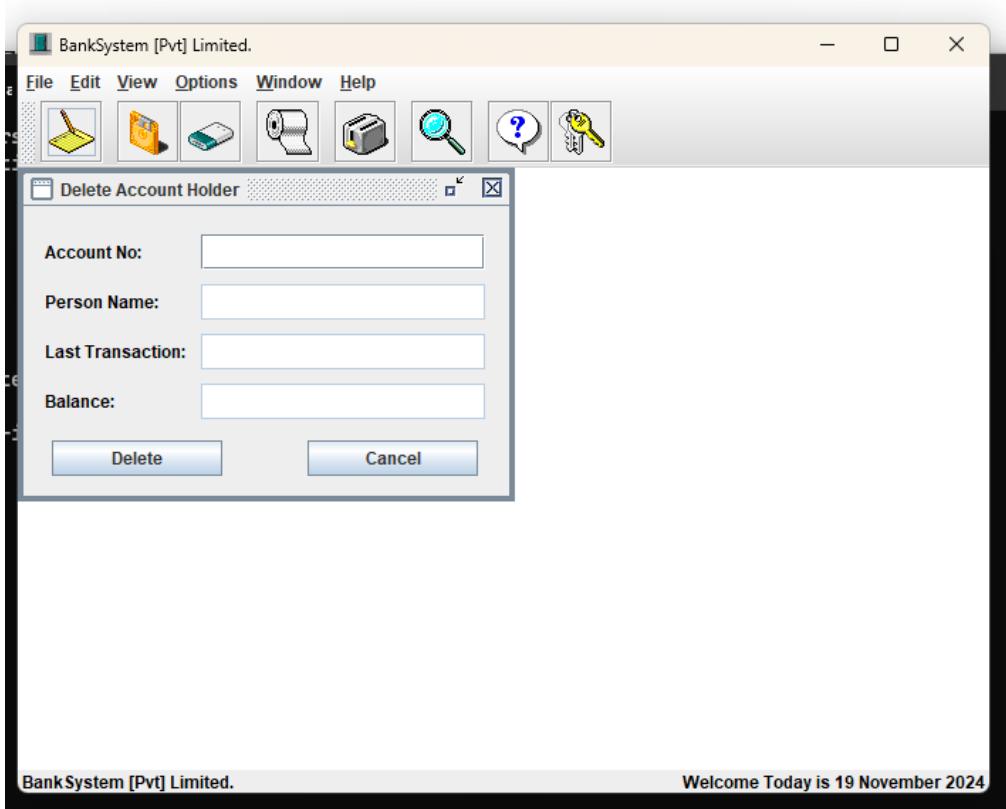
4.1 HOME PAGE



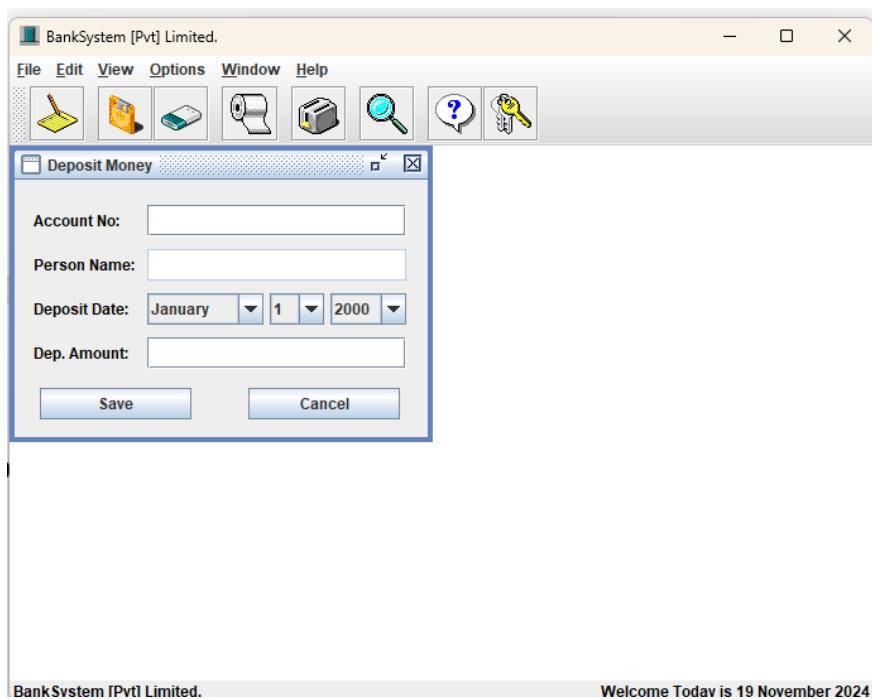
4.2 CREATE NEW ACCOUNT PAGE



4.3 DELETING AN ACCOUNT PAGE



4.4 DEPOSIT MONEY PAGE



4.5 FIND AN ACCOUNT PAGE

Search Customer [By No.]

Account No:	<input type="text"/>
Person Name:	<input type="text"/>
Last Transaction:	<input type="text"/>
Balance:	<input type="text"/>

Search Customer [By Name]

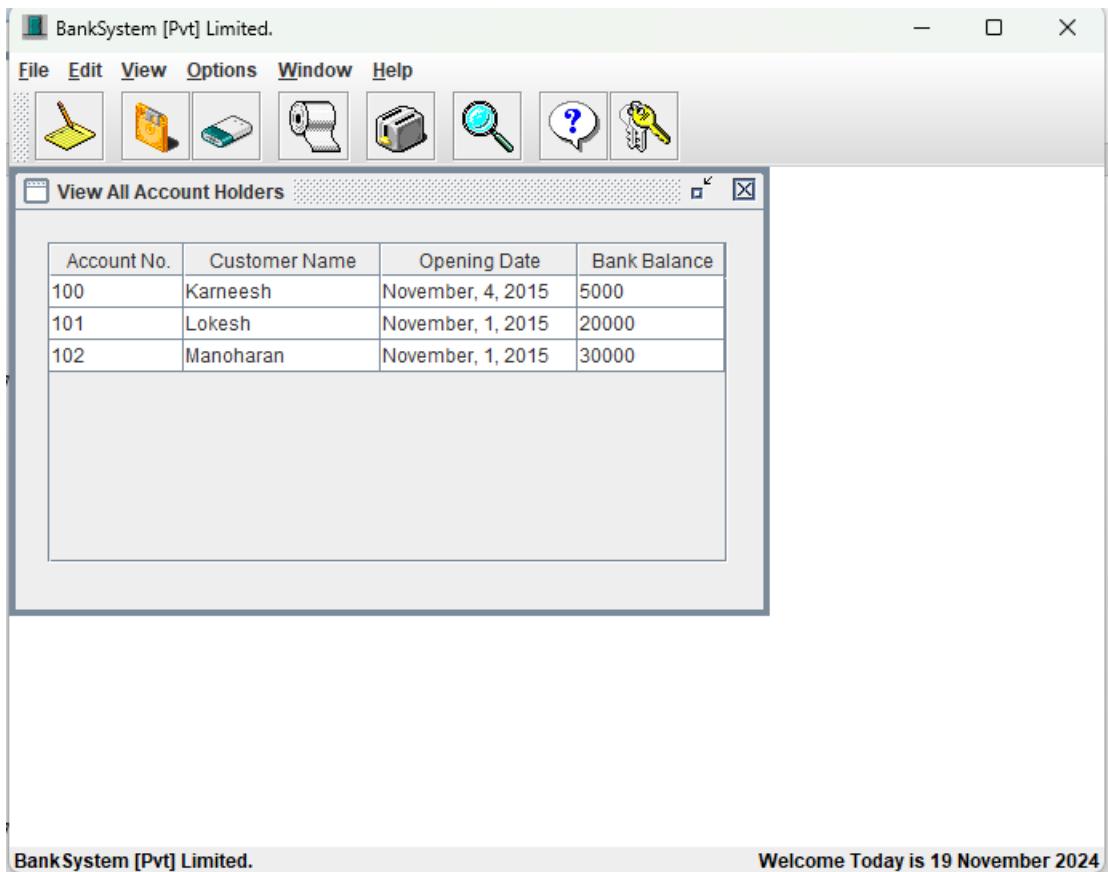
Account No:	<input type="text"/>
Person Name:	<input type="text"/>
Last Transaction:	<input type="text"/>
Balance:	<input type="text"/>

4.6 WITHDRAW AN ACCOUNT PAGE

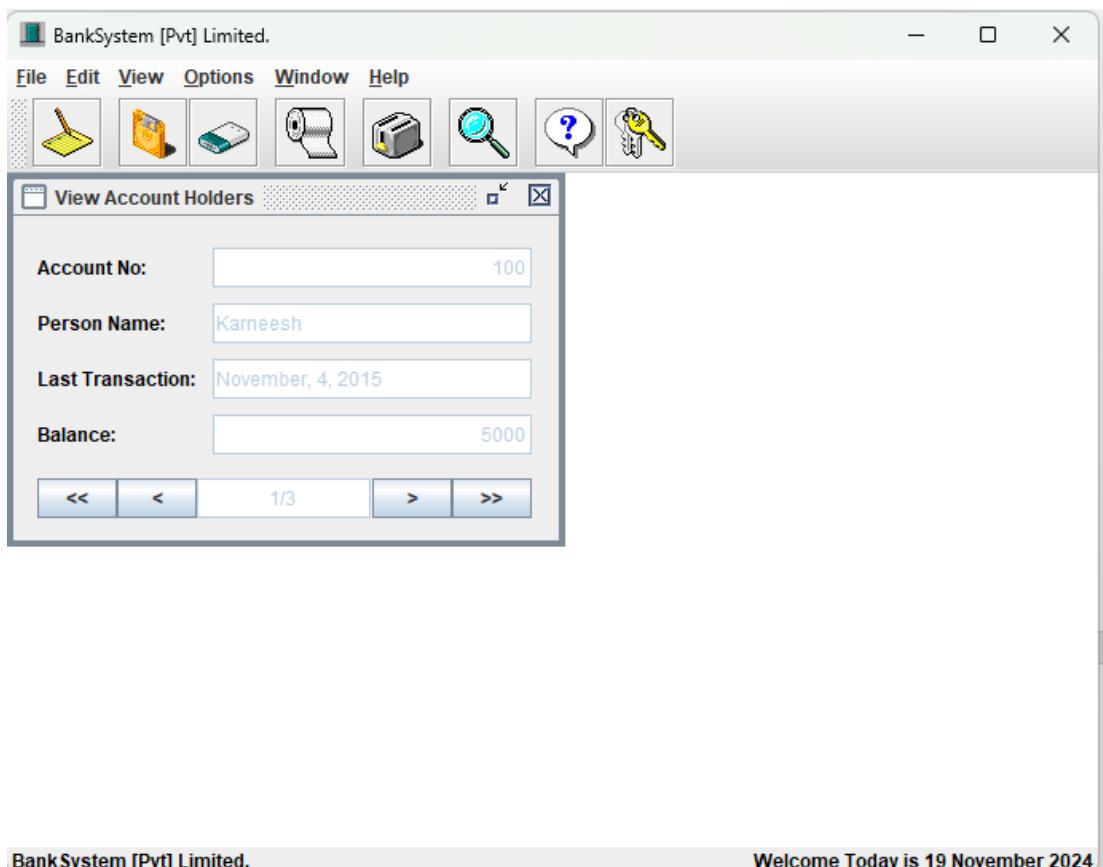
Withdraw Money

Account No:	<input type="text"/>
Person Name:	<input type="text"/>
With. Date:	<input type="button" value="January"/> <input type="button" value="1"/> <input type="button" value="2000"/>
With. Amount:	<input type="text"/>

4.7 VIEW ALL CUSTOMER PAGE



4.8 VIEW PARTICULAR CUSTOMER PAGE



CONCLUSION

With the help of our project, bankers will be able to do their banking work without any cumbersome and save their precious time as well as they can manage many customers at the same time without any difficulty and it makes a good relationship between customers and bankers.

REFERENCES

- 1.** <https://www.javatutorial.com>
- 2.** <https://www.geeksforgeeks.org>
- 3.** <https://www.w3schools.com/sql/>
- 4.** [https:// www.programmiz.com](https://www.programmiz.com)
- 5.** [https:// www.oracle.com](https://www.oracle.com)
- 6.** [SQL | Codecademy](#)